

## Cyber Radar : Zバッファ法を利用したリアルタイム衝突検出法とその応用

山地秀美 新藤義昭  
日本工業大学

著者らは、Zバッファ法を利用し、移動するオブジェクトから移動先、およびその逆方向に描画を行って得られた2つの距離画像から衝突検出をリアルタイムに行う手法であるCyberRadarを提案した。平行移動においては検出方向が一意に定まるが、回転を伴う場合は検出方向の設定に任意性が生まれる。そこで回転軸に直交し互いに逆向き of 2つの方向に対して衝突検出を行う手法を提案し、その有効性の検証を試みた。

### CyberRadar : The Real-time Collision Detection using Z buffers and Application

Hidemi Yamachi, Yoshiaki Shindo  
Nippon Institute of Technology

We proposed a real-time collision detection method, named *Cyber Radar*, for geometric models in a 3D Cyber Space. *Cyber Radar* uses Z buffer method to obtain the distance between moving object and others. Two Z value sets are generated; one is from the moving object to the direction to move and another is from the destination to move to the moving object. In the case of translation, the direction to detect is specifically defined, but not in the case of rotation. In this case we propose using two *Cyber Radars* in anti direction each other on the surface orthogonal to the rotation axis.

#### 1. はじめに

3次元コンピュータグラフィックスにおいて、オブジェクト相互の衝突検出は不可欠の要素の一つである。さまざまな衝突検出のアルゴリズムが提案されている状況は、その重要性を物語っている。そのほとんどは、オブジェクトを構成する面や頂点の交差を幾何学的に計算するもので、交差を調べる対象となるオブジェクトのペアを選び、衝突の可能性のある部分に限定して交差の計算を行う。従ってオブジェクトの数が増加すると、対象となるペアの数は $O(n^2)$ で増加する。これを防ぐために、空間やオブジェクトを分割しペアの数を減らすことが、これらのアルゴリズムの重要な問題となる。空間分割による方法としてはOctreeやBSP-treeなどがある。オブジェクトをバウンディングボリュームで囲む方法では、座標軸

方向に向いた直方体を利用するAABB (Axis Aligned Bounding Boxes)、任意の方向に向いた直方体を利用するOBB (Orient Bounding Boxes)、球を利用したBounding Spheresが良く知られている [1-7]。

これらの手法は、オブジェクトの動きや変形に伴い、空間分割やバウンディングボリュームの再構成をしなければならない。衝突検出のコストは、オブジェクトの数、動き、および複雑さに強く依存する。

これらの方法とは異なり、ラスタライゼーションにおいて利用されるZバッファ法を使い、オブジェクトとの距離を調べる手法も提案されている [8]。Z値を利用する方法は実装が容易であり、グラフィックスハードウェアの高速化の恩恵を直接受けることができるというメリットがあるが、現状では実用に耐えるものとはなっていない。

ない。

著者らはZバッファ法の利点を生かし、オブジェクトの数、動きや変形にあまり依存せず、十分な精度でリアルタイムに衝突検出を行う手法、*CyberRadar*を提案した [9]。衝突検出を行うオブジェクト (**Sensor Object**) の位置に視点 (**Prove Point**) を移動し、センサーオブジェクトの移動領域をビューボリュームとして描画を行う。このときセンサーオブジェクトは描かない。続いて、移動先からセンサーオブジェクトだけを描画する。この2つのZ値から衝突検出を行う。描画は平行投影法で行う。ここでの描画はZ値を取得するために行うので、ライティング、シェーディング、透明処理、テクスチャマッピングなどを省いた、簡略化された描画で十分である。また通常の描画に比べれば、低解像度で描くことができる。

*CyberRadar* は以下のような特徴を持つ。

1. オブジェクトの複雑さ、動き、変形に依存しない。
2. 主要な検出コストは、簡略化された低解像度の2回の描画時間である。
3. 2つのオブジェクトの距離を利用するため、連続時間で衝突を検出できる。
4. 検出精度は検出領域の大きさと画素数によって決まるので、動的に精度を変化させられる。
5. アルゴリズムが単純で、実装が容易である。
6. Z値から、衝突点の中心位置および衝突面の法線を推定できる。

