

没入型多面ディスプレイにおける衝突検出システムの開発

浅野 良直 藤井 勝敏 廣瀬通孝* 広田光一*

Development of Collision Detection System for a Full Immersive Display

Yoshinao Asano Katsutoshi Fujii Michitaka Hirose* Koichi Hirota*

あらまし 没入型 6 面ディスプレイシステム (以下 COSMOS) はユーザの全方位をスクリーンで囲むため没入感が非常に高く, 車や飛行機などの 3D モデルを実物大で表示できる利点がある. 多くの企業は “3D モデルの実物大表示” のみを目的として COSMOS を利用している. そのような利用のなかで “モデルに触った感触が欲しい”, “モデル同士の干渉を検出して欲しい” という要望は常に言われてきた. そこで, 本研究では COSMOS のなかでモデル同士の干渉検出やユーザが直感的にモデルに触れて操作できるような衝突検出システムの開発を行った. 衝突検出の方法としてはモデルの形状を外接直方体で近似した方式とモデルを構成するポリゴンデータを利用する方式の 2 つを開発した. この 2 つの方式を組み合わせることで効率的に衝突検出を図ることが可能となった. また, ユーザへの提示方法として衝突したモデルを点滅させる方法と力覚デバイスを利用する方法について開発を行った. 力覚デバイスを利用することでモデルを直感的に操作することが可能となった. キーワード VR, IPT, COSMOS, 衝突検出, 力覚ディスプレイ

1. まえがき

COSMOS[1] (図 1) の特徴の 1 つに “実物大で 3D モデルを表示できる” という点が上げられる. この特徴は従来, 試作を製作するまで分からなかった問題点をデザイン, 設計段階で解消するだけでなく, 試作開発にかかる時間, 費用のコスト削減に大きく影響を与えることが可能となる. そのため, COSMOS を利用する企業の件数も増えているが, その多くは “3D モデルを実物大で表示” してプレゼンテーションを行う程度にとどまっている. そのようななかで従来から要望されていたのが “3D モデルに触って操作したい”, “モデル同士の干渉を検証したい” などである. そこで, 本研究では COSMOS で利用できる衝突検出システムの開発及び力覚ディスプレイの利用を試みることにした. 衝突検出の処理方法としては 3D モデルの形状を外接直方体 (以下 Bounding Box) で近似して行う方法と 3D モデルを構成するポリゴンデータを利用して行う方法の 2 種類について開発を行った [2]. 衝突を検出した場合の提示方法としては, 力覚デバイスを利用した触覚による提示技術を開発した. 力覚デバイスとしては HapticGEAR[3] (図 2) を利用することにした. HapticGEAR は糸の張力を利用してユーザに力覚を返すデバイスであり, ユーザが装着するため仮想空間内を自

由に歩くことができる. ユーザに触覚を与えることで直感的にモデルを操作することが可能になる. 衝突検出システムで表示するモデルは, 多くの 3D モデリングソフトウェアでサポートされている VRML2.0 フォーマット [4][5] を利用したコンバータの開発を行う事とした.



各スクリーンの寸法: 3m × 3m
CPU:16 Memory:4Gbyte HDD:100Gbyte
グラフィクスエンジン:6

図 1 没入型 6 面ディスプレイ COSMOS

Fig.1 Immersive 6-screen Display COSMOS

* 東京大学 先端科学技術研究センター, 東京都
The University of Tokyo RCAST 4-6-1 Komaba,
Meguro-ku, Tokyo, 153-8904 Japan

2. 衝突検出システムについて

2.1 VRML2.0 モデルコンバータ

衝突検出システムで使用する3DモデルとしてVRML2.0フォーマットを利用したモデルコンバータの開発を行った。VRML2.0フォーマットを選択した理由としては以下の点が上げられる。

