

## 球による高速な衝突判定の一手法

5 E - 7 - 3

藤井 宏道 小堀 研一 西川 禎一  
大阪工業大学

## 1. はじめに

仮想空間ウォークスルーシステムにおいては、現実の世界と同様に、動的オブジェクトである3次元物体同士の衝突を考慮する必要があり、また実時間で処理する必要がある。

一般に、3次元物体同士の衝突判定は物体形状を構成する面などを用いて交差チェックを行う。そのため、形状が複雑になると計算コストが急激に増大し、実時間で衝突の有無を検出することが困難となる。

そこで、本研究では球を衝突判定に用いることを考えた。球同士の衝突の有無は、衝突する関係にある球同士の中心間の距離と球同士の半径の和を比較するだけで容易に検出することが可能である。それにより演算量を大幅に削減することが可能である。また、球同士の判定においては、あらゆる角度から衝突した場合でも計算コストは一定であり、動的オブジェクトがさまざまな角度で衝突する可能性が十分に考えられるウォークスルーシステムにおいては有効であると考えられる。

本手法では、物体内部に大きさの異なる複数の球を充填し、形状を球群により近似する。現在、大きさの異なる複数の球を用いて形状を近似する手法として、ポロノイ領域を用いた手法<sup>[1]</sup>や8分木を用いた手法<sup>[2]</sup>が提案されている。これらの手法は、形状の近似誤差が大きいという問題や球の大きさが小さいほど、球の数が大幅に増大し、その計算コストも同様に増大する問題がある。よって実時間での処理を行うためには、形状内部に充填する球の数をできるだけ少なくし、かつ形状との近似誤差を低く抑えることが重要であると考えられる。

本手法では形状表面に接する大きさの異なる球を、球同士の重複を許す条件で形状の内部に充填する。そして、これらの充填球群に対して外包球を用いて8分木を構成する。

以上の処理を前処理とし、ウォークスルーシステムにおいて、動的オブジェクト各々に対して求めた8分木同士での交差判定を行うことにより実時間での衝突判定を実現する。

## 2. 形状の面を用いた衝突判定

形状の面を直接用いて衝突判定を行う場合、2つの物体に対して以下の順序に従って行う。

- ① 形状全体を外包する直方体を用いて直方体同士の交差判定を行う。
- ② 手順①において交差する場合、形状を構成する三角形ポリゴンの外包直方体を考え、互いの形状を構成する全ての面に対して総当りにより直方体同士の交差判定を行う。
- ③ さらに、手順②において交差する場合、交差した直方体の三角形ポリゴン同士の交差判定<sup>[3]</sup>を行う。
- ④ 手順③において交差している場合、2つの物体は衝突していると判定する。

また、手順①、②、③において非交差の場合、2つの物体は衝突していないと判定する。

## 3. 提案手法

本手法の処理の概要を以下に示す。

1. 形状の内部に形状を近似する大きさの異なる球群を充填する。
2. 充填球を外包する球を用いて、外包球の8分木を生成する。
3. 8分木を用いて衝突判定を行う。

本稿では3次元形状を対象にしているが、説明は簡単のため2次元で行う。

### 3.1 充填球の生成

本手法では球の中心座標を求めため、形状データを境界表現モデルから空間分割モデル<sup>[4]</sup>である2値(白と黒)画像に変換する。このとき、球を充填することが可能な領域を黒画素、球を充填することが不可能な領域を白画素とする。そのため、最初に形状の内部を黒画素、外部を白画素で初期化する。

