

3次元オブジェクトの空間的關係に基づく検索機能を有する 空間データベースの設計と実装について

猪飼 瑞絵^{1,a)} 金 鎔煥¹ 片山 喜章¹

概要：2次元や3次元オブジェクトを管理するためのデータベースを空間データベースという。特に2次元オブジェクトに関しては地図などの時空間情報処理分野で盛んに利用されている。既存の主な空間データベースにおいては、2次元空間上でのオブジェクト間の空間的關係を求める機能（空間関数）が多く用意されている一方で、3次元空間での空間関数は充実しているとは言えない。そこで本研究では、3次元空間上のオブジェクトを管理する3次元空間データベースを構築し、オブジェクト間の空間的關係を求める機能を実現する。また、提案空間データベースを家電制御システムに適用した事例を紹介する。

A Design and Implementation of 3-D Spatial Database with Search Functions Based on Spatial Relations between 3-D Objects

MIZUE IKAI^{1,a)} YONGHWAN KIM¹ YOSHIAKI KATAYAMA¹

1. はじめに

従来のデータベースでは、文字や数字といったデータを格納するように設計されているが、近年では画像や音声といったそれ以外のデータを登録できるデータベースが登場している。その一つに2次元や3次元オブジェクトを管理するためのデータベースである空間データベースがある[1], [2], [3]。空間データベースは主に地図などを扱う時空間情報処理分野で盛んに利用されている。しかし、既存の主な空間データベースにおいては、2次元オブジェクトに対して演算などを行う機能（空間関数）が多く用意されている一方で、3次元オブジェクト間での空間関数は十分に実装されていない。

立体地図[4]のような3次元の地理データなど、時空間情報処理分野でも3次元空間で地図や地理情報を利用することが増えている。そして、そういった3次元空間を含む情報を用いて車の自動走行への適用や防災・減災への適用などが提案されている。たとえば、名古屋大学、長崎大学、産

業技術研究所などが共同開発したAutoware[5]という自動運転ソフトウェアでは、主要機能の一部に3次元位置推定や3次元地図生成といった機能を備えている。実際にこのソフトウェアを用いた公道実証実験が行われている。

また、スマートフォンで撮影した画像から3Dモデルを生成するアプリなど、これまでと比べて簡単に3次元オブジェクトを生成できるようになっている。加えて、GoogleがPoly[6]という無料の3次元オブジェクトのためのライブラリを公開しているが、こういった3次元オブジェクトのライブラリや、そのライブラリを利用して作成されたサービスやものは増えていくと予想できる。これらの事実より、今後3次元オブジェクトを扱う需要は高まっていくと考えられる。

そこで本研究では、以下の機能を持った3次元空間上のオブジェクトを管理する3次元空間データベースを構築し、オブジェクト間の空間的關係を求める機能を実現する。

- (1) ある空間に置かれた3次元オブジェクトに関して、その種類、位置、方向、形状などを記録可能な空間データベースを構築する機能を有する。
- (2) ユーザが指定した、空間データベースに登録されているオブジェクト同士の空間的關係を求める機能を有

¹ 名古屋工業大学大学院
Graduate School of Engineering, Nagoya Institute of Technology, Nagoya, Aichi 466-8555, Japan
^{a)} m.ikai.301@nitech.jp

する。

- (3) 空間データベースに登録されているオブジェクトと、ユーザから与えられた3次元オブジェクトとの空間的關係を求める機能を有する。

空間データベースに3次元オブジェクト間の空間的關係を求める空間関数が実装されることで、例えば、予めオブジェクトが登録されたある空間中にカメラを設置した場合に、そのカメラに映りこむオブジェクトを検索したり、ある部屋のテレビに接する家電を検索するといったことが可能になる。このように、空間データベースの高機能化によってさまざまなアプリケーションが容易に実現可能となる。

本研究報告の構成は以下のとおりである。2章では関連研究について述べ、3章で提案システム説明する。また、4章では提案空間データベースを使用した例として、家電制御システムに適用した事例を紹介する。

