The University of Osaka
Institutional Knowledge Archive

Title	ユビキタス環境における大規模コンテキストアウェアサービスのための基盤技術に関する研究
Author(s)	川上, 朋也
Citation	大阪大学, 2013, 博士論文
Version Type	VoR
URL	https://hdl.handle.net/11094/24946
rights	
Note	

The University of Osaka Institutional Knowledge Archive : OUKA

https://ir.library.osaka-u.ac.jp/

The University of Osaka

ユビキタス環境における
大規模コンテキストアウェアサービスのための
基盤技術に関する研究

提出先　　大阪大学大学院情報科学研究科

提出年月　2013 年 1 月

川上　朋也

関連発表論文

1. 学会論文誌発表論文

1. 川上朋也, 竹内 亨, 寺西裕一, 春本 要, 西尾章治郎, 下條真司: 地理的オーバレイネットワークを用いた位置依存コンテンツ分散管理手法, 情報処理学会論文誌, Vol. 52, No. 12, pp. 3289–3299 (Dec. 2011).

2. 川上朋也, 寺西裕一, 春本 要, 下條真司: センサデータの意味的解釈に基づく分散センサ情報管理システム, 情報処理学会論文誌, Vol. 54, No. 2 (Feb. 2013, 採録決定).

2. 国際会議等発表論文

1. T. Kawakami, S. Takeuchi, Y. Teranishi, K. Harumoto, and S. Shimojo: A P2P-based Mechanism for Managing Location-dependent Contents in Ubiquitous Environments, in *Proceedings of International Symposium on Applications and the Internet (SAINT 2007) Workshops*, pp. 54–57 (Jan. 2007).

2. T. Kawakami, B. L. N. Ly, S. Takeuchi, Y. Teranishi, K. Harumoto, and S. Nishio: Distributed Sensor Information Management Architecture Based on Semantic Analysis of Sensing Data, in *Proceedings of International Symposium on Applications and the Internet (SAINT 2008) Workshops*, pp. 353–356 (July 2008).

3. Y. Ishi, T. Kawakami, T. Yoshihisa, Y. Teranishi, K. Nakauchi, and N. Nishinaga: Design and Implementation of Sensor Data Sharing Platform for Virtualized Wide Area Sensor Networks, in *Proceedings of the 3rd International Workshop on Streaming Media Delivery and Management Systems (SMDMS 2012)*, pp. 333–338 (Nov. 2012).

4. T. Kawakami, Y. Ishi, T. Yoshihisa, and Y. Teranishi: A Delivery Method considering Communication Loads for Sensor Data Stream with Different Collection Cycles, in *Proceedings of the 28th ACM Symposium on Applied Computing (SAC 2013)* (Mar. 2013, to appear).

3. 研究会等発表論文（査読付）

1. 川上朋也, 三原慶彦, 寺西裕一, 春本 要, 下條真司: ユビキタス環境における位置依存情報の分散管理機構の設計と実装, 情報処理学会マルチメディア，分散，協調とモバイル (DICOMO 2006) シンポジウム論文集, pp. 325–328 (July 2006).

2. 寺西裕一, 川上朋也, 石 芳正, 春本 要, 下條真司: 位置コンテンツ共有システムの P2P モデルによる実装, 情報処理学会マルチメディア通信と分散処理ワークショップ (DPSWS 2006) 論文集, pp. 37–42 (Nov. 2006).

3. 川上朋也, Ly Bich Lam Ngoc, 竹内 亨, 寺西裕一, 春本 要, 西尾章治郎: センシングデータの意味的解釈に基づく分散センサ情報管理アーキテクチャ, 情報処理学会マルチメディア，分散，協調とモバイル (DICOMO 2008) シンポジウム論文集, pp. 1746–1753 (July 2008).

4. 川上朋也, 竹内 亨, 寺西裕一, 春本 要, 西尾章治郎: 地理的オーバレイネットワークの使い分けが可能な位置依存コンテンツ分散管理手法の提案, 情報処理学会マルチメディア通信と分散処理ワークショップ (DPSWS 2010) 論文集, pp. 81–87 (Oct. 2010).