- ・ 市販されている多くの3Dモデリングソフトウェアから出力できるため、様々なソフトウェアでモデリングを行うことができる。
- ・ データ構造が容易であり、データ構造に関する書籍も多く市販されているため、ジオメトリデータの作成や衝突検出で必要となるデータなどを容易に抽出することができる。
- ・ マテリアルやテクスチャなどが利用できるため、ソフトウェアで作成したモデルに近い状態をCOSMOSで表示することが可能である。
- ・ VRML1.0と異なりモデルの動作やサウンドなどの機能があるため、COSMOSのなかで動きのあるモデルを表示できる可能性がある。

VRML2.0モデルコンバータは、VRML2.0のモデルデータをIRIS Performerライブラリのジオメトリデータで扱えるようにデータを並べ替えた中間ファイルを作成して、COSMOSでの表示を行っている。中間ファイルにはモデルの構成データ（位置座標、構成データ、マテリアルデータ、テクスチャデータ等）以外に、衝突検出で必要とされる以下のようなデータが記載されている。

- ・ 外接直方体を作成するためのモデルの最小/最大値
- ・ ポリゴン毎に衝突検出を行うためのポリゴン数
- ・ 複数のモデルが存在した場合のTree構造

本研究で利用したVRML2.0のモデルデータとしてはPolyTrans（Okino Computer Graphics製）から出力したデータを使用した。

2.2 力覚デバイス

力覚デバイスとして使用したHapticGEARは、糸の張力を利用した力覚デバイスである。糸の張力を利用した力覚デバイスとしてはSPIDER[6]があるが、これは机上に固定するタイプであるためCOSMOSのようにユーザが動きながら作業が行えるシステム環境には不向きといえる。そのため、ポータブル型の力覚ディスプレイとして考えられたHapticGEARが適していると考えられる。

HapticGEARの4本の糸はユーザが持つペン先に取り付けられている。このペンを動かすと各糸に取り付けられたプーリーが回転し、ロータリーエンコーダが回転を検出する。検出された回転から各糸の長さを求めてペン先の3次元座標が算出される。但し、この座標はHapticGEARの座標系であるため、実際は図2(b)に示すように3次元磁気式位置センサ(Polhemus)を取り付けてCOSMOS内で

のHapticGEARの位置を検出し処理を行ってペン先の3次元座標を算出している。3次元磁気式位置センサは磁場のひずみに影響を受けやすい欠点がある。また、磁気発生源からの距離が近いほど検出の誤差が少なくなる。そこでプーリーに取り付けられているDCモータの磁場の影響を避け、また、COSMOSの磁気発生源は上方にあるため、図2(b)のような場所に取り付けられている。



(a) Back



(b) Left

図2 HapticGERAの全景
Fig.2 Photos of HapticGEAR

表 1 衝突検出で対象とした要素

Table 1 Collision detection object element

	操作モデル	近接モデル
点 × 点	原点	Bounding Box の頂点座標
	Bounding Box の頂点座標	Bounding Box の頂点座標
	原点	ポリゴンの頂点座標
	Bounding Box の頂点座標	ポリゴンの頂点座標
	ポリゴンの頂点座標	ポリゴンの頂点座標
点 × 平面	原点	Bounding Box の構成面
	Bounding Box の頂点座標	Bounding Box の構成面
	原点	モデルの構成面 (ポリゴン)
	Bounding Box の頂点座標	モデルの構成面 (ポリゴン)
	ポリゴンの頂点座標	モデルの構成面 (ポリゴン)

2.3 衝突検出方法

衝突検出では表 1 に示すような，ユーザが操作するモデルと近接するモデルの各対象要素の距離等を算出して衝突の検出を行った．但し，衝突を検出する対象要素間の衝突検出距離 l_{col} はユーザが入力できるようにした．