現在使用されている多くのグラフィックスハードウェアはZバッファ機能をサポートしており、本手法は多くのシステムで利用可能である。

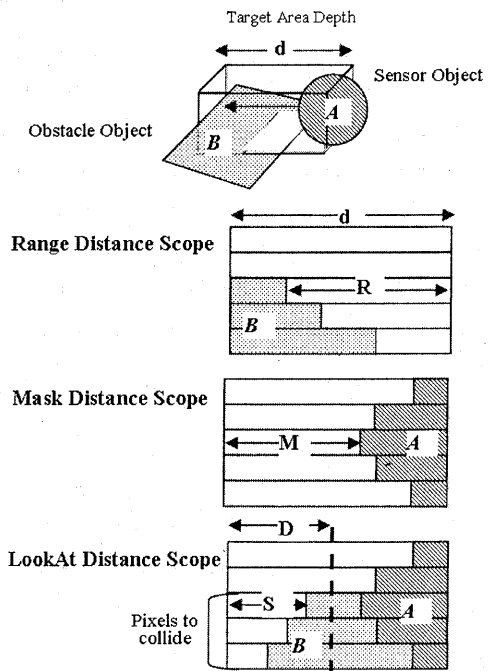
平行移動においては、検出方向が一意に定まるが、回転においては軸の位置によって1回以上の検出が必要になる。そこで、検出方向の定義と制御を検討し、*CyberRadar*による回転における衝突

検出のアルゴリズムを提案する。

以下、2章では*CyberRadar*のアルゴリズムについて紹介する。3章では回転における*CyberRadar*の制御方法の提案、4章では実験の詳細と結果、5章では問題点と課題について述べる。

## 2. *CyberRadar*のアルゴリズム

*CyberRadar*による衝突検出のアルゴリズムを示したのが図1および2である。*CyberRadar*による衝突検出では、Z値を得るための2度の平行投影法による描画を行う。検出対象となる領域 (**Target Area**) は、センサーオブジェクトが移



$d$  : Target Area Depth  
 $R$  : Distance to obstacle object  
 $M$  : Distance to Sensor Object  
 $D$  : Distance to move  
 $S$  : Distance possible to move  
 $S = d - (d - R) - (d - M) = R + M - d$  (1)  
 Pixels to collide :  $S < D$  (2)

図1 *CyberRadar*による距離画像の生成

1. センサーオブジェクトの位置からの描画条件を設定する
2. センサーオブジェクト以外を描画する
3. RDS(R)を取得する
4. if min(R) = 最大値  
    return 衝突なし
5. 逆の位置からの描画条件を設定する
6. センサーオブジェクトだけを描画する
7. MDS(M)を取得する
8. Mの左右を入れ替える
9. LDS(L)を作成する
10.  $L = R + M - d$  d:ターゲット領域の奥行き
11. if min(L) > 移動距離  
    return 衝突なし  
    else  
    return 衝突あり

図 2. *CyberRadar* のアルゴリズム

動することによって掃きだされる領域を囲む直方体となり、これをビューボリュームする。

最初はセンサーオブジェクトから移動方向に向かって描画するが、このときセンサーオブジェクトは描画から除外する。ここで得られたZ値配列（距離画像）を**Range Distance Scope (RDS)**と呼ぶ。この値にZ値の最大値未満の画素がなければ、衝突は発生しないことがわかる。

そうでなければ、逆の位置からセンサーオブジェクトだけを描画し、距離画像を取得する。これを**Mask Distance Scope (MDS)**と呼ぶ。RDSの画素と対応させるために、MDSを左右反転する。

RDSとMDSからセンサーオブジェクトと他のオブジェクトとの間隔を求めた**LDS (LookAt Distance Scope)**を作り、画素ごとに衝突の有無を判定する。これらの手順を図2に示した。