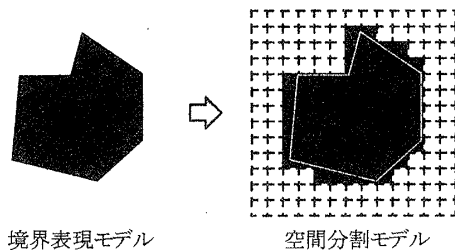


図1 データ変換

次に、形状の内部に充填すべき球(充填球)を生成するための手順の概要を示す。

#### 1. 充填候補球の生成

形状の画像の黒画素に対して、ランレンクス<sup>[5]</sup>を求め、充填球の候補の中心座標と半径を求める。

#### 2. 充填候補球の内外判定

円を高速に描画するミッチェナーのアルゴリズムを用いて、手順1で求めた充填候補球と形状の画像との内外判定を行い、内部と判定された場合に充填球として採用する。

#### 3. 白画素による塗りつぶし処理

手順2において採用された球の内部にある形状の画像を白画素にする。そして、手順3によって更新された画像を形状の画像として、再び手順1を行う。

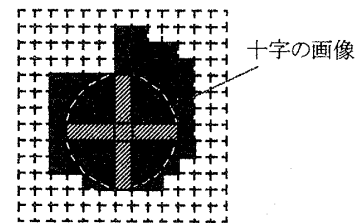
以上の処理を、手順3において形状の画像が全て白画素になるまで繰り返す。

#### 3.1.1 充填候補球の生成

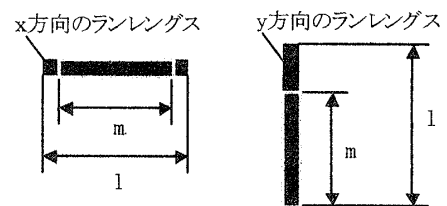
本手法では、2次元の画像であるランレンクスを3次元に拡張して用いるが、ここでは簡単のため2次元で説明を行う。

まず、形状の画像をx,yの各方向に探索し、x,y各方向のランレンクスを生成する。図2(a)に示すように、充填する円を近似する十字の画像を考え、この十字の画像が形状の画像の内部に入るか否かをチェックする。そのため、図2(b)に示すように、形状の画像のx方向のランレンクスの長さmと十字の画像のx方向の長さlを比較する。十字の画像のほうが短い場合、y方向についても同様に比較する。その結果、この十字の画像が形状の画像の内部に入る場合、図2(c)の十字の中心である画素pを充填候補円の中心座標、十字の長さrを候補円の半径とする。本手法では充填円同士の重複を許す条件であるため、形状の画像の内部であると判定された同じ大きさの十字の画像は全て充填候補円とする。

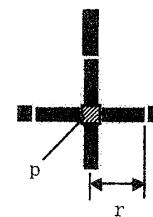
十字の画像が全く形状の画像の内部に入らない場合は、十字の画像の両端を1画素縮めて十字の大きさを小さくし、再び十字の画像と形状の画像の内外判定を行う。この処理は、十字の画像が形状の画像の内部に入るまで繰り返す。



(a) 十字の画像



(b) ランレンクスと十字の画像の比較



(c) 充填候補円

図2 充填候補円の生成

### 3.1.2 充填候補球の内外判定

手順1において内部と判定された充填候補球の表面が形状の画像の内部に入るかどうかを判定する。そのため2次元において高速にデジタル円を描画するミッチェナーのアルゴリズム<sup>[6]</sup>を3次元に拡張する。

ミッチェナーのアルゴリズムは、原点を中心とする半径  $r$  ( $r$  は整数) のデジタル円を  $1/8$  円の円弧として8連結で描く。残りの範囲は円の対称性を利用して描く。例として  $r=4$  のときに描かれるデジタル円を図3に示す。

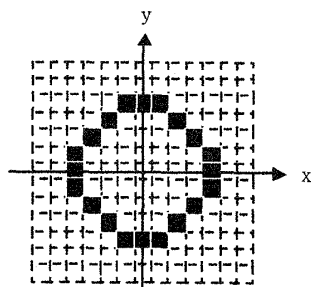
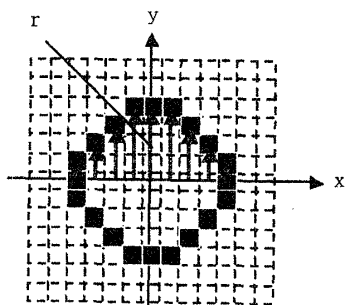