5. 石 芳正, 川上朋也, 義久智樹, 寺西裕一, 中内清秀, 西永 望: 仮想化された広域センサーネットワークを構築可能とするプラットフォームの設計と実装, 情報処理学会マルチメディア，分散，協調とモバイル (DICOMO 2012) シンポジウム論文集, pp. 2344–2350 (July 2012).

6. 川上朋也, 義久智樹, 石 芳正, 寺西裕一: 収集周期の異なるセンサデータストリームのための通信負荷を考慮した配信手法, 情報処理学会マルチメディア通信と分散処理ワークショップ (DPSWS 2012) 論文集, pp. 136–143 (Oct. 2012).

4. その他の研究会等発表論文

1. 寺西裕一, 中内清秀, 西永 望, 石 芳正, 川上朋也, 濱口雄人, 義久智樹, 西尾章治郎: 大規模かつ柔軟なセンサー共有を実現する仮想統合センサーネットワーク, 情報処理学会研究報告, Vol. 2012-DPS-150, No. 1, pp.1–8 (Mar. 2012).

2. 川上朋也, 寺西裕一, 春本 要, 下條真司: 大規模コンテキストアウェアサービスのための地理的コンテンツ配信手法について, 情報処理学会研究報告, Vol. 2012-DPS-153, No. 4, pp. 1–8 (Nov. 2012).

3. 川上朋也, 義久智樹, 石 芳正, 寺西裕一: 収集周期からの負荷推定に基づくセンサデータストリーム配信手法, 電子情報通信学会技術研究報告, Vol. 112, No. 350, pp. 7–12 (Dec. 2012).

5. デモ・ポスター・展示

1. 石 芳正, 川上朋也, 義久智樹, 寺西裕一: 収集周期の異なるセンサデータストリームのための通信負荷を考慮した配信手法の一実装, 情報処理学会マルチメディア通信と分散処理ワークショップ (DPSWS 2012) デモ展示, pp. 121–123 (Oct. 2012).

2. Yoshimasa Ishi, Tomoki Yoshihisa, Tomoya Kawakami, and Yuuichi Teranishi: A Distributed Sensor Data Stream Delivery System with Communication Loads Balancing for Heterogeneous Collection Cycle Requests, in *Proceedings of the 18th IEEE International Conference on Parallel and Distributed Systems (ICPADS 2012) Posters and Demos*, pp. 728–729 (Dec. 2012).

以上

内容梗概

近年，PC などの固定端末に加え，携帯電話やスマートフォンなど，インターネットに接続可能なモバイル端末の利用が普及してきた．このように，インターネットに多様な機器が接続されることにより，ユーザがいつでもどこでも多様なサービスを享受できるユビキタス環境が発展してきた．ユビキタス環境におけるサービスでは，ユーザ自身が提供する情報に加え，センサデータも活用することが可能である．特に，天候などのセンサデータから実世界の状況（コンテキスト）を把握し，その状況に応じたサービスを提供するコンテキストアウェアサービスの実現が期待されている．

従来のコンテキストアウェアサービスは基本的にクライアント・サーバ型アーキテクチャで構築されており，また，多様なサービスがそれぞれ独自に構築されているため，サービス間でのセンサデータやコンテンツの共有が困難である．この問題を解決するためには，広域かつ膨大に発生するセンサデータやコンテンツ，さらには導出されるコンテキストをも共有可能な大規模コンテキストアウェアサービス構築のための基盤技術が必要である．また，クライアント・サーバ型アーキテクチャでは特定のサーバ群が膨大な数のセンサデータやコンテンツを管理することになり，サーバ機器や周辺ネットワークに負荷が集中することとなるため，規模拡張性に乏しい．したがって，そのような基盤技術は複数のコンピュータに情報の管理や処理を分散させる分散型アーキテクチャを基本とすることが必要不可欠である．