2. 関連研究

2.1 空間データベース

空間データベースとは、データベースの一種で、主に地理情報などを扱うために時空間情報処理分野で盛んに利用されているデータベースである。空間データベースでは、空間オブジェクトを格納し、それに対してクエリを実行してレコードを取得することができる。ここでいう空間オブジェクトとは、点、線、複数の直線で構成される多角形(ポリゴン)、およびこれらの集合を表す。主な空間データベースとして、MySQL[1]、Spatialite[2]、PostGIS[3]などがある。

2.1.1 MySQL

MySQLとは、世界で最も多く使用されているオープンソース・データベースである。点、線、ポリゴンといった空間データ型が用意されているが、2次元オブジェクトのみ空間データ型として扱っており、空間関数も用意されているが、3次元オブジェクトでは扱えない。

2.1.2 Spatialite

Spatialiteとは、SQLite[7]を空間データベースとして扱える拡張機能である。2次元オブジェクトの登録や、それらに対する空間関数は用意されている。3次元オブジェクトに対しては、3次元情報(Z座標)を登録することができる。しかし、この2次元情報は空間関数の際には利用されず、3次元オブジェクトに対する空間関数は存在しない。

2.1.3 PostGIS

PostgreSQL[8]の拡張で、空間オブジェクトを格納できる。2次元の空間オブジェクトと3次元の空間オブジェクトそれぞれを登録すること、また、それらに対する空間関数も用意されている。しかし、PostGISの空間関数を見ると、3次元情報を無視しないで使える空間関数はPostGISの空間関数全体の半数であり、その中でも空間的關係を求める空間関数に限定するとその割合はもっと少なく、十分とはいえない。

2.2 関連研究

関連研究として、空間的關係に基づく視点位置可視化機構の実現方式[9]がある。この研究では、2つのオブジェクト間の空間的關係を全方位から計算し、それを2つの3次元オブジェクトの周りを囲むような球体に色分けをして提示する機能を実現している。

直感的には、2つの3次元オブジェクトを透明な球体内に設置し、その球面の任意の場所からオブジェクトを観測すると考える。このとき、その球面上の点から観測した場合のオブジェクト間の關係を色分けすることで表現している。

図1に例を示す。四角であらわされている2つの3次元オブジェクトに対して、薄い灰色の範囲から見ると2つのオブジェクトは完全に重なっているように見える。また、濃い灰色の範囲から見ると空間的關係は一部重なっているように見え、白色の範囲では重なっていないように見える。このようにして、ある視点位置から見たときの空間的關係がわかりやすく球体表面に可視化される。



図1 可視化の出力例

この研究では、ある視点から2つのオブジェクトを観測した場合の空間的關係を求める研究である。一方で、それら2つのオブジェクトの直接的な空間的關係(接触している、重なっているなど)を求めているわけではない。もちろん、様々な視点からの観測結果から直接的な空間的關係を計算可能だと考えられるが、この方法は効率的とは言えない。

2.3 空間的關係

空間的關係とは、2つのオブジェクトが接している、離れているなどの位置關係を表す。本研究では、空間的關係の評価尺度としてEgenhoferの9交差行列[10]の考え方をを用いる。Egenhoferの9交差行列とは、各2次元オブジェクトを点の集合と扱い、それぞれの交差關係を3×3の行列で表したものである。2つのオブジェクトをオブジェクトAとオブジェクトBと置き、オブジェクトAの境界、内部(境界を含まない)、外部(境界を含まない)領域をそれぞれ

∂A , A° , A^- , またオブジェクト B に対しても同様に ∂B , B° , B^- と置く. このオブジェクト A, B の関係をこれら境界, 内部, 外部領域を用いて以下のように示す.

$$M = \begin{pmatrix} \partial A \cap \partial B & \partial A \cap B^\circ & \partial A \cap B^- \\ A^\circ \cap \partial B & A^\circ \cap B^\circ & A^\circ \cap B^- \\ A^- \cap \partial B & A^- \cap B^\circ & A^- \cap B^- \end{pmatrix} \quad (1)$$

行列のそれぞれの値は, 共通部分がないことを意味する空集合 ($\emptyset$) か, 共通部分が存在することを意味するそれ以外 ($-\emptyset$) のどちらかの値をとる.