*CyberRadar*の検出コストは、2回の描画と距離画像の解析に分けられる。多くの場合、その大半

は描画コストが占める。検出のための描画は距離画像を取得するためだけに行われるので、ライティング、シェーディング、透明処理、テクスチャマッピングなどの処理を省いて簡略化し、描画コストを削減することができる。

このとき、カラーコードによるオブジェクト識別が利用できる。オブジェクトごとに付与したコードを、そのオブジェクトの色として描画することで、Z値に対応する画素の色から、オブジェクトを識別する。

### 3. 回転における衝突検出

回転における衝突検出を*CyberRadar*で行う場合に、さまざまな問題が発生する。オブジェクトが凹凸のある面の上を転がる場合を想定すると、以下の問題が考えられる。

1. 検出方向が1つ以上になり、一意に決めることができない。
2. 回転軸で他のオブジェクトに接触していた場合、その位置は衝突検出から除外しなければならない。
3. 回転可能な角度を軸の近傍での間隔から求めると、誤差が大きくなる。
4. 回転角度が大きくなるにつれて、RDSとMDSの画素のずれが大きくなる。
5. 検出方向に直交する面の衝突は検出できない

平行移動に対しては一意にターゲット領域を定義することができるが、回転運動では一意に定まらない。回転軸の位置やオブジェクトの形状によって、複数の方向での衝突検出が必要になる(図3)。

1については、検出コストをできるだけ小さくするために、2つの*CyberRadar*を使用することにした。またその方向は、重心から回転軸を結ぶ線および回転軸に直交し、互いに逆向きとする(図4)。

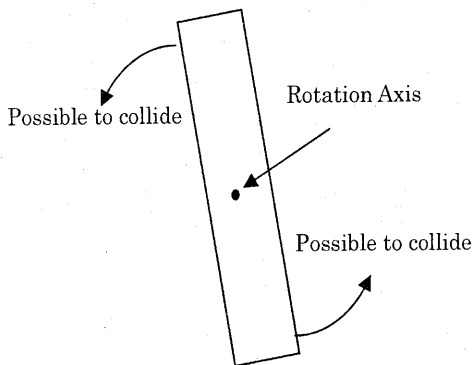


図3 回転における衝突時検出の可能性

回転の衝突検出は、回転軸からの距離とその位置のLDSの値から、回転可能な角度を求めることで行う。従って回転軸が常に距離画像上のXまたはY軸方向になっていれば、検出が容易になる。そこで常にX軸に平行になるとともに、回転によって衝突が起こる可能性のある部分が距離画像上の上になるようにCyberRadarを制御する。これにより、回転軸より下の部分の検出処理を省くことができる。問題点2、3についてはこうした方法を試みた。

4については、回転させてから衝突の有無を調べ、回転を戻す方法も考えられるが、戻すべき角度を求める際に同じ問題が発生するため、ここでは回転可能な角度を小さくすることで対応した。

5に関しては、センサーオブジェクトの形状に応じたCyberRadarの配置を工夫する必要がある。

これらを元に、回転における衝突検出のアルゴリズムを以下のように定義した。ここでCR1とCR2は2つのCyber Radarを、CRは衝突検出したほうのものを意味する。

1. CR 1 で衝突検出を行い、回転可能な角度  $angle1$  を求める
2. CR 2 で衝突検出を行い、回転可能な角度  $angle2$  を求める

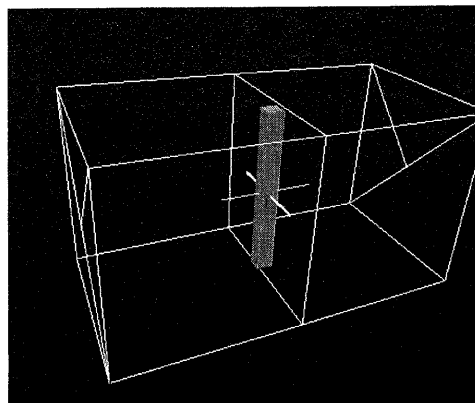


図4 回転における衝突検出のためのCyberRadarの配置

3. if  $angle1$  および  $angle2 >$  検出限界値

(1) 回転を行う

else

- (1) CR の距離画像から、衝突位置を特定する
- (2) 衝突位置から次の回転軸の両端を選択する
- (3) CR を新しい回転軸に平行になるように回転させる

