

D-024

データベースによって制御する物体探索ロボットの 障害物を考慮した経路決定と移動制御

Obstacle-Aware Route Calculation and Movement Control for Object Retrieval Robots Controlled By Database

草野 弘行†
Hiroyuki Kusano

富井 尚志‡
Takashi Tomii

1. はじめに

近年、情報技術の発達により、コンピュータやネットワークの高速化、低価格化が進み、無線 IC タグ (RFID) 技術や加速度センサなどの、実世界の状態を検知できる様々なセンサ技術が普及している。また、u-Japan 政策[1]の推進など、生活環境に溶け込んだコンピュータ支援への関心が高まっている。このような様々な情報技術の恩恵を“いつでも・どこでも”受けられる「ユビキタス環境」が現実味を帯び始めている。

RFID リーダは、非接触での読み取り、耐障害物性、複数同時読み取り可能という特徴を持ち、RFID タグは小型・安価で、物体への貼り付けが容易である。このような特徴は、実空間の物体の状態の検知に有効であり、利用者の特別な入力作業なしに、物体の所在や、おかれている状態をセンサデータから自動的に読み取ることが可能となる。

しかし、意図に反して、物体が RFID リーダの検知範囲から放り出されてしまった場合、この物体は RFID リーダによって検知する事ができず、ユビキタス支援に活用する事ができない。このような問題への対策として、センサの数を増やす事や多種類のセンサを組み合わせる事が考えられるが、その構築や維持に掛かるコストは膨大なものとなる。

そこで我々は、センサを装備し、空間内を自律的に探索する物体探索エージェントによって、このような問題を解決することを考えた。探索する場所 (探索地点) の候補となる、行方不明となった物体が最後に検知された場所や、その物体が頻繁に検知された場所などは、ユビキタス環境データベースに蓄積された情報から取得することが可能である。この情報に基づいて物体探索エージェントの探索を制御する事ができる。一方で、エージェントによる探索の結果に基づいて、データベースを更新することができる。

ここで、このようなエージェントの実現手段として、RFID リーダを装備し、自律的に移動が可能なロボットを用いることを考える。このとき、物体探索エージェントを実現するためには、探索ロボットを探索地点まで移動させるための経路を決定すること、その経路をもとにロボットを移動させることは不可欠な要素である。そこで本研究ではこの問題を「経路決定」と「移動制御」の二つに分けた。探索ロボットが探索地点へ到達するための移動経路を、ユビキタス環境データベースから得られる仮想空間を用いて決定した。この経路に従って移動させる際に必要となる、探索ロボットの現在位置を、ランドマークタグを検知することによって取得し、それに基づいて移動制御を行った。

以上について設計と実装および実験による評価を行った。

2. ユビキタス環境

安価で高速なコンピュータや様々な空間センサが普及し、これらを用いて実現するユビキタス支援の研究が数多く行われている。Tagged World Project[2]では、部屋内の様々な場所に配置された RFID タグと人が携帯するリーダによって人の行動を認識し、その意図を推定することが行われている。また、我々は RFID センサによって取得したセンサデータから実空間の物体の状態を取得し、それらを蓄積するユビキタス環境データベースを持つ「概念共有環境 CONSENT(CONcept Sharing Environment)」を提案してきた[3]。

2.1 物体探索エージェントの導入

ユビキタス環境ではセンサによって得られた物体のすべての状態を蓄積 (データベース化) し、利用者への支援に反映させる事ができる。しかし、空間内のすべてのリーダの検知範囲から外れてしまった物体は検知する事ができない。例として、机の上で検知されていた本が、何らかのアクシデントによって床に落下した場合が挙げられる。このような問題を解決するための手段としてセンサの数を増やす事や多種類のセンサを組み合わせる事が考えられるが、様々な場所で多数の人や物体を検知するためには数多くのセンサが必要になり、その構築や維持にかかるコストは膨大なものとなる。