この 9 交差行列では論理的には 512 通りの組み合わせがあるが, 領域同士での空間的關係は, 現実には以下の 8 種類のどれかが考えられる (図 2). またそれらを示す Egenhofer の 9 交差行列は図 3 のようになっている.

Disjoint 2つのオブジェクトがお互いの外部にあり, 接していない

Equal 2つのオブジェクトが同じオブジェクトである

Contain オブジェクト B がオブジェクト A の内部にあり, かつ接していない

Inside オブジェクト A がオブジェクト B の内部にあり, かつ接していない

Covers オブジェクト B がオブジェクト A の内部でオブジェクト A に接している

CoveredBy オブジェクト A がオブジェクト B の内部でオブジェクト B に接している

Meet 2つのオブジェクトがお互いの外部にあり, 接している

Overlap 2つのオブジェクトが交わっている

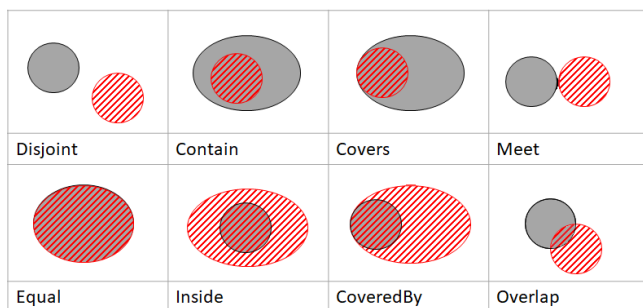


図 2 8 種類の空間的關係
黒:オブジェクト A 赤:オブジェクト B

$\begin{pmatrix} \emptyset & \emptyset & -\emptyset \\ \emptyset & \emptyset & -\emptyset \\ -\emptyset & -\emptyset & -\emptyset \end{pmatrix}$	$\begin{pmatrix} -\emptyset & -\emptyset & -\emptyset \\ \emptyset & \emptyset & -\emptyset \\ \emptyset & \emptyset & -\emptyset \end{pmatrix}$	$\begin{pmatrix} -\emptyset & -\emptyset & -\emptyset \\ \emptyset & -\emptyset & -\emptyset \\ \emptyset & \emptyset & -\emptyset \end{pmatrix}$	$\begin{pmatrix} \emptyset & \emptyset & -\emptyset \\ \emptyset & -\emptyset & -\emptyset \\ -\emptyset & -\emptyset & -\emptyset \end{pmatrix}$
Disjoint	Contain	Covers	Meet
$\begin{pmatrix} -\emptyset & \emptyset & \emptyset \\ \emptyset & -\emptyset & \emptyset \\ \emptyset & \emptyset & -\emptyset \end{pmatrix}$	$\begin{pmatrix} -\emptyset & \emptyset & \emptyset \\ -\emptyset & \emptyset & \emptyset \\ -\emptyset & -\emptyset & -\emptyset \end{pmatrix}$	$\begin{pmatrix} -\emptyset & \emptyset & \emptyset \\ -\emptyset & -\emptyset & \emptyset \\ -\emptyset & -\emptyset & -\emptyset \end{pmatrix}$	$\begin{pmatrix} -\emptyset & -\emptyset & -\emptyset \\ -\emptyset & -\emptyset & -\emptyset \\ -\emptyset & -\emptyset & -\emptyset \end{pmatrix}$
Equal	Inside	CoveredBy	Overlap

図 3 8 種類の空間的關係を示す 9 交差行列

3. 提案システム

3.1 3 次元空間における空間的關係

本研究では, 複数のポリゴンからできる多面体の 3 次元オブジェクトを考える. そして 3 次元オブジェクトを成すポリゴンの集合を境界とし, それらで囲まれた空間を内部, その他を外部として, Egenhofer の 9 交差行列を用いる. 3 次元空間における空間的關係は図 2 のようになっている.