#### 4. 実験と評価

実験では、図5に示すようなさまざまな形状のセンサーオブジェクトと凹凸面を使い、視覚的に自然な回転になっているかを検証した。また、これらのセンサーオブジェクトについて、衝突検出を行った場合と行わなかった場合の1フレームあたりの描画時間をグラフにしたのが図6である。実験はWindows XP, Pentium4 2GHz, 256MByte、GPUはNvidia GeForce3 Ti500 およびOpenGLを使用した。

図7には、回転のプロセスと軸の移動の様子を示した。これから、回転軸の移動がほぼ正しく行われていることがわかる。

これらの結果から、この実験環境においてリア

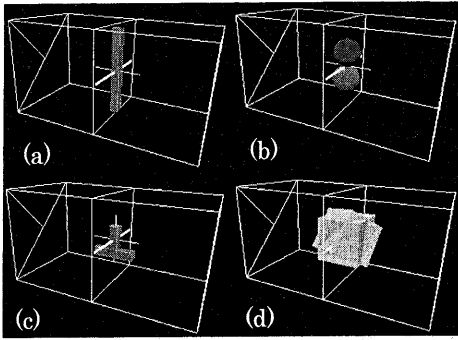


図5 センサーオブジェクト

リアルタイムに回転の衝突検出と軸の選択ができることがわかる。

### 5. 問題点と課題

ここで実験したモデルでは、ほぼ自然な回転運動をリアルタイムで実現することができた。しかし衝突判定の誤差から回転軸の選択を誤ることも多い。凹凸の激しい形状のセンサーオブジェクトでは、こうした問題が顕著になる。距離画像の解析方法および、CyberRadarの数とその配置に工夫が必要になることは明らかである。こうした点についての実験と改良が今後必要となる。

### 参考文献

- [1] G. van den Bergen. "Efficient Collision Detection of Complex Deformable Models using AABB Trees" Journal of Graphics Tools, 2(4):1-13, 1997
- [2] S. Gottschalk, M. Lin, D. Manocha, "OBBTree: A Hierarchical Structure for Rapid Interference Detection", Proc. of ACM Siggraph '96, pp. 171-180, 1996
- [3] P. M. Hubbard, "Approximating Polyhedra with Spheres for Time-Critical Collision Detection", ACM Transactions on Graphics, Vol. 15, No.3, July 1996, pp.179-210
- [4] Gino van den Bergen. "A Fast and Robust GJK Implementation for Collision Detection of Convex Objects", Journal of ACM Graphics Tools, 4(2):7-25, 1999
- [5] J. Cohen, M. Lin, D. Manocha, M. Ponamgi, "I-COLLIDE: An Interactive and Exact Collision Detection System for Large-Scale