2.3.1 衝突の検出

点 $a(x_a, y_a, z_a)$ と点 $b(x_b, y_b, z_b)$ の衝突検出では，式(1)から 2 点間の距離 l_{pp} を算出して検出を行った．

$$l_{pp} = \sqrt{(x_a - x_b)^2 + (y_a - y_b)^2 + (z_a - z_b)^2} \quad (1)$$

点 (x_0, y_0, z_0) と平面 $(ax + by + cz + d = 0)$ の衝突検出では，平面を構成する頂点座標の 3 点 (x_a, y_a, z_a) ， (x_b, y_b, z_b) ， (x_c, y_c, z_c) から平面の傾き (a, b, c) を算出(式(2))して距離 l_{pf} を求めた(式(3))．但し，Bounding Box やポリゴンは無限平面ではないため，式(4)または式(5)より平面上の衝突座標 $P_{col}(x, y, z)$ を算出し，衝突座標 P_{col} が Bounding Box の構成面またはポリゴンの範囲内であることを条件とした．式(4)は点が平面の法線方向に対して正の位置にある場合に利用し，式(5)は負の位置にある場合に利用する．また，式(2)で求められた平面の傾きは法線ベクトルであるため，力覚デバイスを使用する場合の力覚の方向として利用できる．

$$Na = y_a(z_b - z_c) + y_b(z_c - z_a) + y_c(z_a - z_b)$$

$$Nb = z_a(x_b - x_c) + z_b(x_c - x_a) + z_c(x_a - x_b)$$

$$Nc = x_a(y_b - y_c) + x_b(y_c - y_a) + x_c(y_a - y_b)$$

$$N = \sqrt{Na^2 + Nb^2 + Nc^2}$$

$$\left. \begin{aligned} a &= \frac{Na}{N} \\ b &= \frac{Nb}{N} \\ c &= \frac{Nc}{N} \\ d &= -(ax_a + by_a + cz_a) \end{aligned} \right\} (2)$$

$$l_{pf} = \frac{|ax_0 + by_0 + cz_0 + d|}{\sqrt{a^2 + b^2 + c^2}} \quad (3)$$

$$\left. \begin{aligned} x &= \frac{-l_{pf} + a}{\sqrt{a^2 + b^2 + c^2}} + x_0 \\ y &= \frac{-l_{pf} + b}{\sqrt{a^2 + b^2 + c^2}} + y_0 \\ z &= \frac{-l_{pf} + c}{\sqrt{a^2 + b^2 + c^2}} + z_0 \end{aligned} \right\} (4)$$

$$\left. \begin{aligned} x &= \frac{l_{pf} + a}{\sqrt{a^2 + b^2 + c^2}} + x_0 \\ y &= \frac{l_{pf} + b}{\sqrt{a^2 + b^2 + c^2}} + y_0 \\ z &= \frac{l_{pf} + c}{\sqrt{a^2 + b^2 + c^2}} + z_0 \end{aligned} \right\} (5)$$

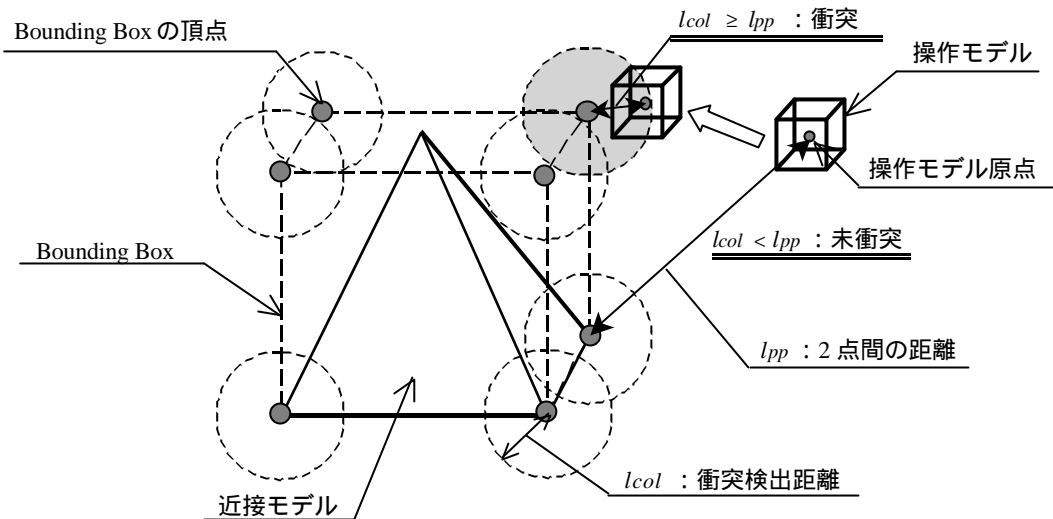


図3 点と点の衝突検出 (例: 原点对 Bounding Box の頂点)