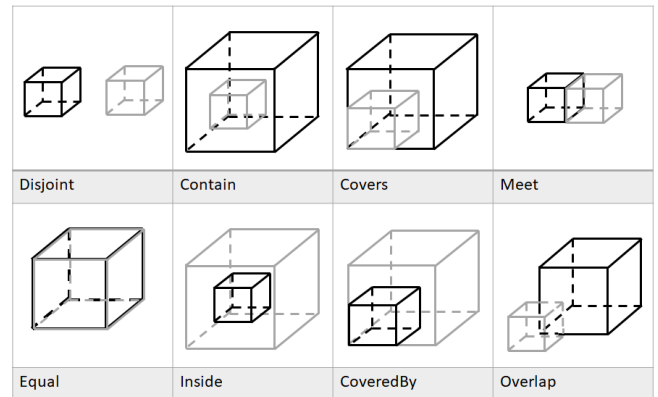


図 4 3 次元空間における空間的關係
黒色:オブジェクト A 灰色:オブジェクト B

主な空間データベースにおいて, 2 次元オブジェクトに対して 8 種類の空間的關係を求める空間関数は用意されていたり, 他の空間関数との組み合わせで実現することができる. しかし, 3 次元オブジェクトに対する空間関数に関しては, PostGIS にいくつかの空間関数が存在するものの, 空間的關係を求める空間関数は存在していない. 例えば, オブジェクト間の距離を求める空間関数やあるオブジェクトが別のオブジェクトから一定の距離以内にあるか判定する空間関数は存在するが, Contain となるか判定する関数などは用意されていない. そのため本研究では, いくつかの空間関数を用いて, 2 つの 3 次元オブジェクト間の空間的關係を返す空間関数を実装する. また以下の点から, 空間データベースとして PostGIS を利用し, 空間的關係を計算する機能を拡張することで提案システムを実装した.

- (1) 3 次元オブジェクトを扱うことができる
- (2) 3 次元オブジェクトに対する空間関数が少ないものの用意されている
- (3) 空間関数の実装などの拡張が容易である

3.2 PostGIS の拡張

PostGIS を拡張することで, 空間的關係を求める空間関数を有した空間データベースを実装する.

3 次元オブジェクトとして多面体を考える. PostGIS を用いて多面体を作成する際には, 複数のポリゴンからなるマルチポリゴンという形で作成をすることになる. このとき, 多面体の中は空洞となっており, 2 つの 3 次元オブジェ

クトの関係をみたときに、内部にあるか外部にあるか判定する空間関数は存在しない。そのため、3次元空間上での空間的関係を求めるために以下の3つの判定を行う空間関数が必要だと考えた。

Intersects(A, B) オブジェクト A と B に共通部分があるかどうか判定する空間関数

Inner(A, B) オブジェクト A の内部に B の頂点のどれか1つでも存在しているかどうか判定する空間関数

Equals(A, B) オブジェクト A と B が同じオブジェクトかどうか判定する空間関数

ここで、オブジェクト A を成す n 個のポリゴン $A_1, \dots, A_n$ とし、オブジェクト B を成す m 個のポリゴン $B_1, \dots, B_m$ とする。また、オブジェクト A を成す k 個の頂点を $V_1, \dots, V_k$ とし、同様に、オブジェクト B を成す l 個の頂点を $v_1, \dots, v_l$ とする。

3.2.1 Intersects()

オブジェクト A とオブジェクト B に共通部分が存在するかどうか判定するために、PostGIS の既存の関数である ST_3DIntersects を用いる。

3.2.2 Inner()

オブジェクト A の内部にオブジェクト B の頂点が含まれるかどうかを判定するために、本研究では内外判定アルゴリズムである交点計算法 [11] を用いる。

交点計算法 m 個の面からなる多面体の j 番目の面を $F_j (1 \leq j \leq m)$ とし、与えられた与点を $G(G_x, G_y, G_z)$ とする。3次元空間で直線 $y = G_y$ と多面体の面 F_j を含む無限平面との交点を調べる。もし交点が存在するならば、交点数を1増やす。この交点数の総和が奇数であれば点 G は与えられた多面体の内部に存在し、もし偶数であれば点 G は多面体の外部に存在する。

多面体であるオブジェクト A に対して、オブジェクト B の各頂点 $v_1, \dots, v_l$ が内外どちらに存在するかを交点計算法を用いて計算する。オブジェクト A と交わるような十分長い線分を直線 $y = G_y$ の代わりに用いて、ポリゴン $A_1, \dots, A_n$ それぞれに対してその線分と交わるかどうかを ST_3DIntersects を用いて判定する。そうして各頂点が内外どちらにあるか判定したあと、ひとつでもオブジェクト A の内部に存在していたら真を返し、すべての頂点が外部に存在していれば偽を返す。