そこで我々は、自律的に空間内を移動し、利用者環境の支援を行うロボットに注目した。このようなロボットは安価に利用できるようになり、情報機器とのインタフェースや視覚障害者のナビゲーションとして人間の日常生活のサポートに用いられている[4]。ユビキタス環境では様々なセンサが用いられており、ロボットがセンサを用いて様々なサービスを行うことに適していると考えられる。そこでユビキタス環境の状態を蓄積、データベース化するために、動くセンサとして自律的に移動し物体探索を行うエージェントを導入した。

2.2 Control By Database

行方不明となった物体を探索するためには、実際に探索を行う場所である探索地点の見当をつける必要がある。行方不明の物体の位置の推定候補として、その物体が最後に検知された場所やその物体が頻繁に置かれた場所などが挙げられる。このような情報は、ユビキタス環境データベースに履歴として蓄積されている情報から、検索によって取得する事ができる。

このようにして取得した情報に基づいて探索地点を決定

† 横浜国立大学大学院環境情報学府
情報メディア環境学専攻

‡ 横浜国立大学大学院環境情報学研究院

し、物体探索エージェントの探索活動を制御する。そして、探索によって得られた物体の情報に基づいて、データベースを更新する。その結果、データベースに蓄積された情報はより正確になり、物体探索エージェントに新たな探索地点を与える事ができる。

この手法はユビキタス環境データベースに蓄積された情報を基にエージェントを制御するものである。我々はこれを Control By Database(CBDB)と定義した[5]。

2.3 物体探索エージェントの制御

エージェントは、探索しようとしている物体（探索物体）が見つかると思込まれる場所（探索地点）へ実際に赴く事で物体探索を行うことができる。そのとき、探索地点への移動経路計画は次の二つの要求事項を満たす必要がある。

ひとつは探索物体が「存在しそうな」場所を探索地点とすることであり、もうひとつは探索地点に効率的に移動することである。解決手法として、探索物体や障害物、移動路を地図化したものを別途作成して使用することが考えられる。しかし、このような地図の作成は手間が掛かる作業であり、特に、大量の物体が存在し、変化が多様であるユビキタス環境においては、実空間と地図の整合性の維持が困難である。また、エージェントは決定した移動経路に従って移動するために、自らの現在位置を取得する必要がある。

そこで、ユビキタス環境をデータベース化し、データベースに蓄積された空間情報とエージェントに取り付けたセンサを用いて、経路決定と移動制御を行うことを考えた。ユビキタス環境データベースに問い合わせる事で、その空間を表す仮想空間を容易に取得する事が可能である。これを用いて決定した経路に従って、エージェントの現在位置をセンサによって取得しつつ探索地点まで移動制御する事ができる。

3. 物体探索エージェントの設計

物体探索エージェントの設計に当たり、以下のような方針を立てを行った。

方針1. エージェントの移動は前進、旋回といった基本的かつ簡単な動きのみで制御する。

方針2. エージェントが持つセンサは自己の位置の特定や物体の検知に用いる必要最低限とする。そしてそのセンサはRFIDとする。

方針3. エージェントはユビキタス環境の空間情報を管理しているデータベースサーバと通信可能で、そこに蓄積されているデータに基づいて制御される。

これらの方針によって、高度な移動性能やセンサを必要とせず、どのような種類のロボットでも実現が容易となる。また、エージェントはデータベースに蓄積された空間情報を活用することができ、一方で、探索の結果に基づいてデータベースを更新することが可能となる。これらの方針に基づいて、物体探索手法を図1のように設計した。

