大規模点群における衝突判定法

東京大学　○平岡美那子，増田宏

Collision Detection for Large-Scale Point-Clouds

The University of Tokyo: Hiraoka Minako, Hiroshi Masuda

We present a method for fast collision detection between an arbitrary moving object and a large virtual environment scanned by a 3D laser scanner. To efficiently process large point-clouds, we convert a point-cloud into a 2D depth map. Moving objects are also projected into the depth map in every time step. Then we can detect interference in a short time by comparing depth values on the depth map. This method enables to detect collision faster than classical spatial partitioning methods. We also present a new collision detection method using multiple depth maps in an out-of-core manner.

1. 目的

近年，中・長距離の計測ができるレーザスキャナが急速に進歩している．レーザスキャナは，対象物の表面にレーザ光を照射し，照射点の３次元座標を点群として取得する[1]．取得データを用いるとコンピュータ上に３次元の大規模仮想環境を構築することができ，地図作成，土木建築，災害シミュレーション，さらにナビゲーションサービスなど多領域での応用が考えられる．

点群で表現された仮想環境でシミュレーション等を行うことを考えると，仮想環境と物体間で衝突判定が必要となることが多い．しかし，離散的な点集合での衝突チェックはポリゴンモデルモデルに比べて難しく，また，高密度・大規模なデータをそのまま用いることはメモリ・計算時間の面からも現実的ではない．さらに，大規模環境の計測では，オクルージョンなどのために計測不能領域が存在するため，単に衝突の有無を判定するだけでなく，「このデータセットからは衝突の有無が不明である」ことの判定も必要である．

本研究ではこうした問題に対処できる大規模点群のための衝突判定手法を考える．従来研究として，画像ベースでの高速な衝突判定を行う研究[2]も存在するが，数千万から数億点規模の大規模点群に対応する方法は示されていない．本研究では，大規模点群をメルカトル図法により２次元のデプスマップに変換し，大規模な点群から高速な衝突判定ができる手法を示す．

2. 衝突判定

2.1 点群からデプスマップへの変換

点探索の高速化のために，点群を平面上に写像し，ピクセルごとに深度をもつデプスマップに変換することを考える．

一般に，１回のレーザスキャンで得られた点群の座標は，計測装置を原点とした座標系(x, y, z)で記述される．レーザスキャナでは，図１左 に示すように，方位角θと仰角φ によってレーザの照射方向が決められる．角度の取り方を図１右に示すように決めると，座標は球面座標(θ，φ, r)に変換できる[1]．球面全体を平面に展開する方法としてよく用いられるのがメルカトル図法である．メルカトル図法では，方位角と仰角(θ, φ)を主軸として用いることで，球面を平面に展開する．図２に，プラントの点群から得られたメルカトル画像の例を示す．

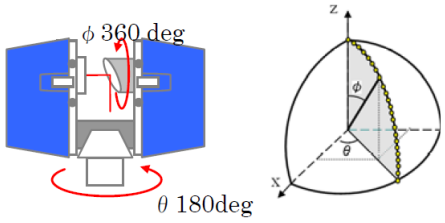

図１　レーザスキャナ（左）と球面座標（右）

図２　プラント点群のメルカトル画像

2.2 衝突物体の表現

衝突判定するオブジェクトは，閉じたメッシュモデルとする．ここで, 計測原点 O からオブジェクトの面上の一点へのベクトル $\mathbf{p}$ と，面の法線ベクトル $\mathbf{n}$ の内積から，表側の面と裏側の面を区別する．すなわち, $\mathbf{p}\cdot\mathbf{n} \geq 0$ であれば「裏側の面」，$\mathbf{p}\cdot\mathbf{n} < 0$ であれば「表側の面」とする．

ここでは，各面に対して，図３に示すように，閾値 ε 内に少な

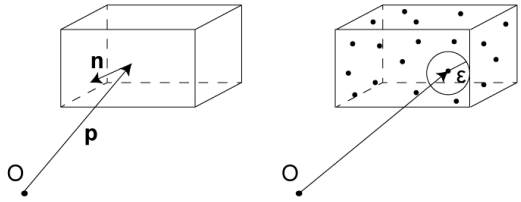

図３　衝突物体の表現

くとも１つの点が存在するように点を発生させる．すなわち，物体を M とすると，M は面の集合{ f_i } で表現され，各面 f_i は，面上の点集合 { p_j } で表現される．

2.3 デプスマップ上での衝突判定

ある点 P(x, y, z)をデプスマップ上に写像し，P のデプス値がデプスマップのデプス値より小さければ，この点は衝突しないと判定することができる．

物体 M との衝突判定は以下のような手順でできる．ここで，物体 M の面のうち，表側の面上のデプス値を r_s，裏側の面上のデプス値を r_e とする．また，点が投影されるデプスマップ上の位置のデプス値を r_E とする．図４は，スキャナ原点から見たときのこれらの位置関係を示している．