Fig.3 Collision detection (point and point)

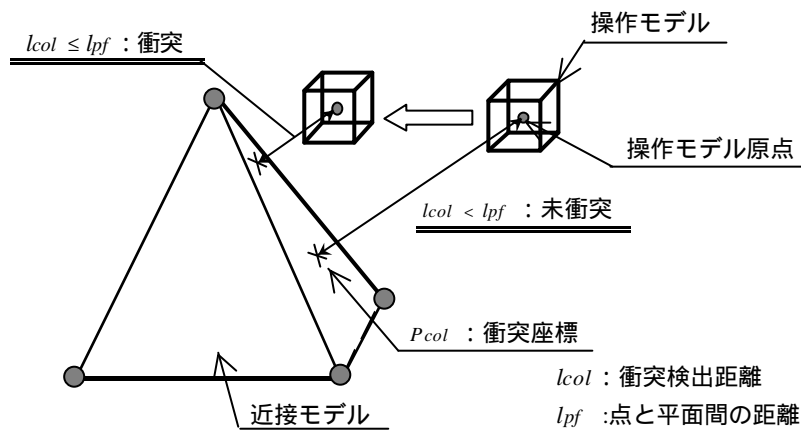


図4 点と平面の衝突検出 (例: 原点对ポリゴン)

Fig.4 Collision detection (point and plane)

2.3.2 点と点の衝突検出

操作するモデルの対象要素 (原点, Bounding Box 及びポリゴンの頂点座標) と近接するモデルの対象要素 (Bounding Box またはポリゴンの頂点座標) の距離が衝突検出距離以下 ($l_{col} \geq l_{pp}$) の値となった場合にのみ衝突と判定する. 図3に原点对 Bounding Box (頂点) における衝突検出のイメージ図を示す.

2.3.3 点と平面の衝突検出+

点と点の衝突検出と同様に操作するモデルの対象要素 (原点, Bounding Box 及びポリゴンの頂点座標) と近接するモデルの対象要素 (Bounding Box の構成面またはポリゴン) の距離が衝突検出距離以下 ($l_{col} \geq l_{pf}$) であり, 衝突座標 P_{col} が対象面上の範囲内にある場合に衝突と判定する. 図4に原点对ポリゴンにおける衝突検出のイ

メージ図を示す.

2.4 提示方法

操作モデルが近接モデルに衝突と判定した場合の提示方法として, 以下の“点滅による提示”と“力覚デバイスによる提示”の2通りを試みた.

2.4.1 点滅による提示

点と点の衝突検出, 及び, Bounding Box を利用した衝突検出の場合には衝突した面の特定ができないためモデル全体を点滅させることで提示を行った. 点とポリゴン面の衝突の場合は衝突した面の特定ができるため, 衝突した面と同じ傾きをもつポリゴンの点滅を行った. 詳細なモデルではポリゴンも非常に細かくなり, 衝突面のみを点滅させても分かりにくい. そこで, 衝突した面と同じ傾きを持つ面を点滅させることで衝突部分を認識できるようにした.

2.4.2 力覚デバイスによる提示

点と平面の衝突検出の場合、衝突した面の傾きを利用して力覚の提示を行った。これにより、点とポリゴン面の衝突検出ではモデルの形状にあわせた力覚をユーザに提示することが可能になる。但し、点と点の衝突検出のように衝突した面の傾きを得ることができない場合には、任意の方向に力覚の提示を行って衝突を認識できるようにした。

3. 衝突検出システムの評価

衝突検出システムの評価実験として、“Bounding Boxの構成面×操作点”、“ポリゴン×操作点”の2つ方法について評価を行った。

Bounding Boxは詳細なモデルでも衝突範囲を簡略化して衝突検出の処理を行うため処理時間が短い利点がある。しかし、簡略化するためにモデルの存在していない空間まで衝突を検出する問題がある。そのため、決められた地点にモデルを設置させることは難しいと考えられる。

