

平成 24 年度 卒業論文

VR 化学実験体験システムのための
タッチパネル 3D インタフェースの提案

指導教員
舟橋 健司 准教授

名古屋工業大学 工学部 情報工学科
平成 21 年度入学 21115019 番

内山 享佑

目次

第 1 章	はじめに	1
第 2 章	粒子・体積に基づく仮想液体モデル	4
2.1	液体モデル	4
2.2	仮想容器のモデル	5
2.2.1	容器の表現	5
2.2.2	液体の流出点, および液位	7
2.3	容器と液体の相互作用モデル	9
2.3.1	自由落下状態の液体と容器の相互作用	9
2.3.2	容器内状態の液体と容器の相互作用	9
2.3.3	液面上昇	10
2.3.4	容器内への液体の流入	10
2.3.5	容器外への液体の流出	11
2.4	粒子・体積モデルの機能の拡張	11
2.4.1	液面の波や渦の表現	12
2.4.2	色の拡散	14
2.4.3	凹形状容器の表現	17
第 3 章	2次元平面を介した 3 自由度回転操作	18
3.1	球体回転のインタフェースモデルについて	19
3.1.1	投影された円から 3次元球体座標への変換	19
3.1.2	中心のみが固定された球の回転	21
3.1.3	スクリーンに平行な軸での回転	23
3.2	地理座標系を取り入れた球体回転モデル	25

3.2.1	3次元球体上の点の地理座標	25
3.2.2	基準点の回転による容器の傾きと方向	28
3.3	投影された円周上付近における判定	31
第4章	実験とその結果	34
4.1	実験システム	34
4.2	実験の様子	34
4.3	典型的な2種類の回転インタフェースの操作感の評価実験	39
4.4	地理座標系を取り入れた球体回転モデルの操作感の評価実験	40
第5章	むすび	42
付録		43
A.1	Native Activity について	43
A.2	OpenGL と OpenGL ES について	43
A.3	OpenGL ES のためのプログラム改変	44
A.3.1	GL_DOUBLE 型の削除	44
A.3.2	glMaterialfv の仕様	44
A.3.3	物体描画方法の変更	44
謝辞		46
参考文献		47

第1章 はじめに

近年，バーチャルリアリティ(Virtual Reality :以下 VR) 技術に関する研究が多く行われており，我々にとっても身近なものとなってきている．ここで，バーチャルとは正しくは「仮想」という意味ではなく，「みかけや形はそのものではないが，本質あるいは効果としてはそのものであること」という意味である．そのため VR 技術とは，コンピュータが人間に視覚や触覚といった様々な感覚をもった環境や物体を人工的に提示し，人間にあたかもそれが現実の世界であるように体験させ，コンピュータと対話することができる技術のことである [1]．

例えば，VR 技術の応用例として訓練や実験用の模擬装置としてのシミュレータが存在する．これは物理法則や化学法則などの様々な法則をモデル化することによって，様々な状況のシミュレーションを行うシステムのことである．また，操作者の行動をシミュレータ上の VR 環境に反映させることにより高い臨場感を操作者に与えることができる．シミュレータを用いることによって，医療分野においては訓練を行うことが難しい外科手術の回転骨切り術や内視鏡外科手術のシミュレーションを行うことが可能である [2][3]．もちろん，医療分野以外においても火災体験のシミュレーションや，航空機開発のためのフライトシミュレータなども研究されている [4][5]．このように，VR 技術は様々な分野において用いられ我々の生活に関係してきている．何故 VR 技術を用いた様々なシミュレータが研究されてきているかと言うと，VR 技術を用いることによる利点として実験や訓練コストの削減や，実施の容易さ，現実世界で行うにはリスクが伴う作業におけるリスクの回避などがあるためである．

これら VR 技術の研究においては，主に以下の二つの点のどちらかに重みをおいた研究を行うかが重要になってくる．1つ目は，如何に現実と同じような法則を再現し，実物体の挙動を厳密に再現するかというものである．例えば，液体の表現を VR 環境で再現しようとする時，流体力学を用いることで水面や水しぶきをいった表現を行うことが可能である [6]．しかし，流体力学を用いた表現を行う際，複雑な計算式を処理する必要があり，操作者の行動

を反映するには時間がかかる。そのため、現実では観察しにくい環境のシミュレーションには向いていると言えるが、操作者との対話操作に向いているとは言えない。2つ目は、操作者が何らかの操作を行い、その上でその操作に対するフィードバックを得るという点に着目し対話操作を重視するものである。対話操作を実現するためには、厳密な挙動よりも高速な処理速度を優先する必要がある。

ところで当研究室では、小学生程度の子供を対象とした VR 化学実験体験システムを開発している [7]。これはピーカーやフラスコを実際に操作することで実験の手順や化学反応を確認することが可能なシステムである。なお、各薬品の組み合わせに対する反応は既知であるとしている。臨場感の高い化学実験体験システムを構築するには様々な技術が必要であるが、その1つに液体の薬品の表現がある。液体の挙動の表現するための研究の一つとして、Smoothed Particle Hydrodynamics (以下 SPH 法)がある。SPH 法とは粒子法の一つであり、流体を粒子の集合であると考え一つ一つの粒子に速度や密度といった情報を与え、物理法則を用いることでより詳細で正確な挙動の再現を可能としている。これを3次元の粘性流体に適用した研究が提案されており、精度の高い対話操作を実現している [8]–[11]。正確な挙動を再現するためには膨大な計算が必要であり、また高い性能の計算機が必要となる。しかし、一般家庭で利用してもらうシステムは安価に構成する必要がある。

そこで、当研究室では VR 化学実験体験システムのために、安価な PC でも高速な処理速度を実現できる仮想液体モデルを提案している [12][13]。本モデルでは対話操作を重視するため、計算量を抑える必要がある。そこで本モデルでは、液体を自由落下状態と容器内状態に分けて考え、液体が前者の状態では粒子、後者の状態では体積とした二つの状態に分けて表現している。なお、この粒子は互いに干渉していないものとし、計算の高速化を図っている。この二つの状態を組み合わせることにより、操作者は流れ落ちる液体を受け止め別の容器に移す、容器内の液体を別の容器ですくう、容器の液体をこぼすなどの動作の実現が可能である。これにより操作者は実際に自身で、容器を介して液体を操作していると感じることができる。

ところで前述のシステムでは、液体をすくったりこぼしたりする際に使用する容器の操作は3次元モーションセンサを介して行っている。近年のゲームコントローラーに見られるように、3次元位置を入力するための装置は安価になりつつあるが、可搬性の面からは、より移動の容易なシステムも需要があると考えられる。また一方、ノート PC の小型軽量高性能化に

加えて、キーボードを持たないタブレット型 PC も普及してきている。タブレット PC は、電子教科書や e-learning 端末として教育現場にも導入が進んでいる [14]。そこで本研究では、タブレット上での VR 化学実験体験システムの実現を目指す。タブレット上でのシステム構築は単なるプログラムの移植作業だけでなく、3次元空間に存在する対象物の6自由度の移動を、タブレットで主に用いられているタッチパネルにより実現する必要がある。この2次元平面を介した3次元空間に対する操作は、マルチタッチでない場合には、本質的にマウスによる操作と同等である。関連する研究として、テーブルトップを用いた三次元身体モデルを操作する研究も行われているが、ここでは視点の移動や物体の変更移動、拡大縮小を行うものであり、対象物の回転については考慮されていない [15]。また 3DCG ソフト、Blender では物体の回転を行う際、 x 軸、 y 軸、 z 軸 および画面に垂直な軸を用いた回転が可能である [16]。しかし、任意の軸まわりの回転を実現したい場合にも、各軸それぞれの回転により表現する必要があり、直感的な操作が可能であるとはいえない。本研究ではタッチパネルによる3自由度回転に注目し、いくつかの実現方法を考察した上で、緯線、経線に沿って地球を回すようなモデルを提案し、直感的な回転を実現する。

以下、2章ではタブレット PC への導入を行う VR 化学実験体験システムのための粒子・体積に基づく仮想液体操作モデルの概略を説明する。また3章ではタッチパネルのような2次元平面を介した3自由度回転を行うモデルについて説明し、4章では実験及び結果、5章では本研究のまとめを述べる。なお、付録として化学実験体験システムのタブレット PC への移植を行った際に生じた問題点とその解決方法について述べる。

第2章 粒子・体積に基づく仮想液体モデル

本章では，当研究室でこれまでに提案されている粒子・体積に基づく仮想液体操作モデルの概要について述べる．

粒子・体積モデルでは，計算量を抑えるため液体全てを粒子で表現するのではなく，液体を自由落下状態と容器内状態に分けて考える．自由落下状態でのみ液体を粒子として捉え，容器に溜まっている液体のように容器内状態である液体については体積で表現を行う．また，実際に操作者が液体との対話操作を行う際に想定する挙動をより自然に可能とするための対話性の高いモデルを実現している．

2.1 液体モデル

当研究室の粒子・体積モデルでは，液体を以下の2つの状態に分けて考える．

- (1) 自由落下状態 (蛇口からこぼれ落ちる液体)
- (2) 容器内状態 (コップなどに溜まっている液体)

最初に状態(1)についての説明を行う．この状態については粒子に基づいた表現を行う．しかし，本モデルでは対話操作を重視しているため粒子一つ一つについての大きさは考慮せず，粒子同士での衝突や分子間力による影響もおこらないものとする．各粒子の次元仮想空間内での時刻 t における座標を $\mathbf{P}_i(t)$ ($i = 1, 2, \dots, N$; N : 粒子数) とし，それぞれの粒子は重力および慣性のみに従い移動する．移動量と重力による加速を考慮すると， $\mathbf{P}_i(t)$ は式(2.1)により決定される．

$$\mathbf{P}_i(t) = 2\mathbf{P}_i(t - \Delta t) - \mathbf{P}_i(t - 2\Delta t) + \mathbf{g}\Delta t^2 \quad (2.1)$$

ここで， \mathbf{g} は下向きの重力加速度ベクトルであり， Δt は描画更新間隔，すなわち描画更新レート ($frames/sec.$) の逆数である．空気抵抗による広がりを表現する場合は，式(2.1)右辺に微小乱数値 $\mathbf{R}_i(t)$ を加える．

次に、後者の状態 (2) の液体については体積に基づいた表現を行う。この状態においては、容器内の対流などは計算量を抑えるため考慮しない。すなわち、ある時刻 t において、仮想空間内に定義された容器内に液体が存在している場合、その容器内には体積 V_t の液体が存在していると表現する。また、先述した粒子としての性質とは異なるものとするため、一切考慮しないこととしている。上記2つの状態間での状態の遷移があった場合には、換算比 N [粒子数/体積] による換算を行う。例えば状態 (2) から状態 (1) への遷移を考える。容器内に体積 V の容器内状態の液体が存在するとし、それらを全て容器外へ落下させた場合は容器内の液体体積は 0 になり、自由落下状態の液体の粒子数は NV 増加する。

2.2 仮想容器のモデル

仮想空間内において液体の操作を行うための仮想容器についての説明を行う。凹形状容器については、凸形状容器への分割を行うことで対応可能である [17]。そこで本節では、容器の並行移動や回転操作が可能である凸形状容器の表現の説明を行う。これらの操作を組み合わせることによって、容器で液体をすくう、液体が溜っている容器を運ぶ、容器から液体を流出させる操作が可能である。

2.2.1 容器の表現

まず、図 2.1 のように容器に対して、それを内包する球を考える。その内包球の中心を C 、半径を r とし、容器の世界座標系における位置はベクトル C により表現される。このベクトル C により、容器の平行移動については表現できる。