3.2.3 Equals()

同じオブジェクトであればポリゴンと頂点の数が等しいはずなので、 $n = m$ かつ $k = l$ となるか判定する。

$n = m$ かつ $k = l$ だった場合、次にオブジェクトをつくるポリゴンが同一かを調べる。そこで、キューを用いてポリゴンを成す頂点とその順を調べる。キューを用いる理由は、頂点の開始する位置が同じとは限らなく、かつ頂点の順が逆順の可能性があるのでその計算のしやすさのためキューを使う。なので、オブジェクト A とオブジェクト B のポリゴン

それぞれの頂点をキュー ($Q_{A1}, \dots, Q_{An}, Q_{B1}, \dots, Q_{Bm}$) に入れる。ポリゴン $A_1, \dots, A_n$ に対応するキュー $Q_{A1}, \dots, Q_{An}$ それぞれに対して、頂点の数と順番が一致する、もしくは頂点の数が一致し、かつ逆順にすると一致するキューをキュー $Q_{B1}, \dots, Q_{Bm}$ の中から探す。キュー $Q_{A1}, \dots, Q_{An}$ とキュー $Q_{B1}, \dots, Q_{Bm}$ がすべて1対1で対応するようであれば真を返し、そうでなければ偽を返す。

3.2.4 Intersects(), Inner(), Equals() による空間的関係の計算方法

上記の空間関数からなる以下のフローチャートを用いて、空間的関係を判定する (図5)。2つあるオブジェクトを A と B とする。

最初に、オブジェクト A とオブジェクト B が交わっているかどうか空間関数 Intersects() を用いて2つのグループに分ける。交わっているグループでは、次にオブジェクト A の内部にオブジェクト B の頂点が含まれているか、さらにオブジェクト B の内部にオブジェクト A の頂点が含まれているか空間関数 Inner() を用いて判定する。この時点で、Overlap, Covers, CoveredBy の関係となるものが判定できる。残りの Equal と Meet を区別するために空間関数 Equals() を用いる。

次に交わっていないグループでも同様に、オブジェクト A の内部にオブジェクト B の頂点が含まれているか、また逆にオブジェクト B の内部にオブジェクト A の頂点が含まれているか空間関数 Inner() を用いて判定する。これで残りの Contain, Inside, Disjoint の関係となるものが判定できる。

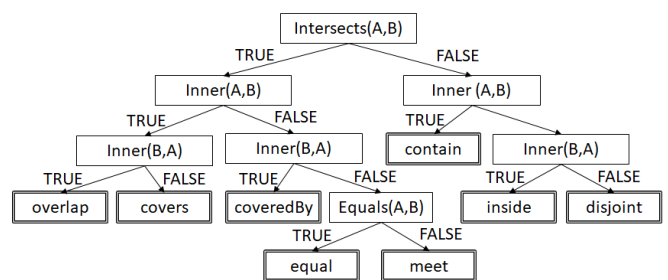


図5 空間的関係を求めるフローチャート

これらの演算や判定は、PL/pgSQL を用いて記述し、空間データベースに実装する。PL/pgSQL とは、PostgreSQL データベースシステム用の読み込み可能な手続き言語である。PL/pgSQL を用いることで、繰り返し処理や条件分岐などの制御構文が利用でき、また SQL の問い合わせや空間関数も用いることができるので、それらを利用した関数をユーザが作成することができる。

4. 家電制御システムへの適用

4.1 ジェスチャを用いた家電制御システム

家電制御システムとは、リモコンやボタン、ジェスチャーな

どの手段を用いて家電を操作するシステムのことである。その中でも、指差しのジェスチャを用いた家電制御システムを考える。ジェスチャを用いた家電制御システムでは、家電を選択し、その家電に指示を送るという手順で操作が行われると考える。中でも家電を選択する部分に注目すると、以下のような問題点が挙げられる。また、指差しのジェスチャを用いた場合、ユーザの指差したものがなにかははっきりわからないという問題がある。