本稿では図1に示した手順の3,4に該当する、探索地点への経路決定および探索地点への移動制御を扱う。以下で経路決定、移動制御の設計について説明する。

3.1 経路決定の設計

探索地点決定後、エージェントを現在地から探索地点まで移動させるための経路を決定する。その際に、経路上の障害物の有無をデータベースに蓄積されている物体の形状

情報を用いて判定する。手順としては以下ようになる。

手順1. 空間内の物体の座標・形状を表すシーングラフとランドマークタグの配置情報をデータベースより取得する。

手順2. 障害物の有無を判定し、探索地点までの最短経路を決定する。

手順1ではエージェントの活動場所に応じて適切なシーングラフを取得する。例えば、机の上で活動するエージェントの場合は、取得されるシーングラフは机の上を再現したものとなる。これはデータベースの検索により柔軟に取得可能であり、エージェント専用のものを別途作成する必要はない。また、エージェントの位置を把握するためのランドマークタグの情報を取得する。ランドマークタグの配置方法は様々な手法が考えられるが、本設計では床に一定間隔で格子状に配置する手法（グリッドタグ）を用いた。グリッドタグを用いる理由として、配置が容易であるという点、エージェントの移動可能範囲を均質に網羅できるという点が挙げられる。

続いて、エージェントの形状を考慮してシーングラフの軽量化を行う。エージェントの背丈を越える位置にある物体はエージェントが移動する際の障害物にはならないため、このような物体をシーングラフから取り除く。このようにシーングラフに制限を加える事で、エージェントの形状に適したシーングラフの軽量化をすることができる。

手順2では、エージェントの移動経路を決定する。方針に基づいた設計により、エージェントの現在位置はRFIDリーダが検知するタグによってのみ判断できる。ゆえに移動中はグリッドタグを適宜検出して現在位置を確認しながら制御する事が望ましい。以上の理由により、経路の最小単位は隣接する縦、横、斜めの二つのタグを結ぶ線分とし、これを探索地点まで接続する。障害物の有無の判定は、エージェントの移動軌跡上に上述のシーングラフのオブジェクトが存在するかどうかで決定する。この判定を経路の各々の最小単位に適用し、通行可能又は通行不可能のいずれかをそれぞれの最小単位に割り当てる。シーングラフに対して行う障害物判定は負荷の掛かる処理であり、できるだけ回数を少なくする事が望ましい。そこでA*アルゴリズム[6]を適用し、効率的に経路を導出する。

以上の方針に基づいた経路の例を図2に示す。

1. 探索しようとする物体（探索物体）の検索
2. 探索する場所（探索地点）の取得
3. 探索地点への最短経路の導出と探索順の決定
4. 探索地点へ移動
5. 探索地点近傍の探索
6. 発見した物体をデータベースに更新

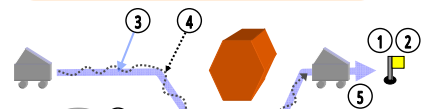


図1 物体探索の流れ

- グリッドタグ
- 経路の最小単位
- 移動可能(障害物なし)
- × 移動不可能(障害物あり)

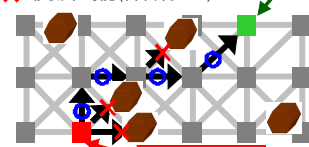


図2 経路決定の概要

3.2 移動制御の設計

経路決定後、エージェントをその経路にしたがって移動させる。どのような種類のロボットでも実現可能にするため、基本方針に従って前進、旋回のみで移動制御を行う。

エージェントの移動制御は、3.1 節の手順によって決定した移動経路に従って行う。移動制御のフローチャートを図 3 に示す。エージェントは、経路上のタグに沿って旋回・前進を行うことで、探索地点に到達する事ができる。しかし、エージェントの移動特性や環境が及ぼす影響によって、エージェントの移動に誤差が生じるため、必ずしも意図した場所に正確に移動するとは限らない。そのため、このような誤移動に対処する必要がある。

まず、経路から離脱したランドマークタグを検知した場合は、移動を一時停止し、その位置から探索地点への経路を計算し直す。そして、新たに決定した経路にしたがって移動を再開する。次に、エージェントが一定時間、グリッドタグを全く検知しなかった場合を考える。これは、タグとタグの間を長時間移動する、あるいは障害物のある方向へ前進を続けるといった状態に陥った時に発生する。タグを検知していないため、エージェントの現在位置は不明である。そこで、エージェントは自身の現在位置の探索を行い、エージェントの現在位置を決定する。