次に容器の回転操作を考える。図 2.2 のように、 C を原点とし容器座標系 (軸 x_c, y_c, z_c) を設定する。容器の傾きは、軸 y_c と球との交点 (図 2.2 の d_1) の位置の変化 (極座標的な表現) により表す。具体的には、 x_c を軸として点 d_1 が点 d_2 に移動するように容器を角度 ϕ だけ回転し、更に z_c を軸として点 d_2 が点 d_3 に移動するように容器を角度 θ だけ回転する。これにより、 ϕ, θ の2つのパラメータで容器の傾きを表現することが可能である。例えば、値がともに 0 の時には、容器が傾いていない状態である。また、容器の方向の変化としては y_c を軸として角度 ψ だけ回転を行う。これにより、実際の容器の向きについては点 d_4 によって表現される。 ψ の回転による容器の傾きについては変化が無い場合、容器内に満たしうる

液量の変化はない．これら $C \theta \phi \psi$ は空間内に存在する容器それぞれに設定されており移動する容器については時刻 t の関数として表す．

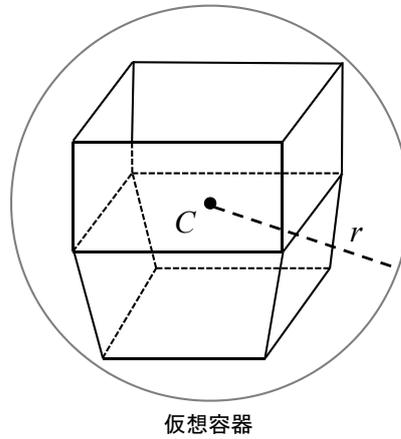


図 2.1: 仮想容器を内包する球

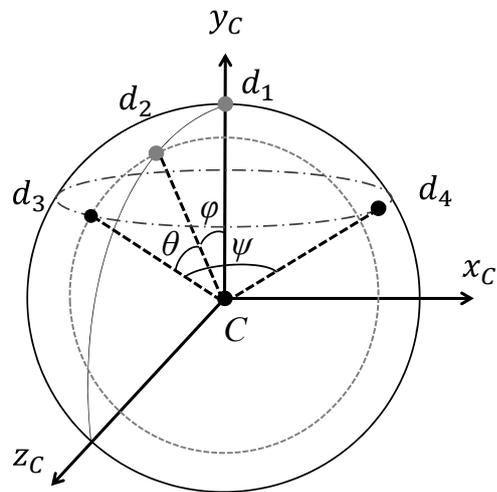


図 2.2: 傾きと向きの変現

2.2.2 液体の流出点，および液位

容器の液体が状態 (2) から状態 (1) に遷移する，つまり容器を傾け液体をこぼす状況を考える．その際の仮想容器の各パラメータを図 2.3 に示す．この時，容器内に入りうる液体の最大量を最大容積 V_m とすると，液体の体積が V_m を超えた場合に流出点 F から流出が起きるものとする．厳密に計算すると，流出点は線分になる場合もあるが実際にはそのようなことは稀であるため，本モデルでは流出は一点からおこるものとしている．流出点 F を点 C を基準とするとする相対的なベクトル \mathbf{F} として式 (2.2) により表す．

$$\mathbf{F} = f(\theta, \phi, \psi) \quad (2.2)$$

また，容器内で静止している液体の高さを液位と呼ぶ．ある時刻 t における液位は， C から鉛直方向へ H の位置であるとする．この時の容器内に存在する液体の体積が $V (V \geq 0)$ であるとする， H は式 (2.3) により表す．

$$H = h(\theta, \phi, V) \quad (2.3)$$

ここで液位 H の取りうる範囲は容器が内包される半径を r としているため， $-r < H < r$ となる．しかし，実際には容器の形状や傾きによってさらに狭い範囲に存在することとなる．また，式 (2.3) を体積 V について変形することで式 (2.4) を得る．

$$V = v(\theta, \phi, H) \quad (2.4)$$

式 (2.4) において容器より下方の H を引数として与えた場合は $V = 0$ ．点 F よりも上方の H を与えた場合は F の鉛直成分の y 座標を F_y とし， $V = v(\theta, \phi, F_y) = V_m(\theta, \phi)$ となる．これは， F_y が角度 ψ に依存しないためである．また，液面よりも下方の容器の厚みを考慮した時の液位 H 時の体積を V^+ とし，式 (2.5) と図 2.4 で表す．

$$V^+ = v^+(\theta, \phi, H) \quad (2.5)$$

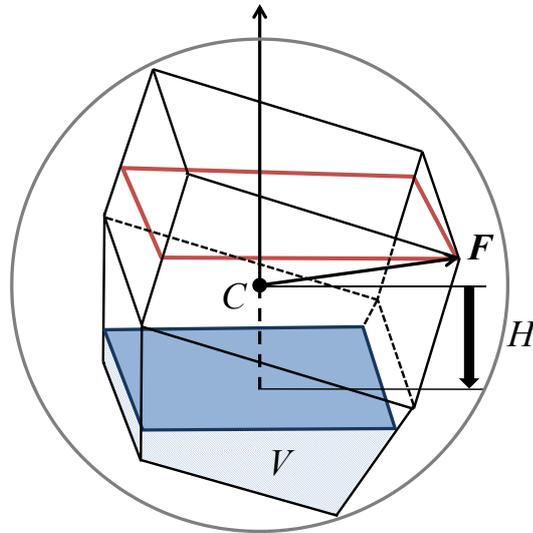


図 2.3: 仮想容器のパラメータ

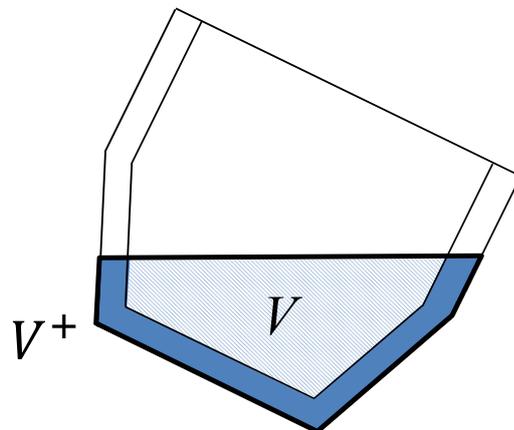


図 2.4: 仮想容器の厚みの考慮

2.3 容器と液体の相互作用モデル

本節では、流れ落ちる液体を容器で受け止める、液体が溜まっている容器に異なる容器を入れ液体をすくう、容器を傾け液体をこぼすなどの動作を行う際の自由落下状態および容器内状態の液体と仮想容器との相互作用について説明する。

2.3.1 自由落下状態の液体と容器の相互作用

自由落下状態の液体の粒子が液面や容器に達した場合、液体の状態を状態 (1) から (2) へと変換させる。すなわち、粒子の状態である液体を消滅させ一定の換算比をもとに体積へと変換させる必要がある。体積に変換するべき粒子が n 個 ($n \geq 0$) の場合、その容器内の時刻 t における液体の体積を式 (2.20) で表す。

$$V(t)^{(0)} = V(t - \Delta t) + n/N \quad (2.6)$$

次節で述べる容器内状態の液体と容器との干渉がなければ、時刻 t における容器内の液体の体積を $V(t)^{(1)} = V(t)^{(0)}$ とし、液位 H は式 (2.7) により決定される。

$$H(t) = h(\theta, \phi, V(t)^{(1)}) \quad (2.7)$$

2.3.2 容器内状態の液体と容器の相互作用

操作者が自由に移動、回転が可能な容器を容器 1、移動させられない容器を容器 2 とし、液体との相互作用について述べる。各式の添字 1, 2 はそれぞれ容器 1, 容器 2 に対応している。ある時刻 t において式 (2.8) が成立する場合、容器 1 と容器 2 内の液体は互いに干渉しているとする。

$$C_{1y} - r_1 < C_{2y} + h_2(\theta_2, \phi_2, V_2^{(0)}) \quad (2.8)$$

ここで C_{jy} は容器 j の内包球の中心 C_j の鉛直方向の座標である。このとき、容器 2 内の液体の液位が容器 1 に上昇させられている体積を $U_{21} \geq 0$ とし、式 (2.9) により表す。

$$U_2 = \sum_j U_{2j} \quad (2.9)$$

容器と液体が干渉しており，かつ式 (2.10) が成立するとき，容器 1 の流出点よりも下の部分と容器 2 の液面が干渉してゐる．

$$C_{2y} + h_2(\theta_2, \phi_2, V_2^{(0)} + V_{m1}^+) < C_{1y} + F_{1y} \quad (2.10)$$

式 (2.10) が成立しない場合，つまり容器 2 内の液体中に容器 1 の流出点が存在していることとなり，容器 2 内の液体が容器 1 に流入する．

2.3.3 液面上昇

容器と液面が干渉しているとき，容器 1 が容器 2 内の液体の液位を上昇させている．このときの液位 H_2 を式 (2.11)，式 (2.12) により表す．

$$H_2 = h_2(\theta_2, \phi_2, V_2^{(0)} + U_{21}) \quad (2.11)$$

$$U_{21} = V_1 + (\theta_1, \phi_1, C_{2y} + H_2 - C_{1y}) \quad (2.12)$$

また，それぞれの容器内の液体の体積を $V_j^{(1)} = V_j^{(0)}$ とし，容器 1 の液位を式 (2.7) より求められる．

2.3.4 容器内への液体の流入

容器 1 に容器 2 の液体が流入し，さらに式 (2.13) が成立するとき，液体の流入は容器 2 の液面が容器 1 の流出点に達した時点で止まる．

$$V_2^{(0)} + U_{21} - v_2(\theta_2, \phi_2, C_{1y} + F_{1y} - C_{2y}) < V_{m1} - V_1^{(0)} \quad (2.13)$$

この時，容器 2 の液位が容器 1 の流出点となるように H_2 を式 (2.14) により表す．

$$H_2 = (C_{1y} + F_{1y}) - C_{2y} \quad (2.14)$$

これに合わせて，各容器の体積を式 (2.15)，式 (2.16) により計算する．また，容器 1 の液位を式 (2.7) により求める．

$$V_2^{(1)} = v_2(\theta_2, \phi_2, H_2) - U_{21} \quad (2.15)$$

$$U_1^{(1)} + V_2^{(1)} = V_1^{(0)} + V_2^{(0)} \quad (2.16)$$

式(2.13)が成立しない場合は、容器1の流出点より容器2の液面が上方に存在していることとなる。この場合、容器1の液体の体積 $V_1^{(1)}$ を最大容量 V_{m1} 、液位 H_1 を r_1 とし、液面は考慮せず表示しないものとする。この時の容器2の液体の体積と液位は式(2.16)および式(2.17)により表す。

$$H_2 = h_2(\theta_2, \phi_2, V_2^{(1)} + U_{21}) \quad (2.17)$$

以上のように設定することにより、時刻 $t + \Delta t$ において2つの容器間での液体の移動がなくなった場合に、液体をすくい取る操作が可能となる。

2.3.5 容器外への液体の流出

容器の移動による容器同士の干渉や、容器の傾きにより、時刻 t において式(2.18)を満たす場合、液体が容器から流出する。

$$V^{(1)} + U > V_m \quad (2.18)$$

流出が起きた場合、流出する液体の体積は式(2.19)となる。

$$V_f = V^{(1)} + U - V_m \quad (2.19)$$

流出した液体は自由落下状態となるため、液体の状態は体積から粒子へと変換される。よって、 NV_f 個の粒子を点 F から流出させる。また、時刻 t における容器内の体積は式(2.20)により表される。

$$V(t) = V_m - U \quad (2.20)$$

なお、式(2.18)を満たさない場合には液体の流出がないものとし、 $V(t) = V(t)^{(1)}$ とする。

2.4 粒子・体積モデルの機能の拡張

本節では、本章でこれまで紹介してきたモデルに基づいて研究されてきた、液面における波や渦の発生、液面や容器側面における色の拡散、また凸形状の容器だけでなくさまざまな容器の形状を表現するための手法を説明する[17]–[21]。なお、より詳しい手法については各小節で紹介する文献を参照されたい。