(1) ユーザ位置の特定

(2) ユーザの指差方向の認識

(3) ユーザが指差した家電の認識

1,2に関しては、画像処理などで認識可能となってきた。

3の問題点を解決するために、

- 空間のどこにどの家電が設置されているか認識する
- 指差された方向に存在する家電を検索する

が必要になる。

さらに、指差しを用いたジェスチャの場合だと、方向などに誤差が生じる可能性があるため、これに配慮する必要もある。このとき、線分ではなく円錐や円柱といった形状を用いることで、誤差を吸収した検索を行うことができる(図6)。たとえば指の角度がぶれることで発生する誤差は円錐の形状を用いることで吸収でき、また、指自体が前後左右にぶれることで発生する誤差は円柱の形状を用いることで吸収できると考えられる。

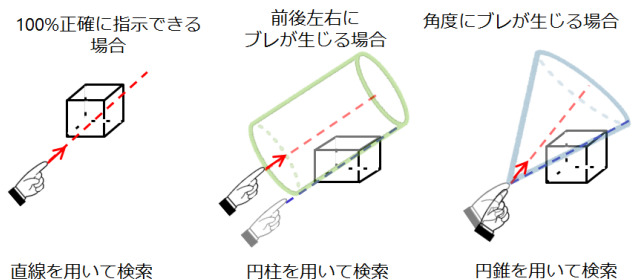


図6 ジェスチャを用いた際に発生する誤差と吸収方法

そのため、登録されているオブジェクトと円錐のような検索に使用しているオブジェクトの空間的関係を求め、Contain や Inside のような関係となるものがあるかどうかを考えれば、誤差を吸収した検索が行えると考えた。

そこで今回、部屋に置かれた家電を3次元オブジェクトとして考え、その3次元オブジェクトからなる空間データベースを構築し、それを室内空間データベースとする。そして、どの家電が空間(室内)のどこに設置されているか認識するためにいくつかの機能を実装し、その中で空間的関係を求める空間関数を利用する。今回家電と家電がおかれている部屋は、簡略化のために直方体で表すものとする。

4.1.1 家電登録機能

家電登録機能とは、家電の大きさと基本的な情報を登録

する機能である。大きさは家電の横幅、奥行、高さのそれぞれの長さで決定する。そして、基本的な情報として家電の名前とその種類も登録をする。

4.1.2 家電配置機能

家電配置機能は、どの家電がどこに置かれているかを登録し、その情報から3次元オブジェクトを生成し、登録する機能である。ユーザから配置したい家電の家電名と、基準となる配置位置と回転角を受け取り、登録された大きさの情報も用いて、家電を示す直方体の8頂点の座標をそれぞれ求める。基準となる配置位置は、ユーザが家電を小絵面から見たときの左下手前にあたる頂点の位置の座標である。その8頂点の座標から3次元オブジェクト(マルチポリゴン)を生成し、それを室内空間データベースに3次元オブジェクトとして登録する。

4.1.3 家電検索機能

家電検索機能では以下のような手順を用いて、円錐などの形状を用いた際にそれと交差する、または含まれるような家電オブジェクトを検索する。

- (1) ユーザは線分や円錐などの検索に使用したい形状とそれらの値を与え、それをユーザ指定オブジェクトとする。
- (2) 室内空間データベースに含まれる3次元オブジェクトをオブジェクトA、ユーザ指定オブジェクトをオブジェクトBと考え、それらに対して空間関係を求める。
- (3) 求めた空間的関係の中で、Meet, Equal, Inside, CoveredBy, Overlap のいずれかの空間的関係となるものを、検索結果として出力する。

4.1.4 実装例

Javaを用いて、家電の登録・検索を行うことができるGUIを作成する。GUIは文字を用いた家電の登録と配置、検索を行う部分(図7)と、3Dで検索結果を表示する部分(図8)に分けられる。文字を用いた部分ではJFrameを、3Dでの結果表示にはJava3Dを用いる。