ポリゴンを利用して衝突検出を行う方法はモデルの表面付近で衝突検出を行うため、Bounding Boxのような問題はないが、全てのポリゴンに対して衝突検出処理を行うためシステムへの負荷が大きくなり、フレームレートが低下する問題が生じる。従って、ポリゴンの数によってはフレームレートが極端に下がりリアルタイムに操作が出来ない可能性も生じる。

そこで、3.1に示す作業を行い、被験者(7人)の作業時間を測定して作業性を検討し、また、描画のフレームレートを測定して検出処理の違いによるCOSMOSにかかる負荷を検討する。

3.1 作業内容

図5に示すようにプレートAの中心に操作するモデルを表示し、それを被験者が把持してプレートBの中心までモデルの先端を移動させる作業を行った。操作するモデルはポリゴン数が異なる7種類のモデル(ポリゴン数:16, 32, 64, 128, 256, 512, 800)を順番に表示、設置の精度としては、実空間における組立作業を考慮して、プレートBの中心から半径1mm以内にモデルの先端が移動するまで行うこととした。操作デ

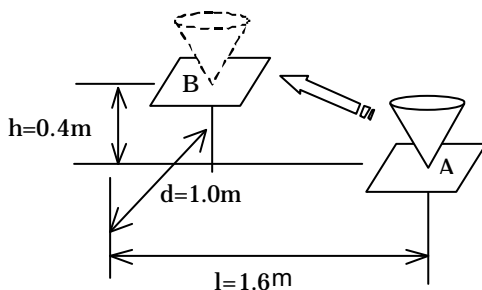


図5 評価実験

Fig5 Evaluation experiment

バイスとしてモデルモデルを直感的に操作するためHapticGEARを利用した。

4. 評価結果

4.1 作業時間

各モデルで要した作業時間の平均を表2に示す。全体的にポリゴンを使用した衝突検出の方が早く作業が終了したことが分かる。そのため、仮想空間内で位置をあわせる操作はポリゴンによる衝突検出を利用した方が有効であると考えられる。しかし、作業時間の平均は全て20秒以上必要であり、被験者によっては180秒以上操作しても設置ができないケースも生じた。このような原因として、本実験ではCOSMOS内のできるだけ広い範囲を作業領域として使用したため、3次元位置センサの精度が悪いスクリーンの付近で位置あわせの作業を行うこととなり、モデルの位置検出が安定しなかったことや、そのような状況下において設置の精度が1mm以下という厳しい条件であったため、このように作業時間が増加したと考えられる。但し、3次元位置センサはCOSMOSの中心付近でも様々な影響を受けるため許容範囲はできるだけ大きくとる必要がある。

表2 操作時間

Table 2 Operation time

[sec]

ポリゴン数	16	32	64	128	256	512	800	平均
Bounding Box	58.9	61.6	25.4	62.9	55.4	39.1	44.4	49.7
Polygon	56.9	26.4	34.3	34.9	28	23.7	32.6	33.8

4.2 フレームレート

図6に各モデルで作業を行ったときのフレームレートの平均を示す。Bounding Boxはモデルのポリゴン数が増えても、ほぼ同じくらいのフレームレートで描画が行えていることが分かる。ポリゴンによる衝突検出はBounding Boxの方法と比較すると半分程度まで描画の速度が遅くなっている。

本実験ではポリゴン数が最大でも800程度であるが、COSMOSを使用する企業の多くは、非常に詳細なモデルを表示する場合や複数のモデルを同時に表示する場合もあるため、モデルの表示だけでフレームレートが落ちてしまうことがある。このような状況で衝突検出を行う場合には、Bounding Boxを利用した衝突検出を利用すれば、衝突検出を使用しない場合のフレームレートに近い値で処理を行うことができる。それに対して、ポリゴンを利用した衝突検出では大きくフレームレートを低下させることになるが、現実に近い状態で作業が行うことができるため、必要な部分のポリゴンに対してのみ衝突検出を行うようにすればフレームレートをあげることが可能になる。