2.4.1 液面の波や渦の表現

液体が容器内に存在し静止状態である時，液面が波立つ表現を行う．この表現のため，液面を図 2.5 のように直行の格子状に分割した平面として考え，各格子点の高さの計算を行う．このため，各格子点は鉛直方向のみに移動可能とし，この格子点の波の伝搬 2 次元波動方程式を差分方程式として解くことで波を表現する [18]．

また渦に関しては，図 2.6 のように液面を直行格子状だけでなく同心円格子としても考える．直行格子上の任意の点に対応する同心円格子状の点を求めることで，波の伝搬が円弧方向となり，擬似的な渦の表現が可能である [19]．

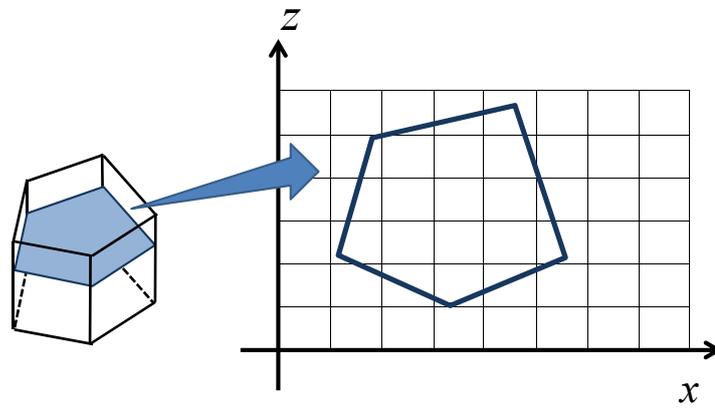


図 2.5: 液面を格子状に分割

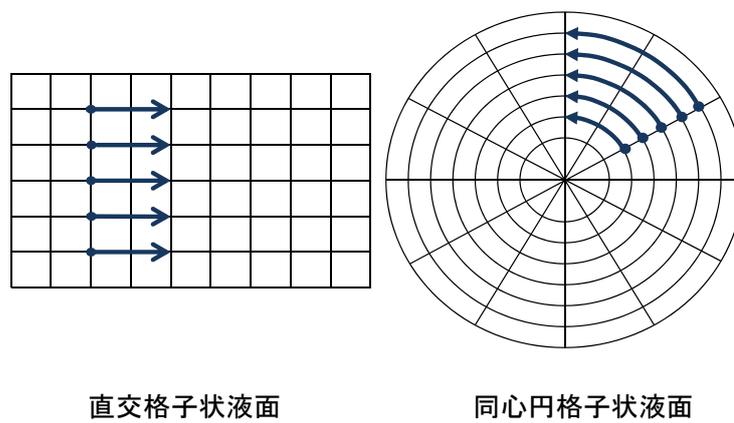


図 2.6: 分割した格子の対応

2.4.2 色の拡散

本モデルでは前小節のように波の表現の際に液面を格子状に分割し、各格子点で2次元波動方程式を計算することで表現を行う(図2.7)。ただし、色の拡散表現に奥行きを持たせるために若干の厚みを持つ3次元格子とする(図2.8)。この各格子点に、更に色のベクトル値(RGBA値)、および色の伝搬を実現するための重みを表す変数を含める。この色の重みを拡散方程式を用いて拡散させ、ある時刻 t の格子点の色と拡散してきた色とを重みの割合にしたがい混合し、時刻 $t + \Delta t$ の色ベクトル値を算出し、液面の色の拡散表現を行う。

また、容器側面における色の拡散に関しては、図2.9のように、容器中央に鉛直方向に平行な直行格子状平面を用意する。この格子面において液面と同様に色の拡散を行い、その後、格子面を分割して容器正面の側面にマッピングすることにより、容器側面においても色の拡散を表現する [21]。

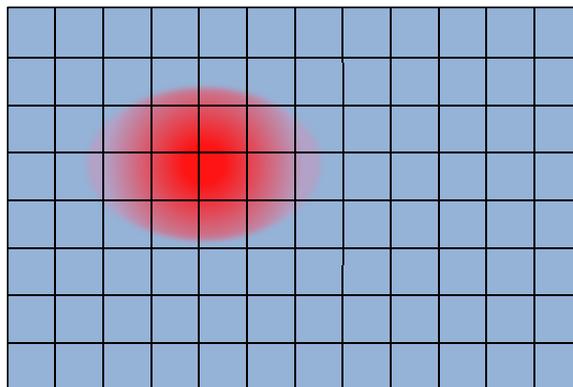


図 2.7: 液面における色の拡散

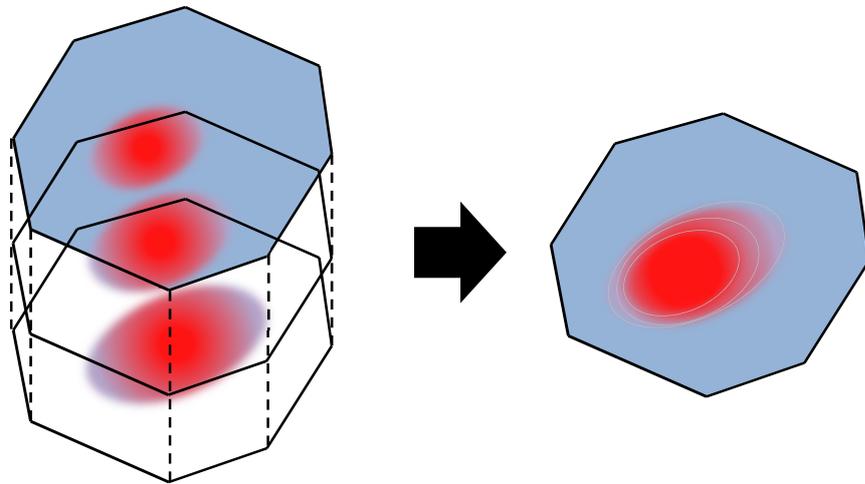


図 2.8: 複数の格子の重ね合わせ

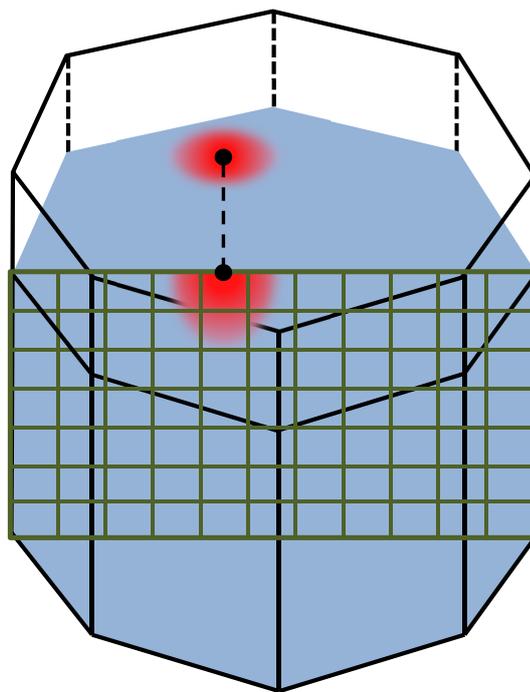


図 2.9: 容器側面の表現のための直交格子

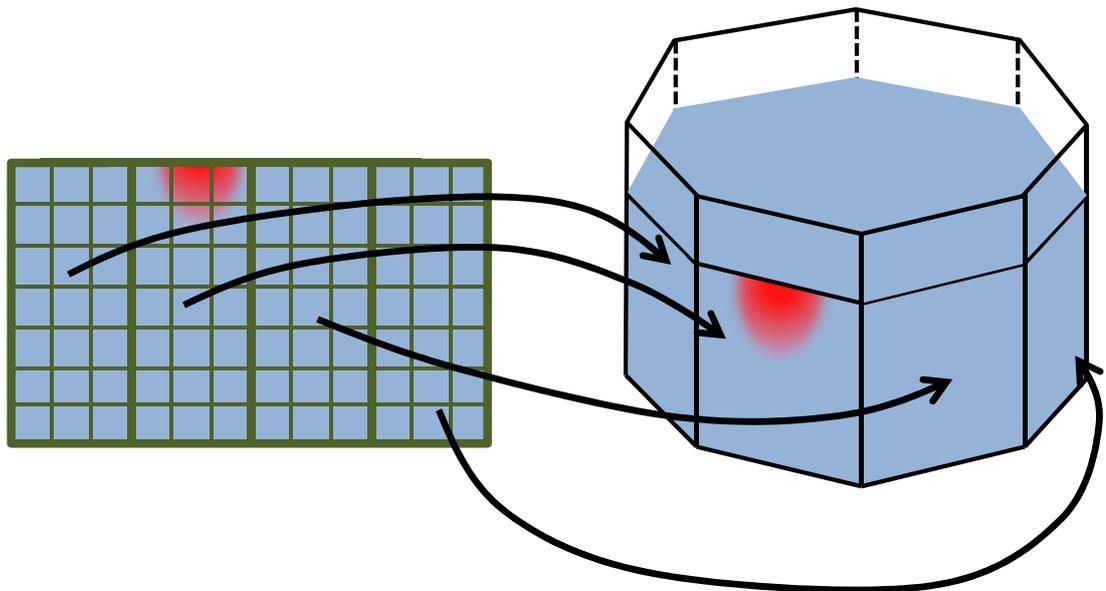


図 2.10: 側面のマッピング

2.4.3 凹形状容器の表現

2.2節で説明した容器モデルは容器が凸形状である場合を想定していた。しかし、実際に液体操作を行う際にはフラスコのように凹形状の容器を用いることもある。そこで、図2.11のように凹形状容器を複数の凸形状容器の組み合わせで考え、それぞれの凸形状容器間での液体の移動を考えることで、凹形状容器における液体操作を実現する [17]。

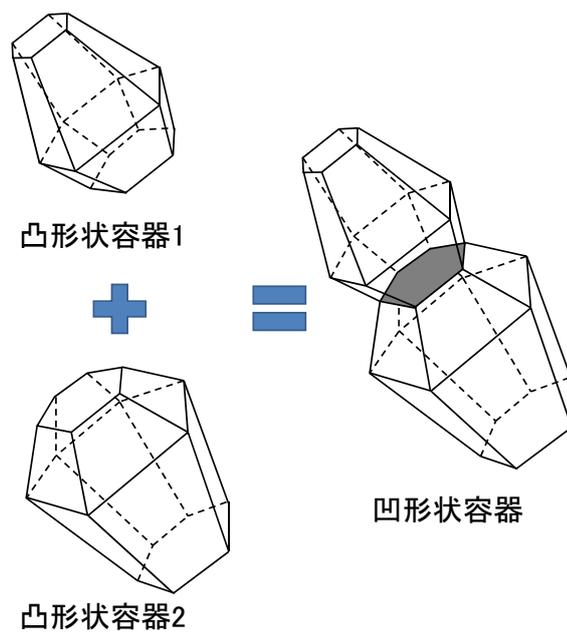


図 2.11: 凹形状容器の表現

第3章 2次元平面を介した3自由度回転操作

当研究室では、2章で述べたように、VR化学実験体験システムの実現を一つの目標とした粒子・体積に基づく仮想液体操作モデルを提案している。本研究では、VR化学実験体験システムの普及を期待し、ノートPCに比べ可搬性が高く実際に教育現場への導入が進んでいるタブレットPCへの実装を目指す。しかしピーカーやフラスコなどの容器操作を従来のように3次元モーションセンサを介して行うのは可搬性の面からも困難である。そのため、本研究ではタッチパネルを用いた容器の移動操作、特に容器の3自由度の回転を行うインタフェースを提案する。具体的には、容器を内包する球を回すようなインタフェースを提供することで、直感的な操作を実現する。ただし、球体の回転についてはいくつかの方法が考えられ、一長一短がある。そこで球体に対し緯度、経度を表す地理座標系を考え、タッチパネルに対する指の動きを緯度成分、経度成分に分解した上で、経線、緯線に沿って地球を回すようなインタフェースモデルを提案する。

