

3次元 GUI におけるウォークスルーに関する一考察

7C-5

杉山 達彦 及川 昌孝 李 殷碩 白鳥 則郎

東北大学電気通信研究所

1 はじめに

計算機内に定義された3次元オブジェクトを直接操作するインタフェースである、3次元 GUI を利用したシステムを効率良く作成するための支援環境の研究を行っており、その基礎となる3次元ウィジェット群、3次元イベント制御機構をもつツールを提案してきた[1].

本稿では、3次元イベント制御機構の一つであるウォークスルーについて考察し、特にウォークスルー処理の際に考慮すべき制約について述べる。ここで、ウォークスルーとはコンピュータ内に定義された仮想空間(CGオブジェクト間)を自由に動き回ることである。

2 ウォークスルーについて

CG を利用したシステムは、通常 PHIGS, PEX, OpenGL といったグラフィクスライブラリを用いて作成される。しかし、ライブラリは座標変換やオブジェクトの定義などの低レベルの機能を提供しており、ウォークスルー機構は開発者の負荷として委ねられている。そのため、ウォークスルー機構をシステムにとり入れるなら、開発者は独自に必要なメカニズムを考案し、実装しなければならない。

ウォークスルーを実現するための基本的なメカニズムとは、視点の位置を変えてそこから見えるものを再描画するというプロセスの繰り返しである。すなわちウォークスルー処理機構は、ユーザからの操作を受け次の視点の位置を計算し、再描画を指示するための役割をもつ。

本研究で実現しているウォークスルー処理機構では、次の視点の位置を計算するプロセスの中で、CG を構成するオブジェクトを、2種類—通り抜けられるものと通り抜けられないもの—に分け、ウォークスルー処理の最適化を図る。これにより、3次元 CG 空間での

ユーザのインタラクションに対し、より人間にとって直感的で自然な応答が実現できると考えている。そのためにウォークスルーは、通り抜けられないもので止まるべきである。すなわち、ウォークスルーの進路上に通り抜けられないオブジェクトがあるかどうかをチェックしたあとで、次の視点の位置を指示しなければならない。

そのために、ウォークスルー処理機構を CG システムに組み込む際には、CG 空間内の各オブジェクトの制約情報を保持するウォークスルーの制約機構も併せ持つ方法をとる。しかし、従来は制約機構はあまり実装されていない。これは、CG システムを高機能にするほどウォークスルーの制約機構は複雑になり、開発者の負荷が増大するためだと思われる。開発者は、通常 CG システムがいろいろな形のオブジェクトを扱えるように設計する。一方、オブジェクトの幾何形状が複雑であればあるほど、ウォークスルーの進路と各オブジェクトとが緩衝しているかどうか、というチェックが複雑になってしまう。

3 従来のウォークスルー制約手法

従来の制約手法の場合は、ウォークスルーの進路上に通り抜けられないオブジェクトが存在するかどうかをオブジェクトの幾何形状をもとにチェックする、という手法で行なわれている。このチェックはウォークスルーのたびに行なわれる。

この手法の欠点は、オブジェクトの幾何形状が複雑であるほどチェックに必要な処理が複雑となり、オブジェクトの数が多いほどチェックに必要な計算量が多くなることである。この影響で、CG が複雑になるとウォークスルーの実行速度が低下してしまう。

4 提案手法

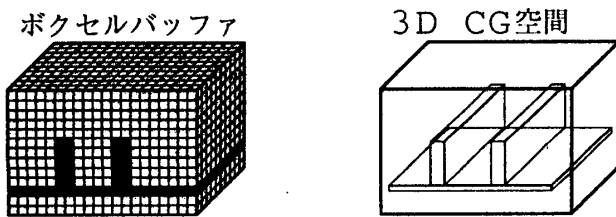
4.1 基本概念

CG 空間内のオブジェクトは大きくつぎの2種類に分類される。ユーザとのインタラクション中にアニメー

A Consideration on a Walkthrough in 3D GUI
Tatsuhiko SUGIYAMA, Masataka OIKAWA, Eun-Seok LEE and Norio SHIRATORI
Research Institute of Electrical Communication, Tohoku University