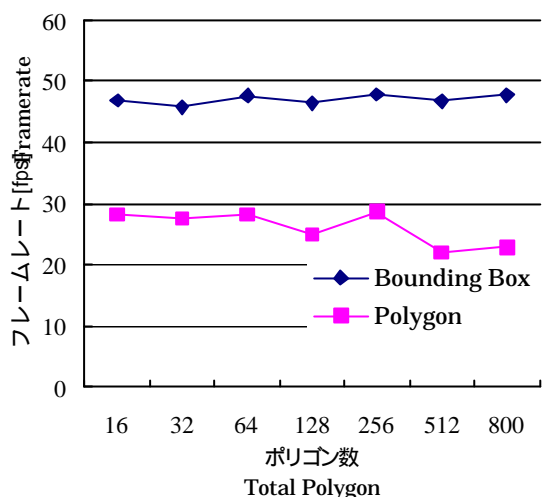


図6 フレームレートの比較

Fig6 Comparison of frame rate



Work scene

図7 アセンブルシミュレーション

Fig7 Assemble Simulation

5. 衝突検出システムの利用方法

5.1 アセンブルシミュレーション

衝突検出システムを利用した利用例としてエンジンのピストン部分の組立作業を行うシミュレーションシステムを開発した(図7)。3次元位置センサは場所によって値が安定していないため、ボルトなどの小さいモデルを把持することは非常に困難である。そのため、衝突検出の範囲が比較的広いBounding Boxによる衝突検出方法を使用してモデルの操作を行った。また、モデルが取り付け位置まで移動した場合には、音を出すことでユーザに取り付け完了の提示を行った。

6. むすび

IPT ディスプレイシステムにおける衝突検出システムの開発を行った。衝突検出の方法としてはBounding Boxとモデルを構成するポリゴンのデータを利用し、ユーザへの提示方法として力覚ディスプレイを用いた。

Bounding Boxを利用するとモデルの形状にあわせた衝突検出はできないが、処理が簡単なためフレームレートを大幅に落とすことなくモデルの衝突検出を行うことができた。ポリゴンによる衝突検出では、正確に衝突部分を検出できるため、モデルのポリゴン数が少なければBounding Boxよりも早く位置あわせを行うことができる。但し、ポリゴン数の多いモデルではフレームレートが大幅に落ちるため操作性が悪くなる。そのため、全てのポリゴンに対して処理を行わずに、操作点に近い部分のポリゴンを検出して衝突検出を行うようにすればフレームレートは改善できると考えられる。また、Bounding Boxによる衝突検出とポリゴンによる衝突検出を組み合わせることで、詳細なモデルを複数表示した場合でも、効率的にモデルの衝突検出を行うことができると考えられる。

謝辞

本研究を進めるにあたり、多大な協力をいただいた東京大学先端科学技術研究センター生命知能システム分野廣瀬研究室の研究生の皆さんに、この場をお借りして深く感謝の意を表します。

文献

- [1] Katsutoshi Fujii, Yoshinao Asano, Naoki Kubota, Hideki Tanahashi, "User Interface Device for the Immersive 6-screens Display 'COSMOS'", 'VSMM2000', pp.275-282, 2000
- [2] 北村喜文, "衝突検出を利用した仮想物体操作," ヒューマンインターフェース学会論文誌, vol.1, No.4, pp.25-32, 1999
- [3] 筧直之, 矢野博明, 齋藤允, 小木哲朗, 廣瀬通孝, "没入型仮想空間における力覚呈示デバイス HapticGEAR の開発とその評価," 日本バーチャルリアリティ学会論文誌, vol.5, No.4, pp.1113-1120,
- [4] Jed Hartman, Josie Wernicke, VRML2.0 ハンドブック-Web上の世界に動きを, アシソン・ウェスレイ・パブリッシャーズ・ジャパン(株)1997
- [5] Rikk Carey, Gavin Bell, 注解 VRML2.0 リファレンスマニュアル, 星雲社, 1998
- [6] <http://kslab-www.pi.titech.ac.jp/study/spidar-j.html>