エージェントはそれまで移動してきた向きから方向転換、前進をし、複数のタグの検知を試みる。異なる二つのグリッドタグを検知した場合、その位置関係からエージェントの向きと位置を決定する。向きは二つのタグを通る直線から、位置は二番目に検知したタグの位置から求める。以上で決定した現在位置から探索地点への経路を計算し、移動を再開する。

4. 物体探索エージェントの実装

これまでに述べた設計に従って、我々の研究室で実際に運用しているユビキタス環境「概念共有環境 CONSENT」のプロトタイプにおいて物体探索エージェントの実装を行った。プロトタイプシステムに用いたデータベースの論理モデルを図 4 に示す。本モデルによって、実空間から取得、蓄積されたデータから、有用な情報をクエリによって取得することが可能となる。表 1 は有用な情報を取得するクエリの一つである、エージェントが必要とする空間内の物体配置情報のシーングラフの構成要素を取得するクエリである。エージェントは、RFID リーダを装備し探索地点へ赴いて物体探索を行う探索ロボットと、ロボットの動きを制御する制御プログラムによって実装した。

まず、制御プログラムは物体の配置情報とランドマークタグの配置情報をデータベースにクエリを発行して取得する。取得した配置情報から、シーングラフを構築する。また、探索ロボットの背丈を越える位置にある物体はエージェントの障害物にはならないため、このような物体を取り除きシーングラフの軽量化を行う。続いて、グリッドタグの配置情報をデータベースより取得する。これらの情報を基に、現在位置から探索地点への経路を決定する。

シーングラフの取得、障害物判定の処理によって決定された経路の例を図 5 に示す。エージェントは探索ロボットと、制御用 PC によって実装した。前進、旋回命令は制御用 PC から探索ロボットへの無線通信によって送信した。