1. 物体 M の点集合の座標をスキャナ原点の座標系に変換する．
2. 裏側の面上の点 p をデプスマップ上に投影する．
   ① $r_e < r_E$ であれば，点 p は衝突しないと判定する．
   ② そうでなければ，スキャナ原点と点 p を通る直線に関して，物体 M の表側の面との交点を計算する．複数の交点がある場合は，r_e に最も近い手前の点を取る．このとき，$r_s < r_E < r_e$ であれば，この点は衝突する．
3. 物体 M に関して，以下の判定を行う
   ① すべての裏側の点が衝突しないと判定されれば，物体 M は衝突しないと判定する（図４(a)）．
   ② 少なくとも一つの裏側の点が衝突すると判定されれば，物体 M は衝突すると判定する（図４(b)）．

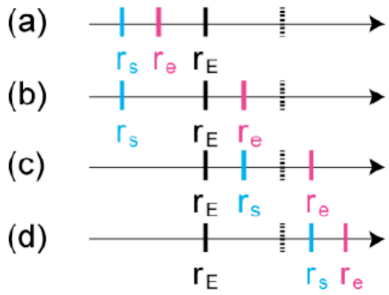

図 4 デプスマップ上の衝突検出

③ 上記のいずれでもない場合は「不明」と判定する（図４(c)(d)）．

3.3 計算時間

上記の手法について，octree を利用した衝突判定法と，１回の衝突判定にかかる平均計算時間を比較した．用いた環境点群は約 250 万点の点が含まれており，衝突物体として球を用いた．octree を用いた手法では，球内部に点が存在すれば衝突すると判定した．デプスマップ法では，球上に１００個の点をランダムに発生させて計算を行った．結果を表１に示す．この結果から，デプスマップを利用することで大幅に計算時間が短縮されることがわかる．

表 1 比較実験結果

	解像度(ピクセル)	計算時間(ms)
octree 全探索		1271
デプスマップ	360×720	1.51
	180×360	2.08

3.4 複数のデプスマップを用いた判定

単一のデプスマップでは，計測されていない領域が大きいため，衝突不明と判定される場合がある．点がデプスマップの奥にあり，衝突する可能性があるが，単に別の物体の背後にあり，実際には衝突していない可能性があるためである．

このような場合には，別の位置から計測したデプスマップを用いる必要がある．本手法においては，N 個のデプスマップが存在する場合，その中の少なくとも１つに対して衝突の有無が判定できればよい．すべてのデプスマップに対して衝突不明となった場合は，このデータセットでは判定ができないという結果が返される．

N 個のデプスマップが存在する場合，毎回，すべてのデプスマップを調べる必要はない．一旦，衝突の有無が判定できるデプスマップが見つかれば，それ以降はそのデプスマップで判定不能となるまでは，同じデプスマップを使い続ければよい．

4. 大規模点群への対応

工場などの大規模な環境においては，広域な範囲をカバーするために，多くの計測を行う必要がある．個々の点群は大容量であるため，すべてのデプスマップをオンメモリに保持することは現実的ではない．デプスマップの解像度を低くすることでデータ量を削減することは可能であるが，その場合，衝突判定の信頼性が低下するという問題が生じる．そこで，本研究では以下のような方法を用いる．

(1) 解像度を落とした粗いデプスマップを作成する．解像度を落とす際は，小さいデプス値を優先させる．

(2) 詳細なデプスマップを格子に分割し，各格子ごとにデプス値をハードディスクに格納する．

(3)衝突物体のスキャナからの距離が閾値 d 以下のデプスマップのみをデプスマップリストに入れる．スキャナからの距離が小さいデプスマップから順に処理を行う．

(4) 粗いデプスマップ上で衝突判定を行う．その際，デプス値の差が閾値 d より小さいならば詳細衝突判定に移る．

(5) 詳細衝突判定を行うときは，該当する箇所のデプスマップをハードディスクから読み込んで実メモリ上に展開する．

(7) 詳細衝突判定で，衝突なし・衝突ありとなればその結果を返す．衝突不明であれば，次のデプスマップを取り出し，(3)以降の処理を繰り返す．

5. まとめ

本研究では，物体と大規模点群との衝突判定を行うためのアルゴリズムを提案した．今後は，様々なデータを用いて，評価実験をしていく予定である．

参考文献

[1] 増田宏:画像インタフェースを用いた大規模点群からのソリッドモデリングシステム,機械学会論文集(C 編), 76 巻 771 号 C 編, pp.2748-2752, 2010.

[2] N. K. Govindaraju, et. al.: CULLIDE: Interactive Collision Detection Between Complex Models in Large Environments using Graphics Hardware, Eurographics Workshp On Graphics Hardware, pages 25-32, 2003