ションをし、その幾何形状や位置を変化させるものと、インタラクション中ずっと形状や位置を変化させないものである。これらをそれぞれ動的オブジェクト、静的オブジェクトと呼ぶ。

提案手法は動的オブジェクトと静的オブジェクトとによる制約を区別しておこなう。具体的には、動的なものは従来のように、オブジェクトの幾何形状や位置を調べ、ウォークスルーごとのチェック(複雑な処理)を行なう。一方、静的なものに関しては、CG空間と対応したボクセルバッファを導入し、ウォークスルーごとのチェック(簡単な処理)を行なう。図1にボクセルバッファのイメージ図を示す。



- オブジェクトが存在するCG空間内座標に対応するセルに1をセット(ウォークスルー前)
- 進路のバッファが1かどうかでウォークスルーを制約(ウォークスルー時)

図1 ボクセルバッファのイメージ図

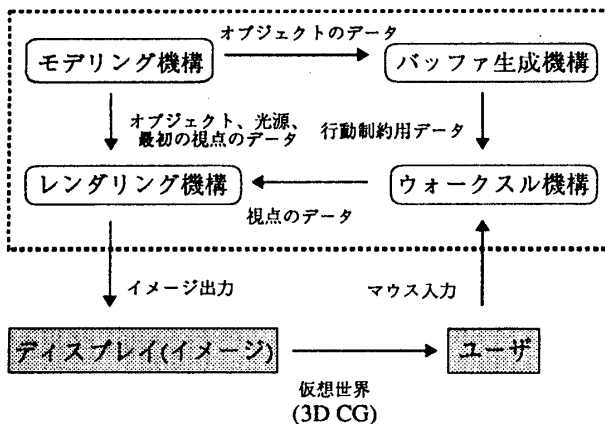


図2 システム構成

図2に本提案手法に基づくシステム構成図を示す。図2の各要素について簡単に説明する。

- (1) モデリング機構はオブジェクトの形状・属性・位置や、光源の属性・位置や、最初の視点の位置などの

CGの描画に必要な情報を内部に保持するところである。

- (2) レンダリング機構はオブジェクトや光源や視点の情報から、視点からオブジェクトがどのように見えるかを計算し表示するところである。
- (3) バッファ生成機構はオブジェクトの形状、位置の情報から、対応するボクセルバッファのセルを計算し、そのセルに1をセットするところである。
- (4) ウォークスルー機構は、ユーザからのインタラクションとボクセルバッファに基づいて、次の視点を生成するところである。

4.2 応用と評価

本手法を評価するために前述の構成図に基づく実験システムの構築を行なった。そのシステムは、静的なオブジェクトで構成されたCGを提供し、その中でのウォークスルー機構を提供する。

この実験を通して得られた定性的な評価は、ボクセルバッファの導入により、静的なオブジェクトに対し、そのデータ量増加に伴うウォークスルー時の速度低下の解消を行なえることである。ここでデータ量は、オブジェクトの数と、オブジェクトの幾何形状の複雑さを意味する。

5 おわりに

今後、本提案手法の定量的な評価を行ないたい。また、ウォークスルー機構を3次元GUI構築用ツールキットに組み込むために、ツールキット利用者の負担を軽くするための仕組みなどに関する検討が残されている。

参考文献

- [1] 及川 昌孝, 李 殷碩, 白鳥 則郎, “3次元GUI構築支援環境に関する研究”, 電子情報通信学会第49回秋季全国大会, 1994
- [2] Marc P. Stevens, Robert C. Zeleznik, John F. Hughes, “An Architecture for an Extensible 3D Interface Toolkit”, In *Proceedings of UIST '94, ACM SIGGRAPH*, November 1994