コンテキストアウェアサービスでは，ユーザやコンテンツのコンテキストをセンサデータから認識し，適切なコンテンツを提供する必要がある．特にセンサデータを含め，地理的な位置に関連する情報である「地理的コンテンツ」の検索は，大規模コンテキストアウェアサービスの基盤技術として重要な機能となる．しかし，従来提案されてきた分散型の地理的コンテンツ管理手法には，実際の位置と無関係にコンテンツが保存され，検索において発生するネットワークトラヒックが大きくなるなどの問題がある．また，コンテキストの認識においては，センサデータを意味的に解釈し，IF-THEN 形式で記述されたルールに基づいて推論することが有効だが，センサデータの解釈を分散構成で行う方法は，いまだ確立していない．単純な分散構成では，センサデータの不足により推論そのものができない，あるいは，同じセンサデータの解釈が複数回実行されて，冗長性が生じるなどの問題がある．さらに，得られたコンテキストに基づく処理においては，ルールに従った推論

処理を効率的に行う Rete アルゴリズムと呼ばれるアルゴリズムが提案されているが，地理的な条件照合も含め効率的にルールを分散処理する方法は確立していない．

本論文では，大規模コンテキストアウェアサービス実現のための分散型アーキテクチャを提案し，サービスにおける検索と推論それぞれについて，新たなアプローチによる解決法を提案する．まず，分散型アーキテクチャによる地理的コンテンツ管理手法の問題点については，コンピュータがもつ位置情報に基づいて構成する地理的オーバレイネットワーク層と，地理的コンテンツを管理する地理的コンテンツ管理層を分離し，任意の地理的オーバレイネットワーク上で構成可能な地理的コンテンツ分散管理手法を提案する．また，センサデータの意味的解釈を効率的に行うため，センサデータを扱う Raw Layer，センサデータの補間処理を行う Abstract Layer，コンテキスト導出処理を行う Logical Layer の 3 層からなる分散センサ情報管理アーキテクチャを新たに提案する．提案アーキテクチャでは，空間補間を行う機構を組み込むことで，センサの不在により推論そのものができない状況を回避する．また，ルールに基づく推論処理の結果をサービス間で共有にすることにより，冗長な推論処理を削減する．さらに，提案する分散センサ情報管理アーキテクチャにおいて，IF-THEN 形式のルールに基づく推論処理を効率的に行うため，Rete アルゴリズムを分散実行する手法を提案する．提案手法は，Rete アルゴリズムにおける各処理を，対象となる地理的コンテンツの位置やセンサデータの有効範囲に基づき異なるコンピュータ上に振り分けることで，冗長な処理を抑制しつつ，処理負荷を分散可能とする．

本論文は 5 章から構成され，その内容は次のとおりである．まず，第 1 章で序論として研究の背景および動機について述べる．第 2 章では，地理的オーバレイネットワーク上でのコンテンツ分散管理手法について述べる．また，提案手法の有効性の検証のために行ったシミュレーション評価の結果を示す．

第 3 章では，センサデータが存在しない場合のコンテキスト導出および冗長な処理の抑制を行う分散センサ情報管理アーキテクチャを提案する．また，提案手法の実装について述べるとともに，有効性の検証のために行ったシミュレーション評価の結果を示す．

第 4 章では，地理的コンテンツやセンサデータを地理的な情報に基づいて分散管理しつつ，ルールに従ってコンテキスト導出や有効なコンテンツの配信を行う手法を提案する．また，提案手法の有効性の検証のために行ったシミュレーション評価の結果を示す．