3.1 球体回転のインタフェースモデルについて

本節では3自由度の回転をタッチパネルを介して行うインタフェースモデルについて考察する。まず、タッチパネル上の座標から3次元空間座標への対応について示した後、2つの典型的なインタフェースモデルについて述べる。

3.1.1 投影された円から3次元球体座標への変換

まず、タッチパネルのような2次元平面を介して3次元空間内の球を回転させるためのインタフェースに必要となる、2次元座標から3次元空間の球体上の座標への変換について説明する。図3.1のように、3次元空間上にある容器の内包球がタッチパネル上に円として投影されているものとする。投影された円内の指の点 A から B への軌跡は、3次元空間の球体上では図3.2のように点 a から b への軌跡に対応付けられる。この時、理解を容易にするためタッチパネル上に半球(断面が円に一致)があるとする。タッチパネル上のタッチ位置が座標 (X, Y) であり、円の半径が R であるとき、この仮想半球上の指のスクリーンからの高さは式(3.1)で表せる。

$$Z = \sqrt{R^2 - X^2 - Y^2} \quad (3.1)$$

すなわち、3次元空間内の球体上の対応点を座標 (x, y, z) とすると、球の半径を r に対して式(3.2)が成り立つ。

$$\frac{r}{R} = \frac{x}{X} = \frac{z}{Z} = \frac{y}{Y} \quad (3.2)$$

この関係式より、タッチ位置に対応する3次元座標を得ることができる。

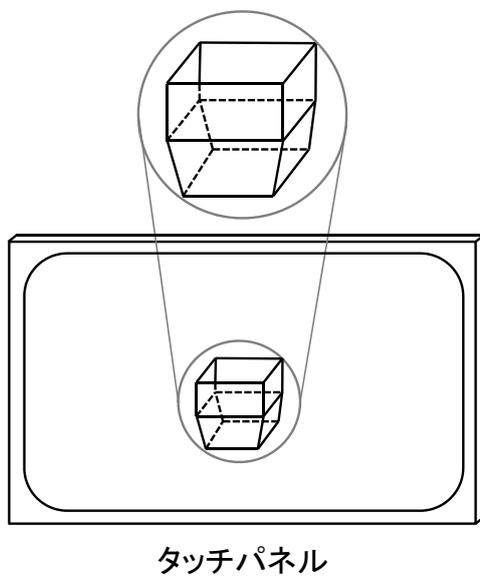


図 3.1: 容器のタッチパネル上への投影

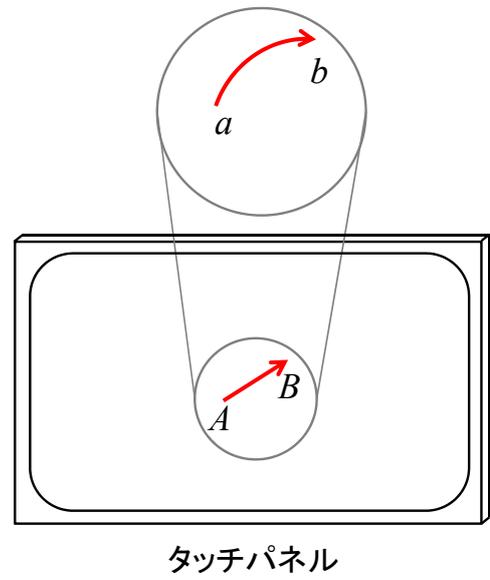


図 3.2: 円上と球上の動きの違い

3.1.2 中心のみが固定された球の回転

図 3.3 に示すように，タッチパネル上で指が点 A から点 B へと移動した場合，ベクトル a と b の外積を回転軸 Q とし，これらのベクトルの成す角 λ を回転角とする．しかし球体をスクリーンに平行な軸で回転したい場合には円の中心を通るように指を動かす必要がある (図 3.4(a))．中心から離れた軌跡による回転 (図 3.4(b)) も誤りがあるわけではないが，図 3.5 に示すような地球儀を回すような回転を思い描いた場合に違和感を覚える．

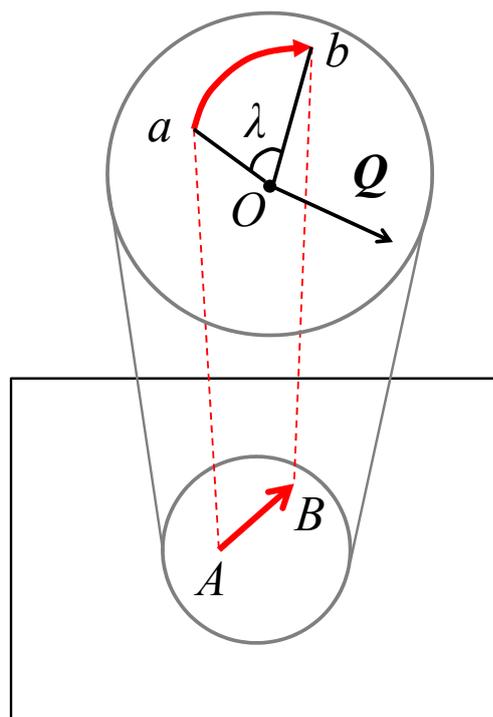


図 3.3: 中心を固定した回転

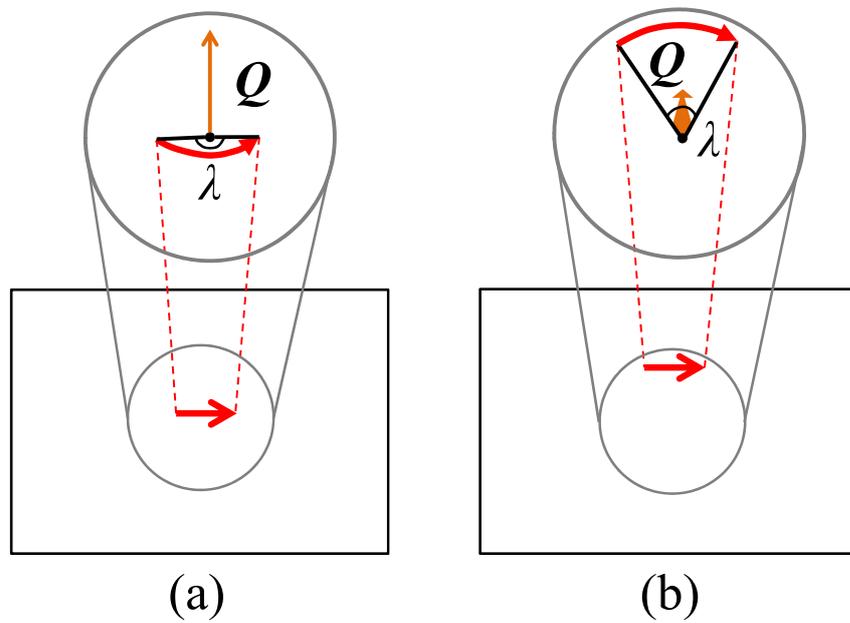


図 3.4: 指の位置により異なる回転軸

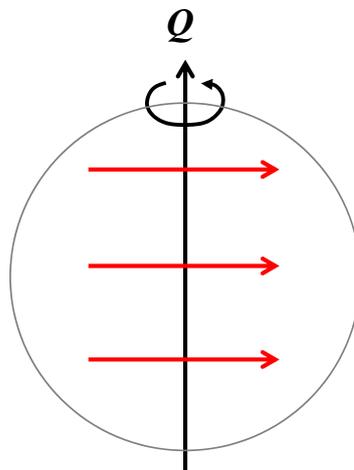


図 3.5: 地球を想定した場合の期待する回転

3.1.3 スクリーンに平行な軸での回転

そこで、回転の軸を常にスクリーンに平行とすることを考える。図3.6に示すように、タッチパネル上の指の動きの軌跡に垂直で球の中心を通る軸において、ベクトル $o'a$ と $o'b$ の成す角 λ だけ回転する。これにより、前節での違和感を解消することができる。しかし、今度は、前節の方法では円周上に軌跡を描くことで可能であったスクリーンに垂直な軸における回転ができなくなる(図3.7)。

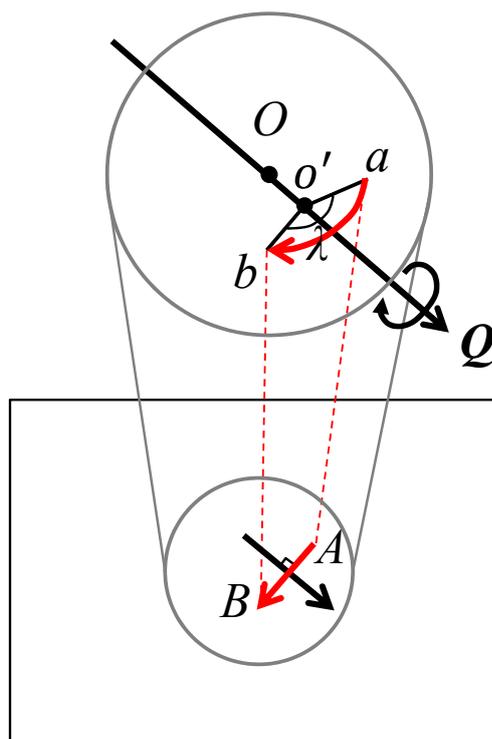


図 3.6: 軸をスクリーンと平行とした回転

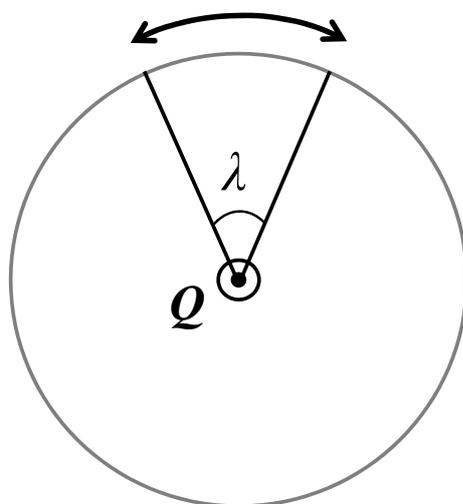


図 3.7: 回転軸がスクリーンに垂直な回転

3.2 地理座標系を取り入れた球体回転モデル

本節では3.1節で考察した2つのモデルの問題点を解消するため、球体に地理座標系(緯度経度座標系)を取り入れたインタフェースモデルを提案する(図3.8)。地理座標系とは地球(球体)の表面上の地理的な位置、つまり経度緯度を表すものである。図3.8において同じ緯度上にある線を緯線、同じ経度上にある線を経線と称する。また、球体上のある点は、直交座標系: (x, y, z) 、極座標系: (θ, ϕ) 、地理座標系: (α, β) のように複数の表現が可能である。なお、球の半径 r に対して $x^2 + y^2 + z^2 = r^2$ が成り立つ。本提案モデルでは指の軌跡を緯度成分と経度成分へ分解し、それぞれの成分において球体の回転を考える。

