

# 3 D表現における視点移動

## 1 . はじめに

この研究では、ハードウェアの進歩により使われる機会の多くなった3D表現における視点移動の方法について考えます。ここで対象となる3D空間を地面で区切られた半空間に限定することで、より具体性を持ったものを提案します。入力機器としては既に広く普及しているマウス、キーボードを想定します。

本概要では、既存の操作方法への考察及び新たな操作方法の提案をし、そしてそれを実装する3Dエンジンの作成について述べます。

最終的には新たな操作方法を実装した3Dエンジンにより、それを評価します

## 2 . 操作方法の考察

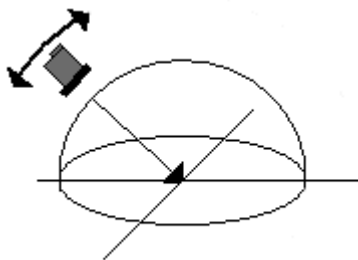
### 2.1 操作方法の定義

計算機上に再現された3D空間に対して我々が行う操作は大きく2種類あります。1つ目が3D空間を撮影しているカメラの操作、そしてもう1つが3D空間内にある物体の操作です。ここではカメラの移動操作を操作方法と定義します。

### 2.2 既存の操作方法

現在さまざまな目的に応じて3D表現を利用したアプリケーションがありますが、それらの操作方法を視点について分析し幾つかの種類に分類します。

#### [周囲回転視点]



[図1:周囲回転の動き]

特定の地点を中心とした球面上をカメラが

移動する。視線の方向は常にその球の中心を向いている。見方によっては、視点を固定して3D空間を回転させているとも取れる。

(長所)縦横方向2つの入力で済むため操作が容易。

(短所)定められた球の内部にある空間しか見ることが出来ない。

#### [平行移動視点]

カメラが3D空間に予め定められた絶対的な座標軸に沿って移動する。回転動作はカメラの位置を中心に行う。

(長所)3D空間の絶対的な座標軸に従って移動するため、正確な移動ができる。

(短所)移動方向と向いている方向が異なる場合、操作感が悪い。

#### [自由視点]

空間内を、カメラの座標系を中心とした向きでそのカメラが移動する。移動方向は前後進のみで、向きは回転によって変える。

(長所)人間が実際に移動する感覚に近いので、分かりやすい。

(短所)基準であるカメラの位置や傾きが3D空間に対して常に変化するため、3D空間内の絶対的な位置関係を見失いやすい。

既存のアプリケーションでは、これらのうち複数のものを共用している場合が多いです。その結果、使用ボタンの増加や視点のモード切替などが必要となります。

### 2.3 周囲回転視点の拡張

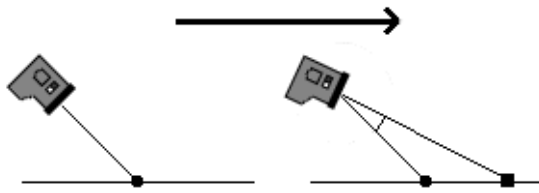
操作方はアプリケーションの目的に応じてさまざまな物が要求されます。一般的に可能な操作を増やすと難しくなり、シンプルにすると出来る操作が少なくなります。

3D表現は期待動作と実際の動作、すなわち入力と出力の結果が食い違くと、混乱を引き起こします。そのため、直感的であることに重点をおく必要があります。

そこで注目したいのが周囲回転視点です。前項にあるように少ない入力情報で操作が出来るため直感的な操作が期待できます。これを球の外の開空間にも適用できるように

拡張をします。

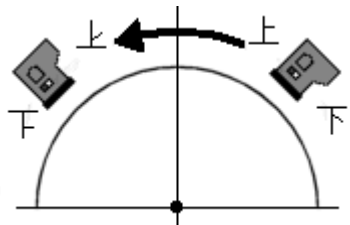
#### [中心点の移動]



[図2:中心点の移動]

回転の中心点(図中の点)を移動しないことにはその球面上以外へ行くことが出来ません。また空間を均一に描写するため中心点は画面中央、すなわち視線上に設定します。ここで新たな中心点を見つけるには視点を中心に回転をする必要があります。そして中心点の指定は目的の点を画面中央に誘導することで行います。視線と地面の交点が新しい中心点(図中の点)となります。

#### [反転防止]



[図3:上下反転]

冒頭でも述べたように、今回の研究では対象となる空間を地面で区切られた半空間に限定しています。そこでは明確な上下の概念が生まれます。そのようななか、中心点の向かい側に行くために前方向へ進んでいくと、地面と視線が垂直になる地点から上下が反転してしまいます。これは本来地面と平行に回転すべきものを行っていないために起こる事象です。それを防止するために、地面と視線が垂直になる地点の通過を禁止します。

### 3. 3Dエンジン

前述の操作方法を実装するための3Dエンジンを作成します。特に、

- カメラの周囲回転移動
- 視線と地面の交点から新しい中心点を求める。

という2点が必要です。ちなみに、この3Dエンジン自体は半空間という性質に依存していません。

まずカメラの移動についてですが、半球上

を移動する動きは視線と地面の角度、及び地面上の1本の軸との角度によって決定されます。従って中心点を原点とする座標系を使い、その角度によるカメラの回転処理を行います。

続いて新しい中心点の設定についてですが、Zバッファ法の仕組みを利用します。Zバッファ法とは画面上の各ピクセルで描画される最も手前にあるポリゴンのZ値(奥行き)を格納するものです。ここに地面のZ値を格納すれば、画面中央のピクセルに位置する地面の座標値が判明します。画面への描写は透視投影で行われているので、Z値より他のXY値(縦横)も求められます。

### 4. 結論

最後に完成した3Dエンジンを使い、提案された操作方法の評価を行います。評価基準を3D空間における視点移動に絞ると、

(1) 操作の難易度

(2) 操作にかかる手間

(3) 予想外の動作を引き起こす確率

が適当です。順に見ていくと、まず(1)は2.3項の通りです。ただし中心点による回転と視点による回転を切り替える感覚を学習するのに、多少時間を要すると考えられます。

(2)は操作内容によります。移動は角度を操作して行うため、中心点とカメラとの距離に関係なく一定の入力で済みます。しかし中心点の向かい側に同じ方向のまま移動したい場合などは、無駄な回転が加わり面倒です。(3)は、確率そのものは抑えることが出来ませんが、移動中の移動可能範囲が球面上に限定されるため、予想外の動きをしても修正が容易です。

これらのことより、提案された視点操作方は、広い空間内を手軽な操作で一望出来ますが、特定の地点で特定の方向を向くと言った操作は少々回りくどくなってしまいます。従ってこれは本格的な操作を必要としない3D空間を観察するタイプのアプリケーションに適していると言えます。具体的には手軽にweb上などで、入り組んだ建造物モデルのように一点中心の周囲回転では不十分なものの活用できる操作方法だと考えます。

(参考文献) ユーザインターフェースの設計  
Bシュナイダーマン著<日経BPセンター>