第 5 章では，本論文の成果を要約したのち，今後の研究課題について述べ，本論文のまとめとする．

目 次

1 序章 **1**

1.1 研究の背景 1

1.2 大規模コンテキストアウェアサービス 4

1.3 大規模コンテキストアウェアサービスに要求される基盤技術 7

1.4 本論文の概要 8

1.5 本論文の構成 9

2 地理的オーバレイネットワークを用いたコンテンツ分散管理手法 **11**

2.1 まえがき 11

2.2 地理的な情報を扱う既存 P2P 技術 12

2.2.1 地理的オーバレイネットワーク構築手法 12

2.2.2 P2P 型地理的コンテンツ管理手法 14

2.3 地理的コンテンツ分散管理手法 16

2.3.1 コンテンツ管理の階層化 16

2.3.2 地理的コンテンツの登録 17

2.4 探索価値に基づく地理的コンテンツ検索手法 18

2.4.1 4 エリア探索手法 18

2.4.2 反復探索手法 21

2.5 評価 29

2.5.1 反復探索手法の設定値による比較結果 29

2.5.2 探索手法による比較結果 33

2.6 むすび 37

3 センサデータの意味的解釈に基づく分散センサ情報管理アーキテクチャ 39
3.1 まえがき 39
3.2 分散センサ情報管理アーキテクチャの提案 40
3.2.1 アプローチ 40
3.2.2 センサ情報管理の階層化 41
3.2.3 処理結果の共有 45
3.3 実装および評価 50
3.3.1 実装 50
3.3.2 仮想センサの効果の評価 52
3.4 むすび 56

4 Rete アルゴリズムを用いた地理的な情報に基づくコンテンツ配信手法 57
4.1 まえがき 57
4.2 ルール処理のための既存手法および分散環境に適用した関連研究 58
4.2.1 Rete アルゴリズム 58
4.2.2 分散環境を想定した既存手法 61
4.3 ルールおよび地理情報に基づく地理的コンテンツ配信手法 62
4.3.1 想定環境 62
4.3.2 地理的な情報に基づく分散処理 63
4.4 評価 68
4.4.1 評価環境 68
4.4.2 処理結果の共有における評価 70
4.4.3 地理的な情報に基づく分散処理における評価 72
4.4.4 考察 77
4.5 むすび 77

5 結論 79
5.1 本論文のまとめ 79
5.2 今後の研究課題 80

謝辞 **83**

第1章

序章

1.1 研究の背景

近年，PC などの固定端末に加え，携帯電話やスマートフォンなど，インターネットに接続可能なモバイル端末の利用が普及してきた．これにより，ユーザはいつでもどこでもインターネットから情報を獲得したり，インターネット上に情報を発信することが可能になってきた．また，複数のセンサがネットワークで繋がり，それらが互いに協調しつつセンシングを行うセンサネットワーク技術の発展により，さまざまなセンサによる観測値（センサデータ）がインターネット上で活用できる環境が実現されつつある．このように，インターネットに多様な機器が接続されることにより，ユーザがいつでもどこでも多様なサービスを享受できるユビキタス環境が発展してきた [55].

ユビキタス環境におけるサービスでは，ユーザ自身が提供する情報に加え，センサデータも活用することが可能である．特に，センサデータから実世界の状況を把握し，その状況に応じたサービスを提供するコンテキストアウェアサービスの実現が期待されている．

文献 [5] では，コンテキストアウェアサービスで利用されるコンテキストを，以下の 4 種類に分類している．

- コンピューティングコンテキスト（Computing Context）

  ネットワーク接続性，通信コスト，通信帯域，付近のプリンタ，ディスプレイなどのリソース情報など

- ユーザコンテキスト（User Context）

ユーザのプロフィール（ユーザの属性情報や嗜好情報など），位置情報，周囲に存在するユーザの情報，社会的地位など

- 物理コンテキスト（Physical Context）

  明るさ，騒音レベル，交通情報，気温など，実世界に設置されたセンサから得られる情報

- 時間コンテキスト（Time Context）

  日，時刻，曜日，季節など

以降，本論文ではこれらを総じて「コンテキスト」と呼ぶ．

従来の情報サービスでは，ユーザが能動的に情報を取得する形態（プル型と呼ばれる），すなわち，必要な情報の所在を指定して取得したり，検索キーワードを入力して所望の情報を探しだしたりする形態が基本である．一方，コンテキストアウェアサービスでは，能動的な情報取得形態のみならず，ユーザが能動的に入力などを行わなくともコンテキストに応じて情報を得るといった，より受動的な情報取得形態（プッシュ型と呼ばれる）をも可能とする．これによって，能動的な情報取得形態ではユーザが気付くことができなかった重要な情報や，潜在的に必要としていた情報が得られるようになる．