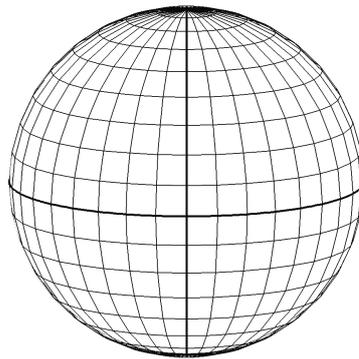


図 3.8: 地理座標系で表される球体

3.2.1 3次元球体上の点の地理座標

本節では、タッチパネル上の点 $U(X_U, Y_U)$ に対応する球体上の点 $u(x_u, y_u, z_u)$ の地理座標表現 (α_u, β_u) を求める(図3.9)。まず、 α_u の導出を行う。球体を xy 平面の断面図で考える(図3.10)。この時、 α_u は u を緯線に沿って円周上へと移動させた点 u' と x 軸との角度によって求められる。 $x_{u'}$ は式(3.3)より求められる。

$$x_{u'} = \pm \sqrt{r^2 - y_u^2} \quad (3.3)$$

r は球体の半径を表し、 $x_{u'}$ の正負については $x_u \geq 0$ の時は正、 $x_u < 0$ の時は負とする。なお本モデルでは $y_u < 0$ の時、つまり南緯を表すときは α_u は負であるものと考えとする。次に、 β_u の導出を行う。球体を xz 平面の断面図で考える(図3.11)。 u と x 軸との角度によ

り β_u は求められる．なお本モデルにおいてはタッチパネルに投影された円から球体の表面の座標に変換する関係上， $z_u \geq 0$ の値しか取らないため， β_u は常に正の値を取る．

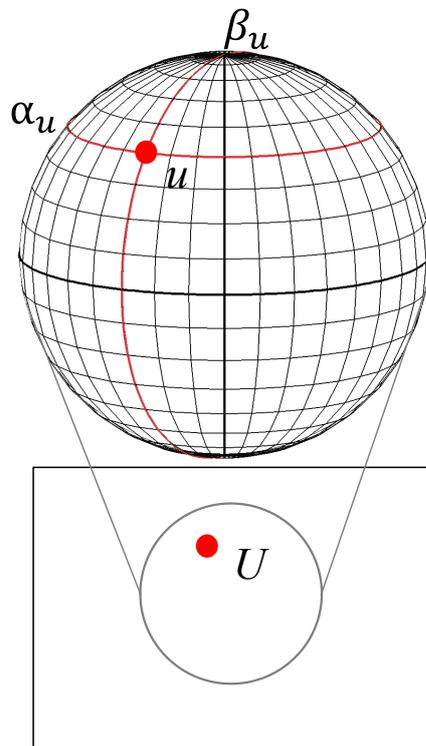


図 3.9: 球体の地理座標系

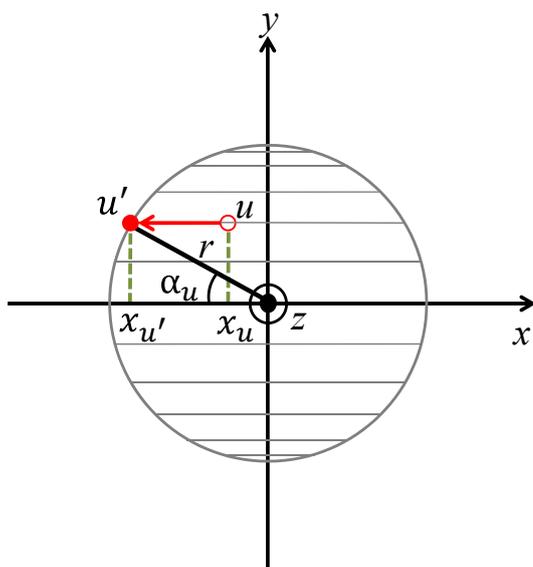


図 3.10: 球体の緯度座標

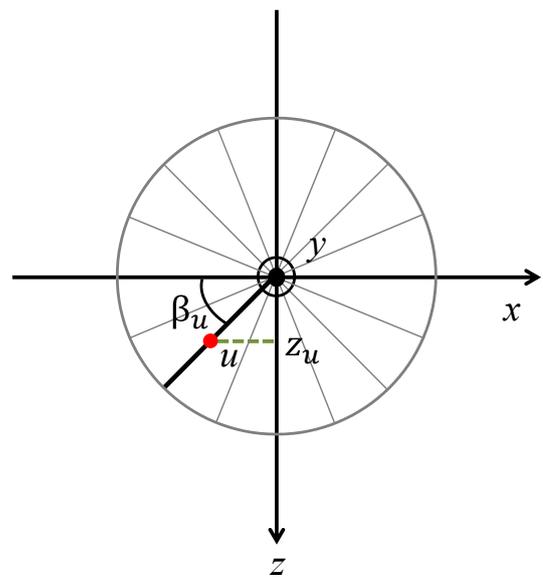


図 3.11: 球体の経度座標

3.2.2 基準点の回転による容器の傾きと方向

球体上の点は，中心からの距離が一定であり，2つのパラメータで表現が可能である．しかし，3次元物体の3次元空間における回転では，例えばこの点と中心を結ぶ軸まわりの回転が加わり，3つのパラメータが必要である．2章ではこれを (θ, ϕ, ψ) と定義した．なお， (θ, ϕ, ψ) は 2.2.1 節より極座標的な表現をするため，値域については， $-90 \leq \theta \leq 90$ ， $-180 \leq \phi \leq 180$ ， $-180 \leq \psi \leq 180$ とする．本節では容器の (θ, ϕ, ψ) を決定するための基準点の球体に伴う回転について述べる．

ある時刻における容器の傾きと向きを表すための基準点を2つ設定する．1つ目の基準点は，図 2.2 を参考に $d_1 = (0, 1, 0)$ とする．この時， d_4 の座標 $(x_{d_4}, y_{d_4}, z_{d_4})$ は式 (3.4) より表される．

$$\begin{cases} x_{d_4} = \sin \phi \sin \psi - \sin \theta \cos \phi \cos \psi \\ y_{d_4} = \cos \theta \cos \phi \\ z_{d_4} = \sin \phi \cos \psi + \sin \theta \cos \phi \sin \psi \end{cases} \quad (3.4)$$

2つ目の基準点は図 3.12 で示す $e_1 = (1, 0, 0)$ に対する e_4 とする．なお，この点は ϕ による回転の影響を受けない． e_4 の座標 $(x_{e_4}, y_{e_4}, z_{e_4})$ は式 (3.5) より表される．

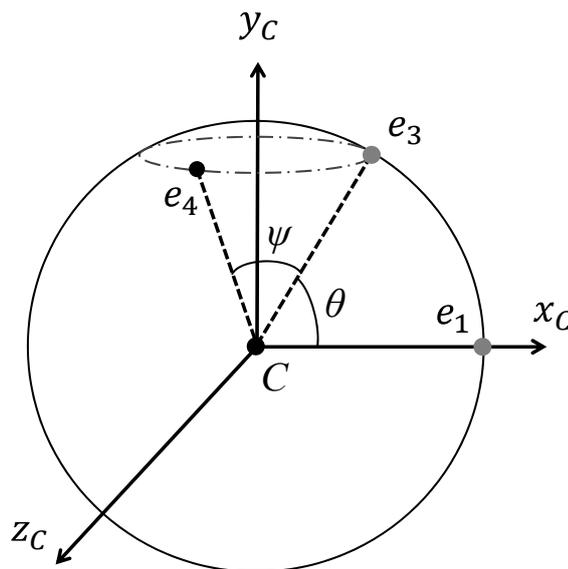


図 3.12: 基準点 e_4 の動き

$$\begin{cases} x_{e_4} = \cos \theta \cos \psi \\ y_{e_4} = \sin \theta \\ z_{e_4} = -\cos \theta \sin \psi \end{cases} \quad (3.5)$$

これら2つの基準点がタッチパネルの入力による球体の回転に伴い回転すると考える。ある時刻 t における容器パラメータ $(\theta_t, \phi_t, \psi_t)$ が既知であるとする。式(3.4), 式(3.5), において $(\theta, \phi, \psi) = (\theta_t, \phi_t, \psi_t)$ とすることで時刻 t における2つの基準点の直角座標が求まる。タッチパネル上の移動前の点を A , 移動後の点を B とする。また, 3次元空間の球体上の対応する点を a, b とする。 a, b の座標は3.1.1節により求まる。この時, 指の動きを緯度成分と経度成分へと分解するため, w を定義する(図3.13)。 a, b の地理座標をそれぞれ $(\alpha_a, \beta_a), (\alpha_b, \beta_b)$ とすると, w は (α_b, β_a) と表せる。なお, w を (α_a, β_b) とすることも考えられるが, 短い時間間隔における微小移動量に対しては問題となるほど大きな違いはないと仮定する。以上の値を用い時刻 $t + \Delta t$ における容器パラメータ $(\theta_{t+\Delta t}, \phi_{t+\Delta t}, \psi_{t+\Delta t})$ を求める。

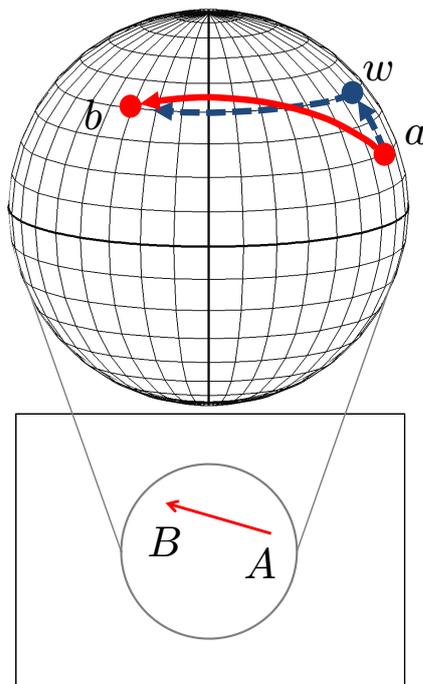


図 3.13: 指の移動成分の分解

まず経度成分，すなわち経度の変化分 $(b - w)$ での回転を考える．すなわち，地球儀が地軸回りに回転することを考える．そのため， ψ のみが $\Delta\psi$ だけ変化する．この $\Delta\psi$ は経度の変化と等しく，式 (3.6) より求まる．

$$\Delta\psi = \beta_b - \beta_a \quad (3.6)$$

次に緯度成分，すなわち緯度の変化分 $(w - a)$ での回転を考える．すなわち，描かれた等経度線が見た目上，動かないような回転を行う．具体的にはまず，時刻 $t + \Delta t$ における回転後の2つの基準点の座標を求める．回転角 λ は緯度の変化と等しく，式 (3.7) により求まる．

$$\lambda = \alpha_b - \alpha_a \quad (3.7)$$

回転軸ベクトル Q は a, w , 原点を含む面の法線ベクトル，つまり β_a の経度線を含む面の法線ベクトルである (図 3.14)．よって Q ベクトルの成分を (x_Q, y_Q, z_Q) とすると式 (3.8) と表せる．

$$\begin{cases} x_Q = -z_a \sin 90 + x_a \cos 90 \\ y_Q = 0 \\ z_Q = z_a \cos 90 + x_a \sin 90 \end{cases} \quad (3.8)$$