5. 物体探索エージェントの評価

我々の研究室では 2006 年 9 月から約 4 ヶ月間、概念共有環境の日常実験を行った。本実験ではその概念共有環境データベースを用いた。

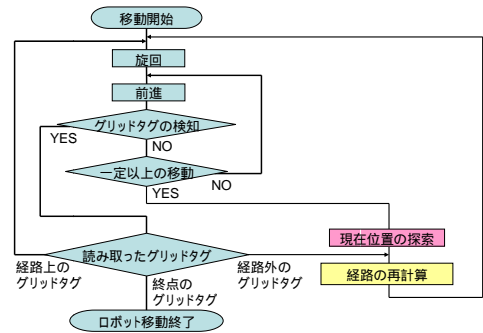


図 3 移動制御のフローチャート

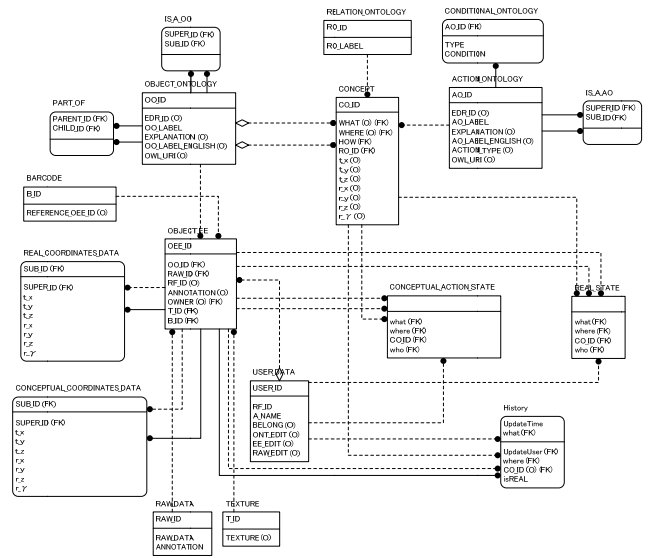


図 4 論理モデル

表 1 物体の配置情報を取得するクエリ

```
WITH cte (OEE_ID, OO_LABEL, RAW_ID, SUPER_ID, L_x, L_y, L_z, r_x, r_y, r_z, r_?, NODE) AS (
    SELECT OBJECT_EE.OEE_ID, OBJECT_ONTOLOGY.OO_LABEL, OBJECT_EE.RAW_ID, NULL,
           CAST(NULL AS float), CAST(NULL AS float), CAST(NULL AS float),
           CAST(NULL AS float), CAST(NULL AS float), CAST(NULL AS float), 0 AS NODE
    FROM OBJECT_EE
    JOIN REAL_COORDINATES_DATA ON OBJECT_EE.OEE_ID = REAL_COORDINATES_DATA.SUPER_ID
    JOIN OBJECT_ONTOLOGY ON OBJECT_EE.OO_ID = OBJECT_ONTOLOGY.OO_ID
    WHERE REAL_COORDINATES_DATA.SUB_ID = 293
    UNION ALL
    SELECT OBJECT_EE.OEE_ID, OBJECT_ONTOLOGY.OO_LABEL, OBJECT_EE.RAW_ID,
           REAL_COORDINATES_DATA.SUPER_ID, REAL_COORDINATES_DATA.L_x, REAL_COORDINATES_DATA.L_y,
           REAL_COORDINATES_DATA.L_z, REAL_COORDINATES_DATA.r_x, REAL_COORDINATES_DATA.r_y,
           REAL_COORDINATES_DATA.r_z, REAL_COORDINATES_DATA.r_?, NODE + 1
    FROM OBJECT_EE
    JOIN REAL_COORDINATES_DATA ON OBJECT_EE.OEE_ID = REAL_COORDINATES_DATA.SUB_ID
    JOIN OBJECT_ONTOLOGY ON OBJECT_EE.OO_ID = OBJECT_ONTOLOGY.OO_ID
    JOIN cte ON REAL_COORDINATES_DATA.SUPER_ID = cte.OEE_ID
    WHERE REAL_COORDINATES_DATA.SUPER_ID <> REAL_COORDINATES_DATA.SUB_ID
)
SELECT DISTINCT OEE_ID, OO_LABEL, RAW_ID, SUPER_ID, L_x, L_y, L_z, r_x, r_y, r_z, r_?, NODE
FROM cte
ORDER BY NODE
```

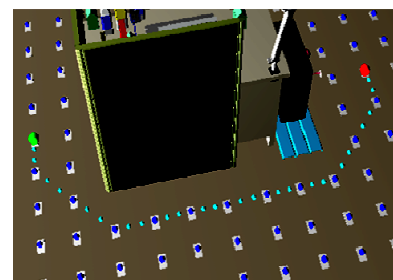


図 5 移動経路

実験環境は以下のとおりである。

物体探索エージェント(図6)

探索ロボット: SONY 社製 AIBO ERS-7M3, RFID リーダ: 富士通社製 RFID タグリーダ・ライタ F3972T110, 制御プログラム実行環境: ロボット制御用 PC から無線 LAN を用いて制御

データベースサーバ

機種名: Dell Precision 530, CPU: Intel Xeon 2.4GHz(DUAL 構成), Memory: 1GByte, OS: Microsoft Windows Server 2003, DBMS: Microsoft SQL Server 2005

ロボット制御用 PC

機種名: 自作構成, CPU: AMD Athlon XP 2500+ 1.82GHz, Memory: 1GByte, OS: Microsoft Windows XP Professional, 開発環境: Microsoft Visual Studio 2005, AIBO Remote Framework

グリッドタグ配置環境

タグ: オムロン社製パッシブタグ TG1FR003
配置範囲: 4.5m x 4m, 配置間隔: 25cm, 配置数: 358 枚

5.1 経路決定の評価

経路決定の評価を行うために、障害物がある場合と無い場合の二通りで経路の計算を行った。A*アルゴリズムを用いた場合と用いなかった場合、シーングラフの軽量化を行った場合と行わなかった場合のそれぞれの組み合わせで、障害物判定を行った回数と経路決定に掛かった時間を計測した。その結果は表2のようになった。