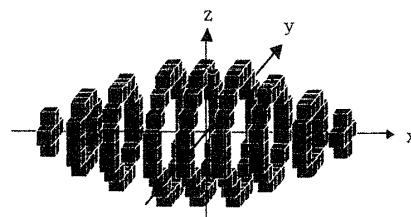
図3  $r=4$  のとき描かれるデジタル円

本手法では、手順1により求めた充填候補球の中心座標  $p$  と半径  $r$  を用いて、ミッチェナーのアルゴリズムにより描かれる座標の画素が形状内部にあるか否かを判定する。そして、全ての画素が形状の内部と判定された場合、充填候補球の中心座標と半径を充填球の中心座標と半径として決定する。

ミッチェナーのアルゴリズムの3次元への拡張は、最初に  $xy$  平面において2次元のデジタル円を描く。次に、図4(a)に示すようにデジタル円の画素までの  $y$  方向の距離  $r$  を半径として図4(b)に示すように、 $yz$  平面において2次元のデジタル円を描くことにより3次元化を行う。



(a)  $yz$  平面のデジタル円の半径



(b) 3次元化

図4 ミッチェナーのアルゴリズムの3次元への適用

以上の処理により、デジタル球である充填候補球と形状の画像との内外判定を行い、内部と判定された場合、図5に示すように、充填候補球の中心座標と半径を充填球の中心座標と半径として採用する。また、本手法では球同士の重複を許す条件であるため、ミッチェナーのアルゴリズムにより形状の画像の内部であると判定された全ての充填候補球を充填球として採用する。

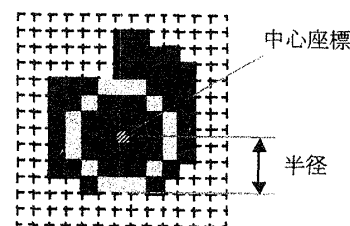


図5 充填球の中心座標と半径

### 3.1.3 白画素による塗りつぶし処理

ここでは、簡単のため2次元で説明する。

手順2において採用された全ての充填円に対して、図6に示すように、充填円の内部に含まれる形状の画像を白画素で塗りつぶす。そして、再び手順1に戻り、充填候補円から充填円を決定し、同様の処理を行い、黒画素がなくなった時点で処理を終了する。

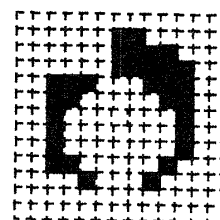


図6 白画素による塗りつぶし処理

### 3.2 8分木の生成

上記において求めた充填球群を8分木構造とすることにより、衝突判定において検索する球の数を減少させ計算コストをより低く抑える。ここでは簡単のため2次元の4分木で説明する。