検索に用いるユーザ指定オブジェクトの形状として、線分、円柱、円錐、球を実装する。線分は始点と終点、円柱は始点と終点と円柱の半径、円錐は始点と終点と底面の半径、球は中心点と半径の値をユーザから受け取る。線分、円柱、円錐に関しては、終点の座標が部屋の外部となるように、終点側を延長する。また検索の結果については、ユーザからの距離やユーザ指定オブジェクトの中心からの距離でソートを行える。

5. おわりに

主要な空間データベースにおいて、3次元オブジェクトに対する空間関数は2次元オブジェクトに対する空間関数に比べて少なく、十分とは言えない。しかし、3次元オブジェクトを扱う需要は今後増えていくと予想される。そこで、PostGISを用いて3次元空間での空間的関係に基づく検

d_id	家電名	家電の種類	幅	奥行	高さ	p_id	基準点のx座標	y座標	z座標	回転角
6	空気清浄機	空気清浄機	37	28	60	1	28	200	40	0
7	デッキ1	DVDデッキ	40	50	5	2	325	400	28	0
8	デッキ2	DVDデッキ	35	50	5	3	327	400	33	0
10	PC2	PC	45	20	40	4	75	15	70	0
11	PC3	PC	60	50	50	5	500	440	75	0
9	テレビ2	テレビ	96	50	64	6	310	450	50	0
14	蛍光灯2	電灯	80	80	15	7	150	210	215	0
14	蛍光灯1	電灯	80	80	15	8	575	210	215	0
15	エアコン1	エアコン	60	150	5	9	500	250	225	0

図 7 文字を用いて家電の登録・配置・検索を行う部分

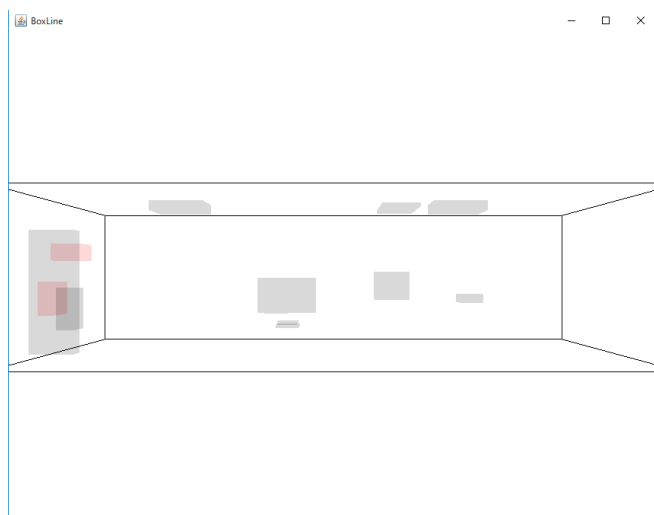


図 8 3D で結果表示を行う部分

索機能を有する空間データベースの設計と実装を行った。空間的關係は PostGIS の空間関数である ST_3DIntersects と, Inner, Equal という作成した空間関数を用いたフローチャートを用いて求めることを行った。

また, 空間的關係を求める機能を有する空間データベースを使用した実例としてジェスチャを用いた家電制御システムへの適用も紹介した。

参考文献

- [1] MySQL 入手先 (https://www.mysql.com/jp/)
- [2] SpatiaLite 入手先 (https://www.gaia-gis.it/fossil/libspatialite/index)
- [3] PostGIS 入手先 (PostGIS http://postgis.refractory.net/)
- [4] 国土地理院「立体地図 (地理院地図 3D・触地図)」入手先 (https://maps.gsi.go.jp/3d/)
- [5] Autoware 入手先 (https://www.pdsl.jp/fot/)

- [6] Poly 入手先 (https://poly.google.com/)
- [7] SQLite 入手先 (SQLite https://sqlite.org/)
- [8] PostgreSQL 入手先 (PostgreSQL https://www.postgresql.org/)
- [9] 木村, 他. 空間的關係に基づく視点位置可視化機構の実現方式., DBSJ Letters, Vol. 2, No. 2, 2003.
- [10] Max J.Egenhofer et al. "Categorizing binary topological relationships between regions, lines, and points in geographic databases." Technical report, Department of Surveying Engineering, University of Maine, Orono, ME, 1992.
- [11] 長島. 多角形・多面体の内外判定アルゴリズム., 図学研究 22, II1-II4, 1988.