A*アルゴリズムの適用によって障害物判定の回数を削減し、シーングラフを軽量化することで一回の障害物判定に掛かる時間を削減した。また、障害物の有無に関わらず経路の決定が可能であり、専用の地図を別途用意する必要なしに、ユビキタス環境データベースによって経路決定を行うことが可能である事が示された。

5.2 移動制御の評価

移動制御の評価を行うために、始点・探索地点間に存在する一つの机を迂回する、道のり 4m の経路の移動を 10 回繰り返し行った。その結果は表3のようになった。

本実装で探索ロボットとして用いたロボットのように、ロボットの直進や旋回には誤差が発生する。この誤差が、経路離脱などの誤移動を発生させる要因となっている。本実験で用いたロボットは 10 回の実験で平均 3.4 回の経路離脱や、平均 0.1 回のランドマークタグ未検出が発生している。しかし、このような誤移動に対しても経路の再計算や現在位置の探索により、人手による修正を行わずにロボットを探索地点に到達させる事が可能である事が示された。

6. まとめ

本研究では、ユビキタス環境における物体探索エージェントの実現のために、障害物を考慮した探索ロボットの経路決定と移動制御を設計、実装し、評価実験を行った。

ユビキタス環境データベースによって得られる空間情報によって、障害物の存在する空間においても専用の地図を別途用意する必要なしに経路決定を行うことが可能であることを示した。また、RFID リーダとランドマークタグによる現在位置把握と経路離脱からの復帰手法により、探索ロボットの移動精度に依存せずに探索地点まで移動制御できることを示した。以上より、物体探索エージェントの実

現に、ユビキタス環境データベースと RFID タグ、リーダを用いる本手法の優位性を示した。

今後の課題として、空間内の物体やランドマークタグの配置、エージェントの種類などの様々な環境条件での評価、大量の探索対象を効率的に探索する手法の設計を行う必要があると考えている。

謝辞

本研究は平成 17 年度および 18 年度横浜国立大学大学院環境情報研究院共同研究プロジェクトの助成を受けて行った。

参考文献

- [1] 総務省, u-Japan 政策 HP, http://www.soumu.go.jp/menu_02/ict/u-japan/index2.html
- [2] 楓 仁志, 山原裕之, 藤原聡子, 野口豊司, 東 辰輔, 島川博光 “タグ付けられた世界における個人行動特性を用いた意図推測” 組込みソフトウェアシンポジウム 2005, Oct, pp.126-133, 2005.
- [3] 佐々木貴司, 富井尚志 “クエリフィードバックによる意味情報推奨機構を有する成長型ユビキタス環境データベース” Proc. of Data Engineering Workshop (DEWS2007) D7-9, March, 2007
- [4] Kulyukin V., Gharpure C., Nicholson J., Pavithran S. “RFID in Robot-Assisted Indoor Navigation for the Visually Impaired”, Proc. 2004 IEEE/RSJ Int. Conf. on Intelligent. Robots and Systems (IROS2004), pp.1979-1984, Sep. 2004
- [5] 那須洋之, 草野弘行, 富井尚志 “ユビキタス環境データベースによるエージェントの物体状況探索制御手法” Proc. of Data Engineering Workshop (DEWS2007) D7-6, March, 2007
- [6] 白井良明著, 斎藤信男, 有澤誠, 簡捷彦編, 人工知能の理論, コロナ社, 1992



図6 物体探索エージェント

表2 経路決定の実験結果

A*アルゴリズム		未使用		使用	
シーングラフ軽量化		無	有	無	有
障害物なし	判定回数	266	266	13	13
	時間(sec.)	42.31	25.48	2.17	1.31
障害物あり	判定回数	196	196	61	61
	時間(sec.)	31.14	18.88	9.98	6.13

表3 移動制御の実験結果

経路外のタグへ移動した回数 (経路再計算回数)	タグを見失った回数 (現在位置探索回数)	探索地点到達に 要した時間(sec.)
5	0	104
5	0	102
6	1	190
3	0	70
3	0	50
1	0	49
2	0	49
3	0	62
2	0	56
3	0	69
平均	3.4	80.1