図7(a)のように、形状の内部に複数の円が充填されている場合の4分木の生成手順を以下に示す。

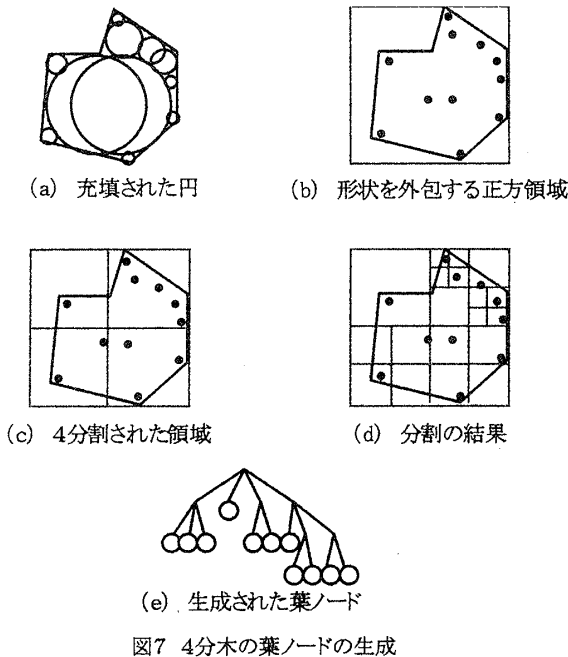


図7 4分木の葉ノードの生成

まず、図7(b)に示すように形状を外包する正方領域を考える。そして、正方領域に対して以下の処理を行う。

1. 正方領域に円の中心座標が複数あるか否か調べる。
2. 手順1において複数ある場合、図7(c)のように正方領域を4分割する。4分割した各々の領域について手順1を繰り返す。
3. 手順1において、各々の正方領域に円が1個しか存在しない場合、その円の中心座標と半径を4分木の葉ノードのデータとし処理を終了する。また、球が0個の場合は処理を終了する。

分割の結果を図7(d)に示す。また、分割の結果により生成された4分木を図7(e)に示す。上記の処理では末端の葉ノードのみに円のデータが生成された状態であるので、4分木の各ノードに外包円を生成する。

本手法では、図8(a)に示すように木構造の親子関係

を用いて外包円を生成する。まず葉ノードを子とし、複数の子の円を外包する外包円を生成する。さらに、複数の外包円を子とし、その親である外包円を外包する円を生成する。以上の処理を木の根であるルートノードに達するまで行う。

図8(b)に示す3個の子の円A,B,Cを例にとって外包円の生成手順を示す。

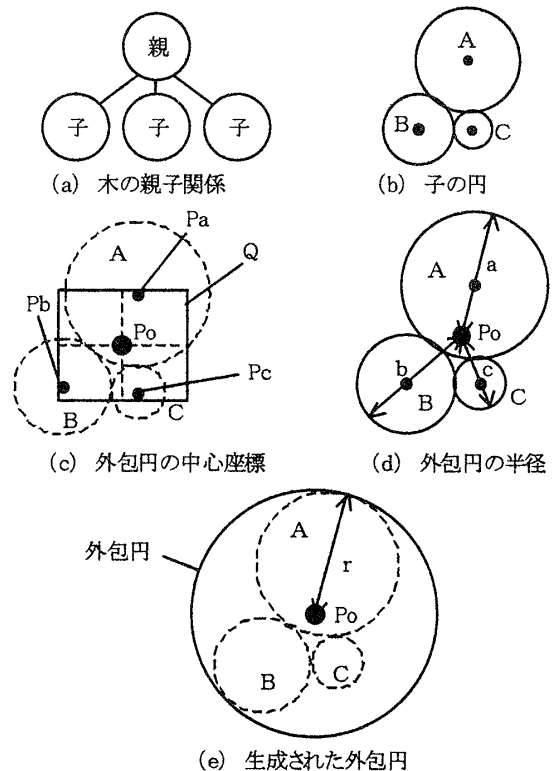


図8 外包円の生成

外包円の中心座標は、同図(c)のように円A,B,Cの中心座標  $P_a, P_b, P_c$  を外包する長方形  $Q$  を考える。そして長方形  $Q$  の中心座標  $P_o$  を外包円の中心座標とする。

また外包円の半径は、図8(d)に示すように外包円の中心座標  $P_o$  から子の円の中心点を通り、円の最も遠い地点までの距離  $a, b, c$  を求める。そして、距離  $a, b, c$  の中で最長のもので外包円の半径とする。図8(d)においては距離  $a$  が最長であるため外包円は図8(e)のようになる。