回転軸ベクトル Q と角度 λ での回転による，ある点 (x, y, z) の点 (x', y', z') への移動は式 (3.9) より表せる．なお， e_Q は Q の単位ベクトルを表し， e_Q の成分を $(x_{e_Q}, y_{e_Q}, z_{e_Q})$ とする．

$$\begin{cases} x' = (x_{e_Q} x_{e_Q} (1 - \cos \lambda) + \cos \lambda)x \\ \quad + (x_{e_Q} y_{e_Q} (1 - \cos \lambda) - z_{e_Q} \sin \lambda)y \\ \quad + (x_{e_Q} z_{e_Q} (1 - \cos \lambda) + y_{e_Q} \sin \lambda)z \\ y' = (y_{e_Q} x_{e_Q} (1 - \cos \lambda) + z_{e_Q} \sin \lambda)x \\ \quad + (y_{e_Q} y_{e_Q} (1 - \cos \lambda) + \cos \lambda)y \\ \quad + (y_{e_Q} z_{e_Q} (1 - \cos \lambda) - x_{e_Q} \sin \lambda)z \\ z' = (z_{e_Q} x_{e_Q} (1 - \cos \lambda) - y_{e_Q} \sin \lambda)x \\ \quad + (z_{e_Q} y_{e_Q} (1 - \cos \lambda) + x_{e_Q} \sin \lambda)y \\ \quad + (z_{e_Q} z_{e_Q} (1 - \cos \lambda) + \cos \lambda)z \end{cases} \quad (3.9)$$

回転軸ベクトル Q と角 λ による回転で2つの基準点 d_4, e_4 が基準点 d'_4, e'_4 に移動する． d'_4, e'_4 の座標は式 (3.9) により表せる．この2点は直交座標系で考えている．これを式 (3.4) および式 (3.5) より，以下の通り極座標系で示す．なお $-90 \leq \theta' \leq 90$ であるため，

$$\theta' = \sin^{-1} y_{e'_4} \quad (3.10)$$

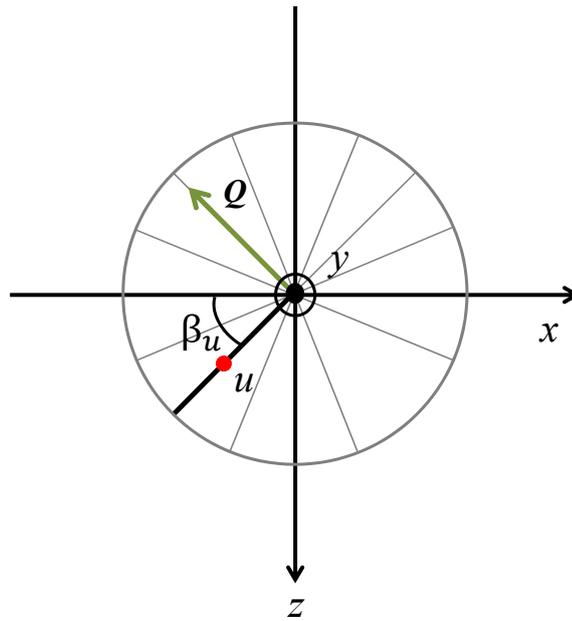


図 3.14: β_a の経度線を含む面に垂直な法線

また $-180 \leq \psi' \leq 180$ であるため ,

$$\psi' = \tan^{-1} \frac{-z_{e'_4}}{x_{e'_4}} \quad (3.11)$$

さらに $-180 \leq \phi' \leq 180$ であるため ,

$$\phi' = \tan^{-1} \frac{x_{d'_4} \sin \psi' + z_{d'_4} \cos \psi'}{\frac{y_{d'_4}}{\cos \theta'}} \quad (3.12)$$

よって , 時刻 $t + \Delta t$ における容器パラメータ $(\theta_{t+\Delta t}, \phi_{t+\Delta t}, \psi_{t+\Delta t})$ は式 (3.13) の通りである .

$$\begin{cases} \theta_{t+\Delta t} = \theta' \\ \phi_{t+\Delta t} = \phi' \\ \psi_{t+\Delta t} = \psi' + \Delta\psi \end{cases} \quad (3.13)$$

このように , タッチパネル上での指の軌跡をもとに時刻 $t + \Delta t$ における容器の極座標的表現を得る .

3.3 投影された円周上付近における判定

3.2 節で説明したモデルでは円周上をなぞる指の動きを行えばタッチパネルに垂直な軸による回転が可能である . しかし , 実際にはタッチパネル上で円周上を正確になぞる事はタッ

チパネルや指の大きさ，タッチ位置の判定を考えると困難である．

そこで本モデルではタッチパネル上に投影された内包球を表す円を S_1 としたとき，スクリーン上で半径が事前に定めた分だけ長い補助円 S_2 を同時に描画する (図 3.15)．円 S_1 と補助円 S_2 の間が触られた時，その点を I_2 とする (図 3.16(a))．この時， I_2 と円の中心を結んだ直線と S_1 との交点を I_1 とし， I_1 を実際に処理する点とする (図 3.16(b))．投影された円の中心を c ，半径を R とすると I_1 は c から I_2 への直線の単位ベクトルを R 倍したものである．よって I_2, c の座標をそれぞれ $(x_{I_2}, y_{I_2}), (x_c, y_c)$ とすると， I_1 の座標 (x_{I_1}, y_{I_1}) は式 (3.14) により表される．

$$\begin{cases} x_{I_1} = x_c + R \frac{x_{I_2} - x_c}{\sqrt{(x_{I_2} - x_c)^2 + (y_{I_2} - y_c)^2}} \\ y_{I_1} = y_c + R \frac{y_{I_2} - y_c}{\sqrt{(x_{I_2} - x_c)^2 + (y_{I_2} - y_c)^2}} \end{cases} \quad (3.14)$$

I_1 を用い 3.2 節の処理を行うことで，円周上をなぞる操作の実現が容易に可能である．

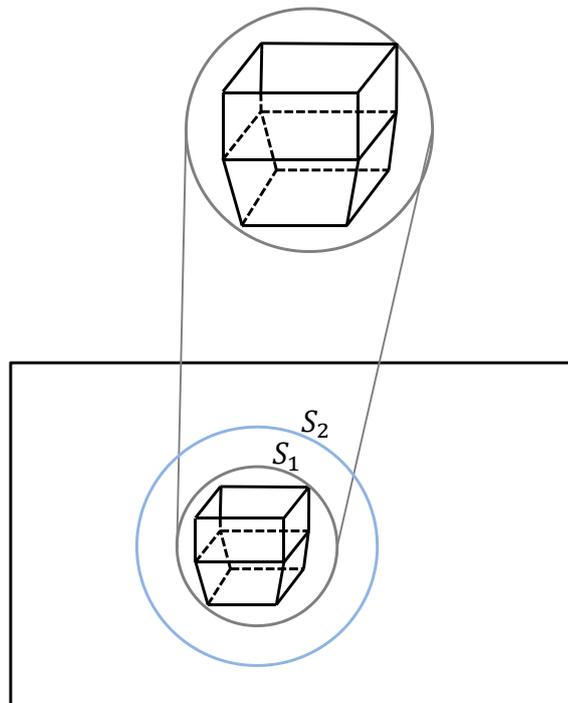


図 3.15: 円周上と判定するために設定する円

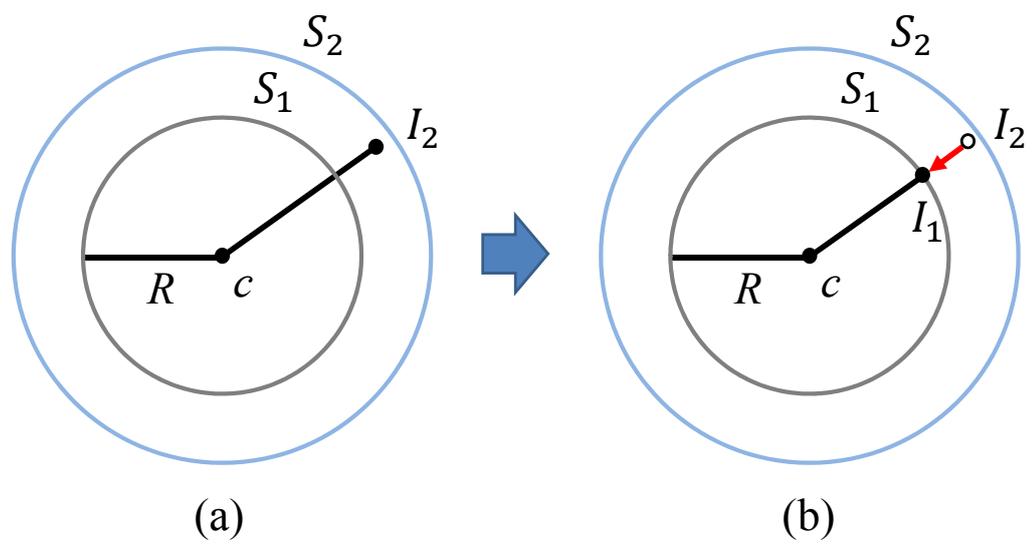


図 3.16: 円と補助円の間と判定されたときの点

第4章 実験とその結果

本章では、3章で提案した地理座標系を取り入れた球体回転モデルを実際にタブレット PC 上で動作させると共に、操作者の期待する回転が可能であるかの評価実験を行う。また、あわせて 3.1 節で述べた典型的な 2 つのモデルについても比較を行うために評価実験を行う。

4.1 実験システム

上述のモデルに基づいた VR 化学実験体験システムのための実験システムを

- タブレット PC: ICONIA TAB A500-10S16
- OS: Android Honeycomb 3.1
- CPU: Tegra2 1GHz
- MEM: 1024 MB

上に Native Activity を用いた C 言語により実現した。Native Activity を用いた AndroidOS へのプログラム移植については本論文付録を参照されたい。

本実験システムでは操作者が 6 自由度の移動を行える移動容器の操作については、通常のドラッグ操作による 2 自由度の平行移動、2 本指のドラッグ操作によるスクリーンに垂直な平行移動、容器をタップすることで回転移動モードに変更した上で 3 章で提案した回転モデルによる操作とした。なお、本実験においては側面の描画や液面の波および色の拡散表現などの表現は行っていない。

4.2 実験の様子

実際に実験を行っている様子を図 4.1 に示す。容器の回転を行う際には図 4.2 のように 2 つの円が表示される。

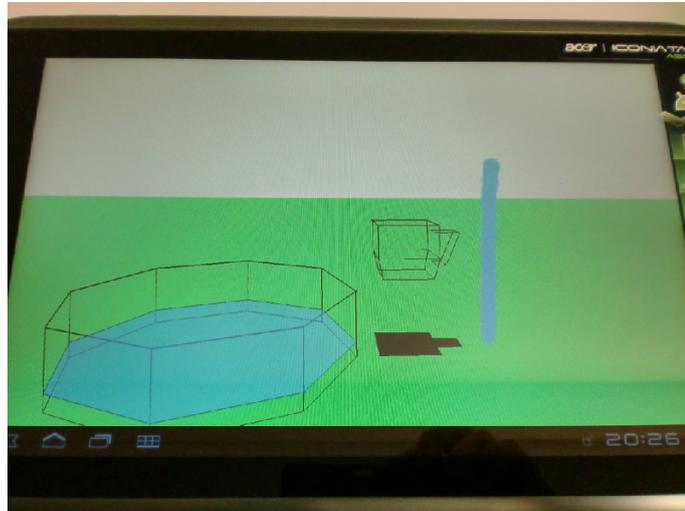


図 4.1: 実験の様子

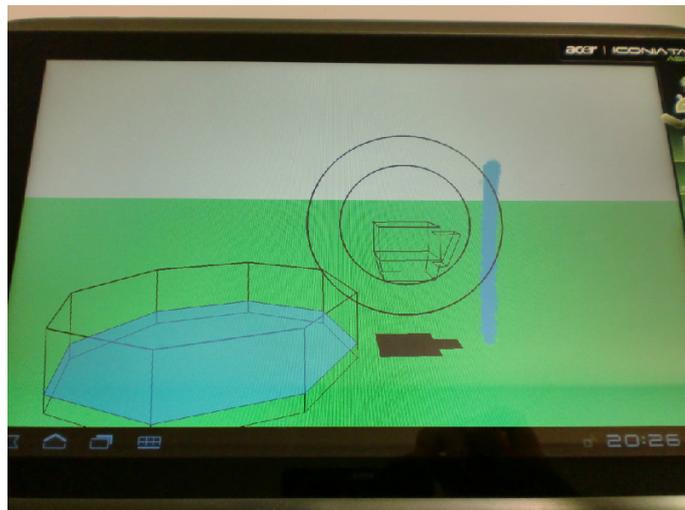


図 4.2: 容器の回転を行う際の画面表示

内側の円をなぞって容器の回転を行っている様子を図4.3,に示す.内側の円を斜めになぞる指の軌跡を描くことで,容器が斜めに傾いている.