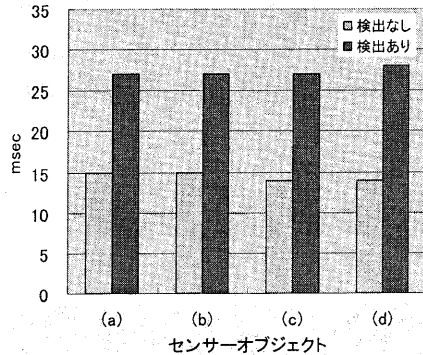


図6 衝突検出の有無による描画時間

Environments", Proc. of ACM Int. 3D Graphics Conference, pp. 189-196, 1995

- [6] T. Hudson, M. Lin, J. Cohen, S. Gottschalk, D. Manocha, "V-COLLIDE: Accelerated Collision Detection for VRML", Appeared in Proc. of VRML'97
- [7] M. Lin, D. Manocha, J. Cohen, S. Gottschalk, "Collision Detection: Algorithms and Applications" <http://WWW.cs.unc.edu/~geom/collide.html>
- [8] M. Shinya and M. Forgue : "Interference detection through rasterization", Journal of Visualization and Computer Animation, Vol. 2, pp.132-134, 1991
- [9] 山地秀美、新藤義昭："Zバッファ法を利用したオブジェクトおよび衝突検出", 情報処理学会論文誌 第43巻 第6号 pp 1899-1909 (2002)

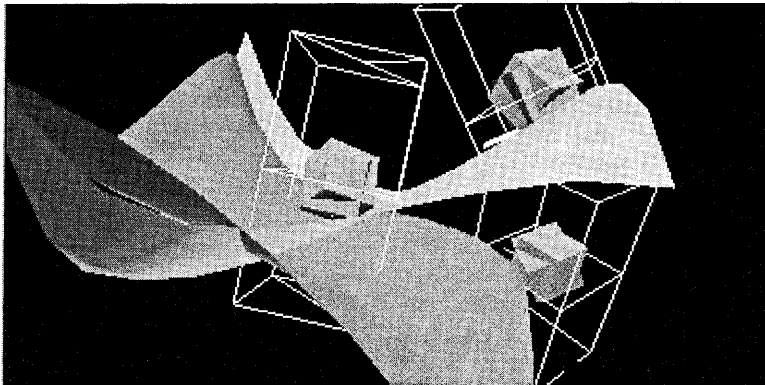
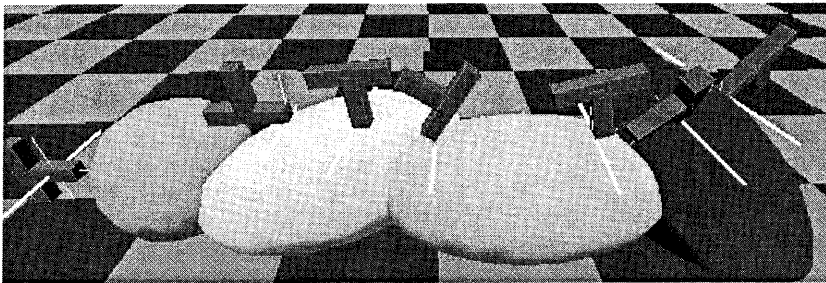
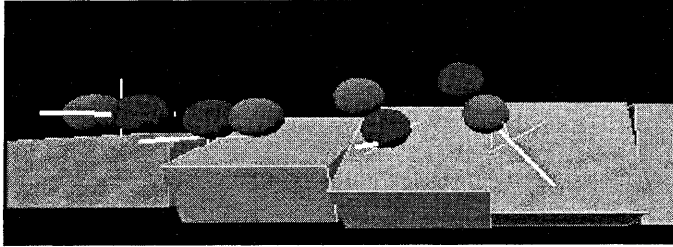
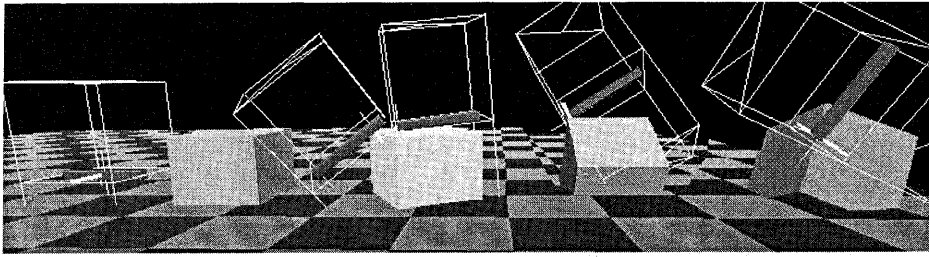


図7 CyberRadar による回転シミュレーション