コンテキストアウェアサービスの例としては，例えば以下の情報サービスが挙げられる．

- コンテンツ推薦サービス

  ユーザの現在地周辺の飲食店や名所を提供するなど，ユーザやコンテンツのコンテキストに基づいて，ユーザに適すると判断されたコンテンツを推薦する．コンテキストを把握することで,「オープンカフェに関する情報は晴れの日のみユーザに提示する」など，ユーザにとってリアルタイム性や利便性の高いサービスを提供できる．コンテキストとしては，ユーザの現在地やプロフィール，過去の行動履歴などのユーザコンテキスト，コンテンツが関連する位置の物理コンテキストなどが用いられる．例えば米田ら [41] は，観光地内で RFID によりユーザの位置を検知し，サーバからインターネット経由で携帯電話へ情報をプッシュ型で発信するシステムを提案している．これらの位置に基づくコンテンツ推薦では，ユーザや推薦対象であるコンテンツは位置情報をもち,「コンテンツの位置がユーザの半径 1km 以内ならば提示する」など，ユーザの位置とコンテンツの位置を照合することでそのコンテンツを推薦するか

どうかを判断する．位置情報の取得には GPS（Global Positioning System）のほか，住所やカメラ映像，無線基地局などを用いて測位する位置推定手法も提案されている [15, 54, 61] .

- 経路ナビゲーションサービス

  ユーザの現在地および目的地に基づいて，目的地までの経路のナビゲーションを行う．ユーザや目的地の位置だけでなく，徒歩か自動車かなどのユーザの移動媒体や道路状況に関する情報も用いることで，より利便性の高いサービスを提供できる．ユーザコンテキストとしての位置の測定には前述と同様の測位技術が用いられる．道路状況はカメラ映像の解析や車両の速度センサなどの情報をもとに,「速度が 20km 以下で走行していればその道路は混雑している」といった判断がなされる．また，ウェアラブルコンピューティングによるナビゲーションシステムとして，ヘッドマウントディスプレイを用いた研究も行われている [26, 44, 60].

- 健康支援サービス

  和泉ら [24] の提案する健康支援システムでは，自宅で生活している患者に脈拍や血圧などを測定するセンサを装着し，健康状態を長期的に計測することで，健康の向上に繋がる支援を患者自身に行う.「患者が高血圧と判断されれば，具体的な食生活の改善案を提示する」などの行動支援がなされる．また，患者の異常を検知した場合に担当医へ連絡するなど，患者の健康状態に応じた他者への通知も行える.

上記のように，ユビキタス環境においてさまざまな局面で有効な情報提供が可能となるコンテキストアウェアサービスの実現への期待は高い．また，マイクロブログの普及などにより膨大な情報がインターネット上に提供され，必要な情報を選別して得ることが困難となってきており，そうした問題に対処可能なコンテキストアウェアサービスの実現に対する要求も高まっている．こうした動向を踏まえ，数多くの研究開発が行われている.

コンテキストアウェアサービス構築のための要素技術としては，これまで，コンテキストの記述方法，コンテキスト抽出に必要なセンサデータの収集方法，収集されたセンサデータの解釈方法，解釈されたコンテキストに基づくアプリケーション開発方法などが数多く検討されている．以下にそれぞれについてまとめる.