以上の処理により、衝突判定に用いる8分木を生成する。

### 3.3 衝突判定

本手法では、生成された8分木を用いて衝突判定を行う。説明は簡単のために2次元の4分木で行う。

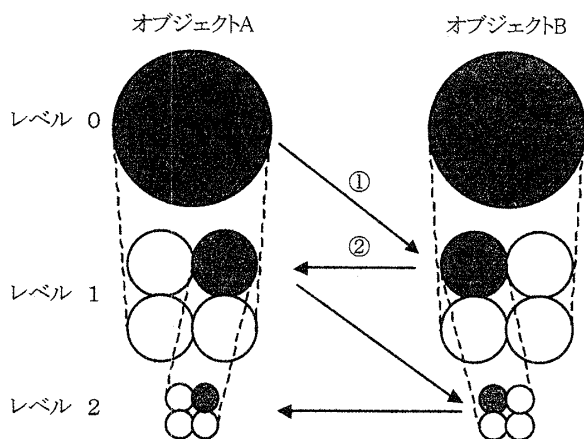


図9 衝突判定の手順

まず最初にオブジェクトA,Bの4分木のルートノードであるレベル0の外包円同士で衝突判定を行う。そして、レベル0の外包円同士が衝突する場合、図9の矢印①に示すように、オブジェクトAのレベル0の外包円とオブジェクトBのレベル0の外包円の内部にあるレベル1の外包円との衝突判定を行う。また逆にオブジェクトBのレベル0の外包円とオブジェクトAのレベル0の外包円の内部にあるレベル1の外包円との衝突判定を行う。そして、矢印②に示すように、衝突した全てのレベル1の円同士により衝突判定を行う。そして、図9に示すレベル1の灰色の外包円が衝突していると判定された場合、衝突したレベル1の外包円の内部に含まれる外包円に対して、以下同様の処理を葉ノードに達するまで繰り返す。そして、オブジェクトA,Bの葉ノードである充填円同士で衝突判定を行い、衝突と判定された場合に2つのオブジェクトは衝突したと決定される。

### 4. 実験

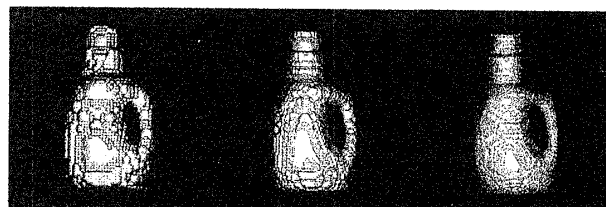
本手法の有効性を検証するため、製品形状に対して従来法である三角形ポリゴンを用いた衝突判定と本手法の衝突判定の比較実験を行った。なお実験にはシリコングラフィクス社製GWS (CPU: R10000, 175MHz)を使用した。

実験で扱った形状の一例を図10(a)に示す。なお空間分割の解像度はレベル6 (64×64), レベル7 (128×128), レベル8 (256×256)として各々について実験を行った。また、各解像度における充填球の結果を図10(b)に示す。なお、図10(b), (c)のレベルの横に充填球の個数を示す。



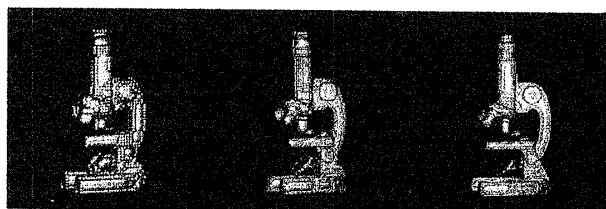
bottle (12090 面)      microscope (7690 面)

