

## 3 点タッチを利用した 3 次元仮想空間内視点移動手法の検討 A Method for Free Viewpoint Navigation in 3D Virtual Environments using 3-finger Tap

王 宏宇<sup>†</sup>      梅澤 猛<sup>‡</sup>      大澤 範高<sup>‡</sup>  
Hongyu WANG      Takeshi Umezawa      Noritaka Osawa

### 1. はじめに

人々は 3 次元の実世界を自由に移動し、3 次元オブジェクトを操作し、3 次元空間を特に意識しなくても自由に暮らしている。私たちはコンピュータを利用して 3 次元仮想環境を構築し、その中で 3 次元物体を操作したり、視点の位置や方向を変更するナビゲーションをしたりすることができる。通常、3 次元仮想環境は、スクリーンから見る事ができるシーンより広い空間を含み、ユーザーが仮想環境を把握するためには、環境内で移動や回転などによるナビゲーションの必要がある。しかし、2 次元のマルチタッチスクリーンにおいて操作が容易で効率的な手法は十分研究されていない。そこで、3 次元仮想環境と 3 指を活用したナビゲーション手法を提案する。

### 2. 関連研究

シーン内のオブジェクト上の関心のある点(Point of Interest,POI)[1]を目標として選択し、視点をこの目標に向かって移動する手法がある。その手法の一つとして Navidget[2]が高速で容易なカメラポジショニングを可能とする。ペンやマウスで POI を選択して、球形のウィジェットの表面に視点を指定して視線の方向は視点から球の中心に向く。プレビューウィンドウと仮想ウィジェットによるフィードバックによって、ユーザーは自由に視線方向を制御できる。しかし、この手法は POI 選択後に視点選択のモードがあり、マルチタッチを活用していない。

タブレットなどにおける 3 点タッチの一般的な方法は、非利き手の 2 本の指でオブジェクトをピン止めし、利き手の指の一本を使用し、オブジェクトをスイングする。つまり、非利き手の 2 つのタッチ点を通る直線により回転軸を定義し、その軸の周りにオブジェクトを回転させることができる。しかし、利き手の指が左右にスワイプする時、回転の向きを指定できない。

### 3. 提案手法

#### 3.1 手法 1

タッチスクリーンにおける 3 点タッチを通して、3 次元仮想空間で視点移動をする手法を提案する。仮想空間内の 3 点を指定して、それらを頂点とする三角形を構成し、その三角形が視野に大きく含まれ、三角形の法線方向を向いた視点に移動する。3 点が構成する三角形によって ROI(Region of Interest)を指定する。たとえば、図 1 に示すように、ある立方体の表面の視野に含まれるべき点を 3 本の指(一般には人差し指、中指、薬指)で順にタッチする。タッチの順番に 3 点を A、B、C とし、それらを頂点とする。

<sup>†</sup> 千葉大学工学部

<sup>‡</sup> 千葉大学大学院融合科学研究科

三角形 $\Delta ABC$ を考える。 $\Delta ABC$ の法線ベクトル $\vec{n}$ とし、重心 $G$ とする。視点 $V$ は、重心 $G$ を通る法線上で視線が重心 $G$ を向き、A、B、C が視野の外周付近に来るようにする。図 1 の場合には $\vec{VG}$ が視線の向きとなる。視線の方向は 3 点タッチの順番によって決まり、背面に視点を移動することが可能である。

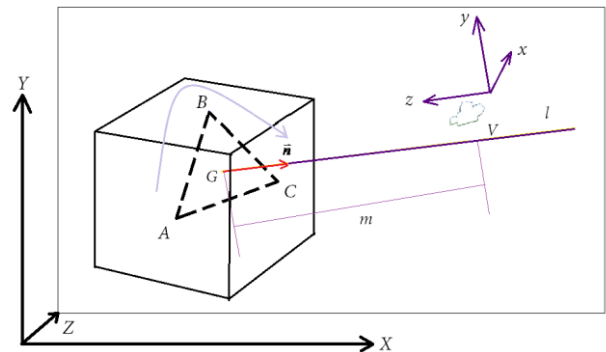


図 1: 手法の原理

#### 3.2 手法 2

3.1 項で提案した手法 1 において ROI は三角形 $\Delta ABC$ によって示す。手法 2 は 3 点で ROI を指定するのではなく、回転中心と回転角を制御するために 3 点タッチを利用する。ROI を手法 1 によって指定するためには、ROI が見えていなければならない。ROI が見えていない場面で視点を容易に移動するための手法を提案する。中指のタッチによって指定される仮想空間内の点 B を回転中心とし、点 A と点 C によって回転角を制御する。視線の向きは図 2 の $\vec{VB}$ にする。点 A と点 C をシーン内の床や他の物体上に指定する。

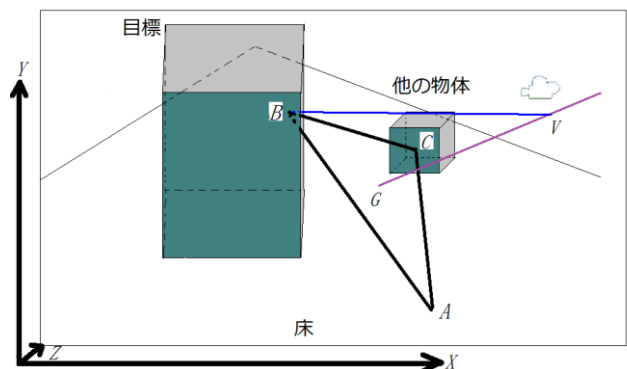
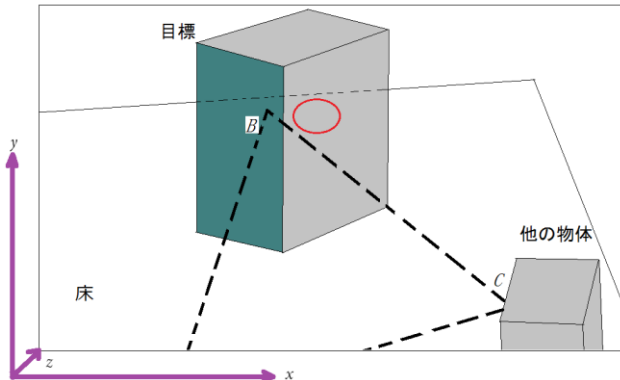