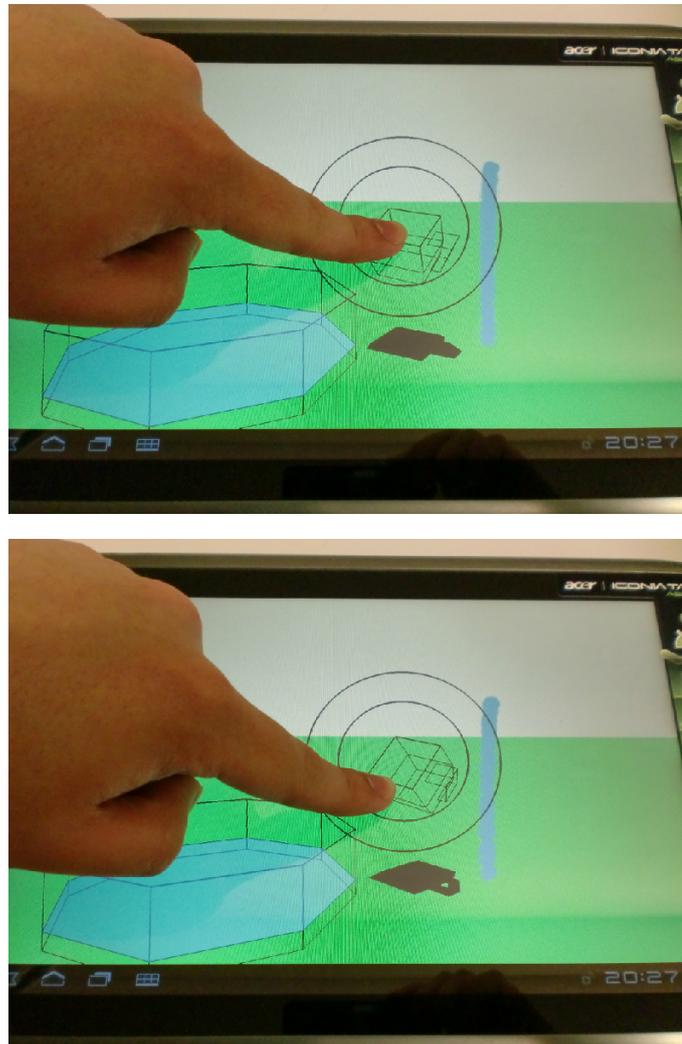


図 4.3: 内側の円内での容器の回転

次に内側の円と外側の円の間をなぞり容器がスクリーンに垂直な軸で回転を行っている様子を図 4.4 に示す．球体の周上をなぞり容器を回転させ，容器が横に傾いている．

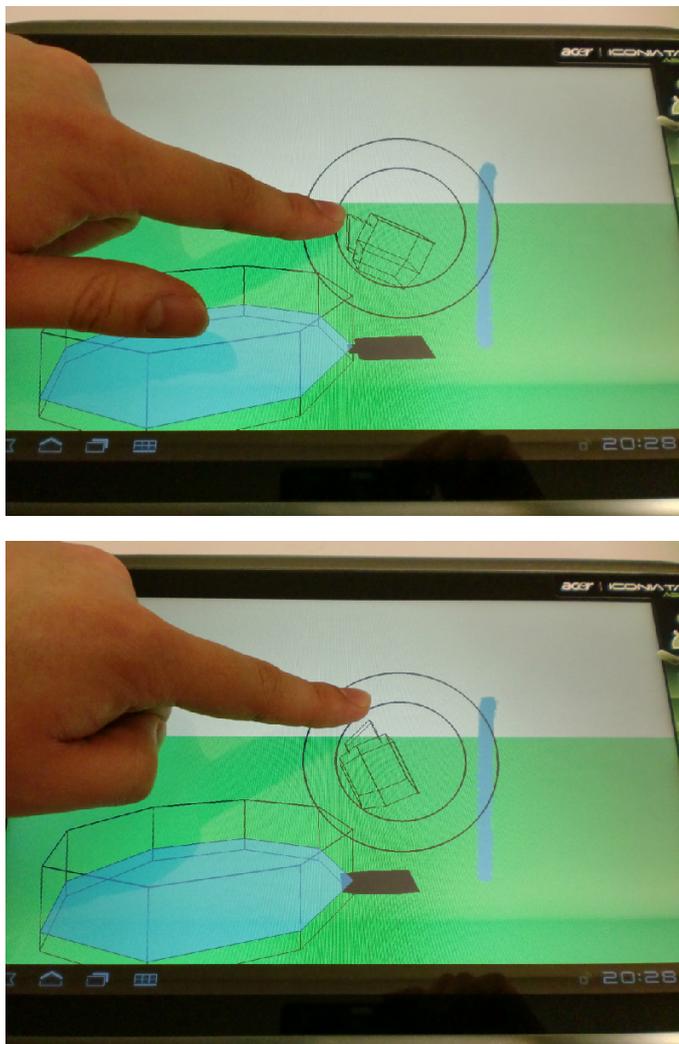


図 4.4: 円と円の間での容器の回転

また、実際に液体が入っている容器を傾けることで液体が容器外へ流出している様子が
図 4.5, 図 4.6 より確認できる。

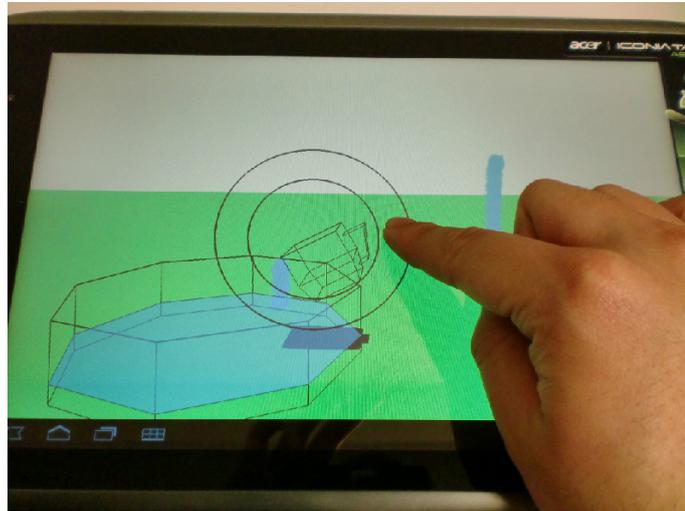


図 4.5: 容器を横に傾け液体を容器外へ流出させている様子

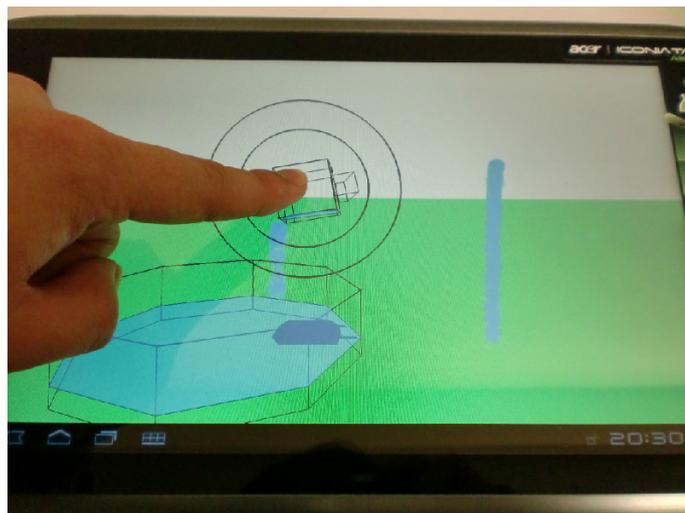


図 4.6: 容器を手前に傾け液体を容器外へ流出させている様子

4.3 典型的な2種類の回転インタフェースの操作感の評価実験

本研究では容器の3自由度回転については球体を回しているようなモデルを考え、はじめに3.1節で述べたように2つの典型的なインタフェースモデルについて考察した。本節では実際にこの2種類のインタフェースモデルについて20代男性9名に操作してもらい、操作感の感想を頂いた。以下にその感想と考察を述べる。

(1) 中心のみが固定された球の回転

- 円上部で横になぞった場合、容器が横に回転しない点について違和感。
- 横方向の回転以外は期待通りの回転ができた。

上述のように、3.1.2節で考えられた問題点が、実際に操作することでも発生することがわかった。特に多くの被験者が、地球を回しているかのようなイメージで容器の回転を行っていたとの感想とともに、違和感を抱いていたと述べていた。そのため、タッチパネル上での指の軌跡が横方向のみの場合には軌跡の場所に関わらず同じ回転を実装する必要があると考えられる。

(2) スクリーンに平行な軸での回転

- 容器を横に傾ける動作(スクリーンに垂直な軸での回転)ができないのに違和感がある。
- 思った通りに動かせない。

こちらのモデルについても、3.1.3節で考察した問題点が発生することが確認できた。また、多くの被験者が容器を手前に傾げるだけでなく、横に傾げることで液体を容器外へと流出させたいというイメージを持っていることがわかった。そのため、スクリーンに垂直な軸での回転は必要であると考えられる。

以上より、2つのインタフェースモデルで問題となっていた球体の回転を可能にした本研究提案モデルは、操作者の望む3自由度の回転を実現するにあたり有用であると考えられる。

4.4 地理座標系を取り入れた球体回転モデルの操作感の評価実験

本節では 3.2 節より実現される容器の回転操作の操作感を 20 代男性 9 名に 5 段階評価してもらった結果について述べる。5 段階評価の目安は以下の通りとした。評価の結果を図 4.7 に示す。

1: 全く期待通りに動かせなかった

2: ほとんど期待通りに動かせなかった

3: おおよそ期待通りに動かせた

4: ほぼ期待通りに動かせた

5: 完璧に期待通りに動かせた

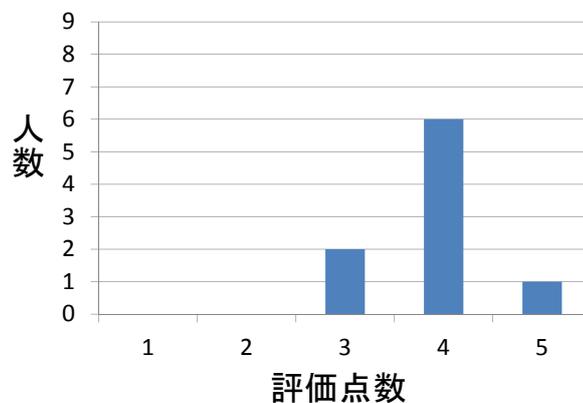


図 4.7: 容器の回転操作が期待通りにできたかの評価結果

5 段階評価の結果、9 人中 7 人から 4 点以上の評価を得ることができた。特に、3.1.2 節や 3.1.3 節で問題となっていた挙動を解決し、期待通りの回転が行えるようになったことで「直感的に操作できる」、「期待通りの回転ができ違和感がない」などの高い評価が得られた。これにより、本研究の提案モデルは 2 次元平面を介した 3 次元空間に対する操作として有用である。

しかしその一方で、「容器の移動と回転の切り替え方法や、容器の回転のイメージ(地球を回すような回転操作)を画面に表示させているとよい」、「指を 1 本だけでなく複数使っ

た操作があってもよい」, 「指で容器が隠れて見づらい場面があった」, 「内側の円での回転を始めた外側の円にはみ出しても内側の円での動きを続けてほしい」などの, 容器操作全体を含めたインタフェース面での意見を頂いた. これらの意見を参考に今後はより使いやすいインタフェースの実装を課題とし, インタフェースの改善を目指したい.