(a) 実験形状



レベル 6: 2760      レベル 7: 13845      レベル 8: 71374

(b) bottle の充填球



レベル 6: 1846      レベル 7: 8385      レベル 8: 40166

(c) microscope の充填球

図10 実験形状と充填球

次に、本手法における8分木の生成時間を図11に示す。

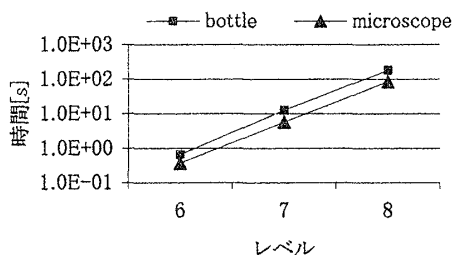


図11 8分木の生成時間

また、衝突判定において形状をランダムに1000回配置した時の各々の判定時間を総和し、その平均時間を結果として図12に示す。

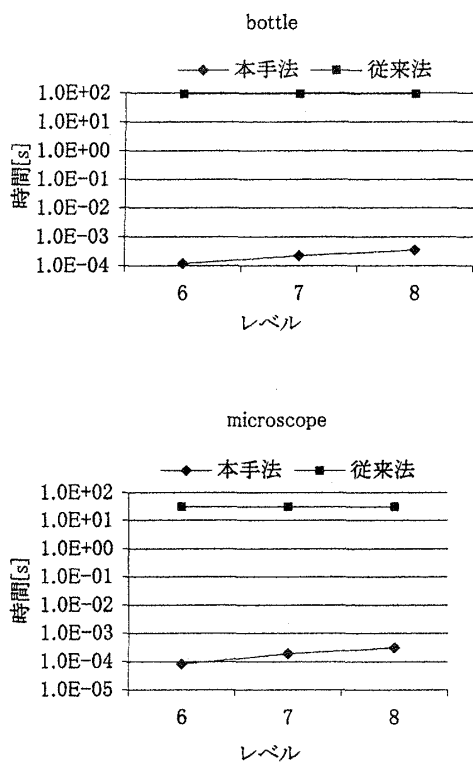


図12 衝突判定時間

### 5. 考察

まず、図10の(b),(c)より球の充填状態から、レベルを上げていくと形状を効率よく近似していることがわかる。

一方、図11に示す8分木の生成にかかる時間については、各解像度において約10倍程度で増加している。しかし、本手法では、8分木の生成は前処理として行うため、ウォークスルー時には処理負荷はない。

次に、図12に示す衝突判定時間において、本手法と三角形ポリゴンを用いた従来法とを比較して1.0E+05程

度の開きがあることがわかる。従来法は形状を構成する三角形ポリゴンの面数に依存するため、実験に用いた両形状の面数になると実時間で衝突判定を行うことは困難であると考えられる。

一方、本手法の判定時間においては、bottle と microscope の両形状とも解像度に比例して若干増加しているが、各解像度について実時間で衝突判定を行うことが可能となっている。

最後に、衝突判定において、三角形ポリゴンの面数が非常に少ない単純な形状においては、本手法よりも従来法が有効であると考えられる。しかし、ウォークスルーシステムにおけるオブジェクトは、今後、より大規模な仮想環境が用いられる傾向にあるため、本手法は有効であると考えられる。

### 6. おわりに

本手法は、形状の内部に効率よく球群を充填するために、2次元の画像ランレングスを3次元に拡張した。また、衝突判定において判定を行う球の数を削減するために、外包球により充填球を8分木構造とした。そして、8分木を用いて衝突判定の実験を試みた。また、実験結果から従来法に比べて判定の処理コストを大幅に抑えることができ、本手法の有効性を確認した。

今後は、外包球の生成手法の効率化を計り、より高速な判定手法を確立していくことを検討する。

### <文献>

- [1] P.M.HUBBARD: "Time-Critical Collision Detection", ACM Transactions on Graphics, 15(3)pp.179-210(1996).
- [2] I.J.Palmer and R.L.Grimdale: "Collision Detection for Animation using Sphere-Tree", Computer Graphics, 14(2)pp.105-116(1995).
- [3] 北村, 竹村, ナレンドラ, 岸野: "octree と多面体表現を用いた三次元物体間の衝突検出", 信学技報, IE94-16pp.45-51(1994).
- [4] F.Foley, A.VanDam, S.Feiner and J.Hughes: "Computer Graphics Principles and Practice Second Edition", Addison Wesley Publishing Company, pp.549-555(1990).
- [5] テレビジョン学会編: "画像情報圧縮", オーム社, pp.150-153(1991).
- [6] 中嶋編: "コンピュータグラフィクス", 財団法人画像情報教育振興協会, pp.41-62 (1998).