図 2: (a) 視点移動前

図 2 の例において図 2 (b)の赤丸を目標の POI として、図 2 (a)のように現在の視点からは POI を見ることができないとする。図 2(a)に示すように目標の正面(青緑)上に点 B を定め、点 A と点 C を他の物体や地平面上で移動させることで視点を点 V に移動させる。点 B によって回転中心を選択し、点 A と点 C の中点が回転角を制御し、点 A と点 C の間の距離

によってカメラのズームを制御する。なお、ユーザーが理解しやすいように、点Bの中指を放すと視点の移動が行われるとする。視点移動後の画面は図 2(b)のようになる。黒い破線は移動前の三角形である。手法 1 では 3 点は必ず視野に含まれるが、手法 2 では指定した点が視野外になることもある。図 2(b)の場合には点Aが視野外になっている。



(b) 視点移動後  
図 2: 視点移動例

### 3.3 プレビュー

操作後の状態を理解しやすくするためにプレビューが必要である。3 点タッチをする際に画面の右上にプレビューウィンドウを表示する(図 4)。もしも、この小さいウィンドウで理解しづらい場合には、四番目の指(小指など)をスクリーンにタッチするとプレビューが全画面に拡大される(図 3, 黒い四角形はプレビューと普通の画面を区別するため)、指を放すと全画面プレビューが消える。

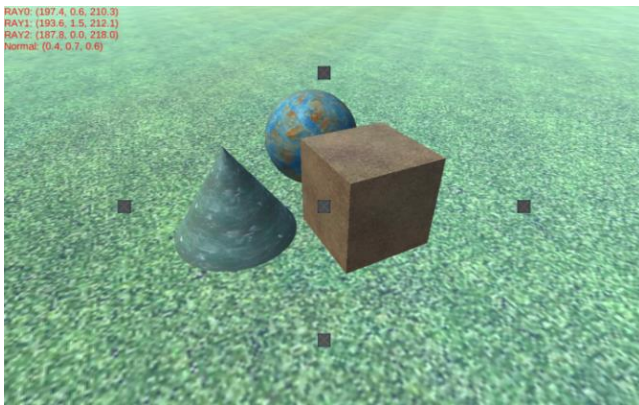


図 3: タスク 1 全画面のプレビュー

### 4. 評価

Unity によって仮想空間を構築し実験環境とする。被験者は指定の領域を視野に収める視点移動の効率を評価するタスクと実際のシーン内をナビゲーションをするタスクの 2 種類のタスクに参加し、質問紙調査に回答する。タスクの完了時間および操作回数を測定し、分析する。操作手法としては提案手法 1、提案手法 2、2 点タッチまでを利用する既存手法を比較する。

タスク 1(図 3)では物体の表面上のマークが、画面上の指定された範囲内に表示されるように視点を移動させる。こ

のタスクの目的は製品などのデザインを見るような基礎的なシーンで提案手法が高速で容易な回転ができるのかを判断する。タスク 2(図 4)では広い町のような空間内に設置されたボタンを指定された順番にタッチする。3 次元仮想環境において物や場所を探すためのナビゲーションをする際、提案手法で効率的な操作が可能かを評価する。

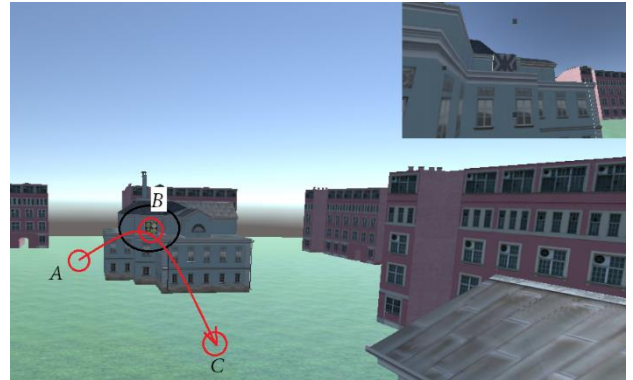


図 4: (キャプチャーの一部) タスク 2 遠い物体の斜めを見たい操作

### 5. おわりに

本研究では 3 点タッチにより、3 次元仮想空間の地形や物体を利用して視点を移動するナビゲーション手法を提案した。今後評価を行い、さらなる改善手法を検討していく予定である。

#### 参考文献

- [1] Jock D. Mackinlay, Stuart K. Card, George G. Robertson, Rapid controlled movement through a virtual 3D workspace. ACM SIGGRAPH Computer Graphics, Vol. 24, No. 4, pp. 171-176(1990)
- [2] Martin Hachet, Fabrice Declé, Sebastian Knödel, Pascal Guittou, Navidget for 3D interaction: Camera positioning and further uses. International Journal of Human Computer Studies. Vol. 67, Issue 3, pp. 225-236 (2009)