第5章 むすび

当研究室はこれまで VR 化学実験体験システムの実現を一つの目標とした粒子・体積に基づく仮想液体モデルを提案してきた。VR 化学実験体験システムの普及を目指すにあたり、移動の容易なデバイスでの動作の実現も必要である。そこで本論文では、2次元平面を介した3自由度回転のインタフェースモデルを提案した。これにより3次元モーションセンサーなどを利用しなくても、タッチパネルのような2次元平面を介した入力で操作者が希望する3次元物体の回転を実現することが可能となった。

VR 化学実験体験システムの普及には、今後の課題として次のようなものが挙げられる。

- 容器操作モデルのマルチタッチインタフェースへの拡張

一般に普及しているタブレット PC は複数のタッチを検出可能である。そこで、複数の指を用いることでより操作性の高いインタフェースへ改良したい。

- タブレット PC の各種センサーを用いたインタフェースの考案

タブレット PC にはマルチタッチスクリーンだけでなく、加速度センサーやジャイロセンサーといった機能も搭載されている。そこで、指で触って操作するだけでなくタブレット PC そのものを動かすことにより操作するインタフェースを提案し、より操作性を高めたい。

- 実際の教育現場での評価実験

実際の教育現場でタブレット PC 上で実現された VR 化学実験体験システムを使用してもらい、小学生程度を対象とした場合にはどのようなインタフェースが有効であるかの評価実験を行いたい。

これらの課題を含め、様々な VR システムの実現、普及につなげていきたい。

付録

本付録では，C 言語を用いて作成された OpenGL プログラムのタブレット PC(android OS) へのプログラム移植時における問題点とその解決方法について簡単に説明する．

A.1 Native Activity について

本来，Android の開発環境は Java 言語が想定されており，C 言語を用いたプログラムをそのまま流用するのは困難である．そこで主に利用されるのが Android NDK(Native Development Kit) である．この Native Development Kit の名前通り，実効用コードを Java 中間コードではなくネイティブコード (機械語) とするものである．Android NDK を用いることで，今まで Java で記述することでしか呼び出せなかった機能を C/C++ のコードから直接呼び出し，Android アプリケーションの一部を C/C++ で作成することが可能となる．しかし，このままではアプリケーション画面の表示，タッチパネルなどの入力のようなアプリケーションの基本となる動作については Activity クラス，つまり Java コードを使う必要性があり，既存の C 言語のみによるシステムをそのまま移植するのは困難となる．この場合に利用されるのが Native Activity である．Native Activity を用いることにより，Activity が C/C++ からでも作成出来るようになり，C 言語ベースのプログラムの Android 移植を行う際には適している．Native Activity は Android 2.3 より提供され Android NDK r5 より利用できる [22] ．

A.2 OpenGL と OpenGL ES について

OpenGL は UNIX 系 OS や Mac OS X , Windows 等で使用できる 3D グラフィックスライブラリであり，これを組み込み機器向けに必要な機能に絞り込んだものが OpenGL ES (Embedded System) である．そのため，OpenGL で良く利用される GLU や GLUT に対応する機能がなく別途対応する必要がある．特に GLUT に関してはウインドウの制御やバッ

ファスワップなど環境依存の操作を行っているため、OpenGLES でも同じような機能を使用する必要がある。そこで環境依存部分を切り出した EGL インタフェースを用いる。EGL インタフェースではウィンドウサーフェイスの生成、画面の初期化、ダブルバッファの切り替えなどを行なうことができる。GLU の利用は mesaGL より必要な関数のソースコードを用いることで実現が可能である [23]。

A.3 OpenGLES のためのプログラム改変

本節では OpenGL から OpenGLES への移植を行なった際、書き換えが必要になった箇所の問題点と解決方法について述べる。

A.3.1 GL_DOUBLE 型の削除

OpenGLES では GL_DOUBLE 型が削除され、GL_FLOAT 型のみとなる。そのため、移植を行うプログラムにおいて GL_DOUBLE 型を用いた OpenGL の描画処理をしていた場合、GL_FLOAT 型を用いた変数などに変更する。

A.3.2 glMaterialfv の仕様

OpenGL では物体の材質 (拡散反射係数や鏡面反射係数) は前面と裏面に異なるものを設定することができた。しかし、OpenGLES では両方の面に同じものを設定する必要がある。具体的には glMaterialfv の引数において GL_FRONT, GL_BACK, が使用できなくなり、GL_FRONT_AND_BACK のみが使用できる。

A.3.3 物体描画方法の変更

OpenGL では glBegin, glEnd を用いて描画する物体の頂点 1 つ 1 つ指定していく方法が可能であった。OpenGLES ではこの機能が削除され、頂点を 1 つの配列にまとめ、glVertexPointer でまとめて描画する。また、OpenGL では図形の描画を行う際に GL_QUADS や GL_POLYGON のように四角形や多角形を指定できる。しかし、OpenGLES では多角形の描画は三角形のみに限定される。そのため、多角形の描画を行う際には GL_TRIANGLE_FAN を用いて、三角形の集合として描画する。

以上のようなプログラムの改変を行なうことで Android への OpenGL プログラムの移植が可能である。

謝辞

本研究を進めるにあたって、日頃から多大な御尽力を頂き、ご指導を賜りました名古屋工業大学、舟橋健司 准教授、山本大介 准教授、伊藤宏隆 助教、に心から感謝致します。また、本研究に対して御討論頂きました本学 中村研究室の皆様ならびに中部大学 岩堀研究室の皆様へ深く感謝致します。最後に、本研究に多大な御協力頂きました舟橋研究室諸氏に心から感謝致します。

参考文献

- [1] 舘暲, 佐藤誠, 廣瀬通孝, 日本バーチャルリアリティ学会, “バーチャルリアリティ学”, 工業調査会, 東京, 2010 .
- [2] 富川盛雅, 橋爪誠, “バーチャルリアリティシミュレータを用いた内視鏡外科手術トレーニングシステム”, 日本外科学会雑誌, Vol.112, No.4, pp.255-261, 2011 .
- [3] 濱田友貴, 鍵山善之, 黒田嘉弘, 井村政孝, 大城理, “回転骨切り術シミュレータにおける弾塑性体衝撃破壊の力覚提示法”, 映像情報メディア学会技術報告, Vol.34, No.25, pp47-52, 2010 .
- [4] 須賀晶昭, 山田常圭, 阿部伸之, “VR 技術を用いた火災体験シミュレータ”fire cube”の開発”, フジタ技術研究報告, No.42, pp43-48, 2006 .
- [5] 片柳亮二, “AR・VR 航空機開発に必須なフライトシミュレータ”, 画像ラボ, Vol.23, No.3, pp30-37, 2012 .
- [6] 中林功一, 伊藤基之, 鬼頭修己, “流体力学の基礎 (1)”, コロナ社, 1993
- [7] Y. Natsume, A. Lindroos, H. Itoh and K. Funahashi, “The Virtual Chemical Laboratory Using Particle and Volume Based Liquid Model”, Proc. SCIS & ISIS 2010, pp.1354-1359, 2010.
- [8] M.Muller, S.Schirm, M.Teschner, “Interactive Blood Simulation for Virtual Surgery Based on Smoothed Particle Hydrodynamics”, ACM TechNology and Health Care, Vol.12, No.1, pp.25-31, 2004.
- [9] Oh-young Song, Doyub Kim, Hyeong-Seok Ko, “Derivative Particles for Simulating Detailed Movements of Fluids”, IEEE Transactions on Visualization and Computer Graphics, Vol.13, No4, pp711-719, 2007.

- [10] Jun Chen, Kejian Yang, Yuan Yuan, “SPH-based visual simulation of fluid”, The 4th International Conference on Computer Science and Education(ICCSE 2009), pp690-693, Nanning, Chine, 2009.
- [11] C. Lacoursiere, M. Servin, K. Bodin, “Constraint Fluids”, IEEE Transactions on Visualization and Computer Graphics, Vol.18, No.3, pp.516-526, 2012
- [12] 舟橋健司, 岩堀裕之, “仮想容器による仮想液体の対話操作モデルと一実現方法”, 日本バーチャルリアリティ学会論文誌, Vol.5, No.4, pp.1087-1094, 2000 .
- [13] K. FUNAHASHI, Y. IWAHORI, “Representation of Swinging Liquid on Virtual Liquid Manipulation”, Proc. ICAT2001, pp.179-184, 2001.
- [14] 長谷川聡, 佐原理, 長谷川旭, 田川隆博, 尾崎志津子, “タブレット端末の教育利用 - 名古屋文理大学における iPad 導入-”, ヒューマンインタフェース学会誌, Vol.12, No.4, pp.245-252, 2010.
- [15] 寺川晃司, 浦正広, 中貴俊, 山田雅之, 遠藤守, 宮崎慎也, 安田孝美, “テーブルトップを用いた三次元身体モデルの操作インタフェースの提案”, Vol.34, No.25, pp.63-66, 2010 .
- [16] blender.org <http://www.blender.org/>
- [17] 夏目祐樹, 伊藤宏隆, 舟橋健司, “粒子・体積ベース仮想液体操作モデルにおける凹形状容器での液体操作の実現”, 日本バーチャルリアリティ学会第 15 回大会講演論文集, 3C1-6 (DVD-ROM), 2010 .
- [18] T. Tanabashi, H. Itoh, K. Funahashi and Y. Iwahori, “Representation of Wave Surface on Virtual Water Manipulation”, SCIS & ISIS 2008, pp.1460-1465, 2008.
- [19] 棚橋拓也, “対話性を重視した仮想液体操作モデルにおけるかき混ぜ動作の実現”, 平成 21 年度名古屋工業大学修士論文, 2010.
- [20] 蒲将寛, “粒子・体積モデルに基づく仮想液体操作システムにおける色の表現”, 平成 20 年度名古屋工業大学卒業研究論文, 2009.

- [21] 菅瀬和弘, “VR 化学実験システムのための仮想液体モデルにおける側面描画”, 平成 21 年度名古屋工業大学卒業研究論文, 2010.
- [22] 出村成和, Android NDK ネイティブプログラミング, 秀和システム, 東京, 2011.
- [23] The Mesa 3D Graphics Library <http://www.mesa3d.org/>