- コンテキストの記述方法

前述の和泉ら [24] は，セマンティック・ウェブ技術のオントロジで情報を記述している．多くのコンテキストアウェアサービスではコンテキストに基づいた処理が IF-THEN 形式のルールで行われ，文献 [24] では，ルールの記述に SWRL（Semantic Web Rule Language）が用いられている．同様に，文献 [6,46] でも，オントロジによりコンテキストを記述している．

- センサデータの収集方法

  服部ら [7,16,17,64] は，ユーザが持つ携帯電話などに軽量なエージェントを配置し，ユーザに関するデータの収集や情報の通知を分散的に行い，統合的に管理するサーバと連携するモデルを提案している．

- センサデータの解釈方法

  センサデータを用いたコンテキストアウェアサービスを実現するためのコンテキスト導出手法として，Ni ら [47] はセマンティックセンサネットワークを実現するためのフレームワークを提案している．また広田ら [18,21] は，物体の情報や物体同士の関係，物体と環境との関係を論理表現に基づく推論規則によって推定する，セマンティックセンサネットワークを提案している．センサデータの推論によってコンテキストを解釈する研究 [14] も行われている．

- アプリケーション開発支援

  藤波ら [12] はコンテキストをベースコンテキストとメタコンテキストに分離するアーキテクチャを提案し，メタコンテキストをフレームワーク内で扱うことで，センサやコンテキスト抽出アルゴリズムを意識せずにアプリケーションを開発できる仕組みを提案している．また，Dey ら [9] は，アプリケーション開発者が視覚的にルールを生成できるプログラミング環境として，iCap を提案している．

1.2 大規模コンテキストアウェアサービス

コンテキストアウェアサービスの実現にあたっては，特定の建物や敷地の内部を想定するのではなく，ユビキタス環境であらゆる場所に広域かつ膨大に発生するセンサデータやコンテンツを扱うことで，さまざまな場所や状況のユーザの要求を満たすことができる．本

研究では，この広域かつ膨大なデータを扱うコンテキストアウェアサービスを「大規模コンテキストアウェアサービス」と呼ぶ．

大規模コンテキストアウェアサービスにおいては，膨大な数の端末や機器が任意のタイミングでコンテンツやセンサデータを発生させる．従来のクライアント・サーバアーキテクチャでこのような膨大な数のコンテンツやセンサデータを管理し提供するサービスを構築した場合，特定のサーバやある組織内のサーバクラスタですべての情報を管理し提供することになり，サーバ機器や周辺ネットワークに負荷が集中してしまう．すなわち，クライアント・サーバアーキテクチャでは，コンテンツやセンサーデータの数の増大へ対応するにはサーバ機器や周辺ネットワークの性能を増強するしかなく，規模の拡大に対応することが困難である．また，さまざまな組織が開発する多様なサービスからコンテンツを共有して活用できることが望ましいが，単一の組織が大規模コンテキストアウェアサービスを構築する形態ではそのような有効なコンテンツ共有が困難である．そのため，扱うコンテンツやセンサデータ，ユーザの規模に制限があり，さまざまなデータやユーザが存在する大規模コンテキストアウェアサービスへは適用できない．

このようなクライアント・サーバアーキテクチャにおける問題を解決するために，個々のコンピュータや機器が相互に協調して情報共有や分散処理を実現する，分散型アーキテクチャに基づくシステム構成方法が提案されている．分散型アーキテクチャのうち特に，集中サーバが存在せず，コンピュータ間が自律的に動作することによってシステムを構成する P2P（Peer-to-Peer）アーキテクチャは，耐障害性や拡張性の高さから有望なシステム構成手法であり，多くの研究開発がなされている．P2P アーキテクチャでは，情報共有機能や分散処理を担うコンピュータや機器をピアと呼び，ピア間に論理的なネットワークであるオーバレイネットワークを構成する．代表的な既存手法として，CAN [52] や Chord [59], Pastry [53] などの DHT（Distributed Hash Table）がある．情報の登録や検索，処理はこのオーバレイネットワークに接続したピア間で分担するため，特定のピアへの負荷の集中を避けることが可能である．

1.1 節で挙げた既存のコンテキストアウェアサービス実現手法は，いずれもクライアント・サーバアーキテクチャを基本としており，そのまま P2P アーキテクチャに適用することができない．本研究では，P2P アーキテクチャに基づき，規模拡張性を確保可能な大規模コンテキストアウェアサービスを構築するための基盤技術の確立を目指す．

大規模コンテキストアウェアサービスでは，位置情報は重要かつ最も基本的なコンテキ

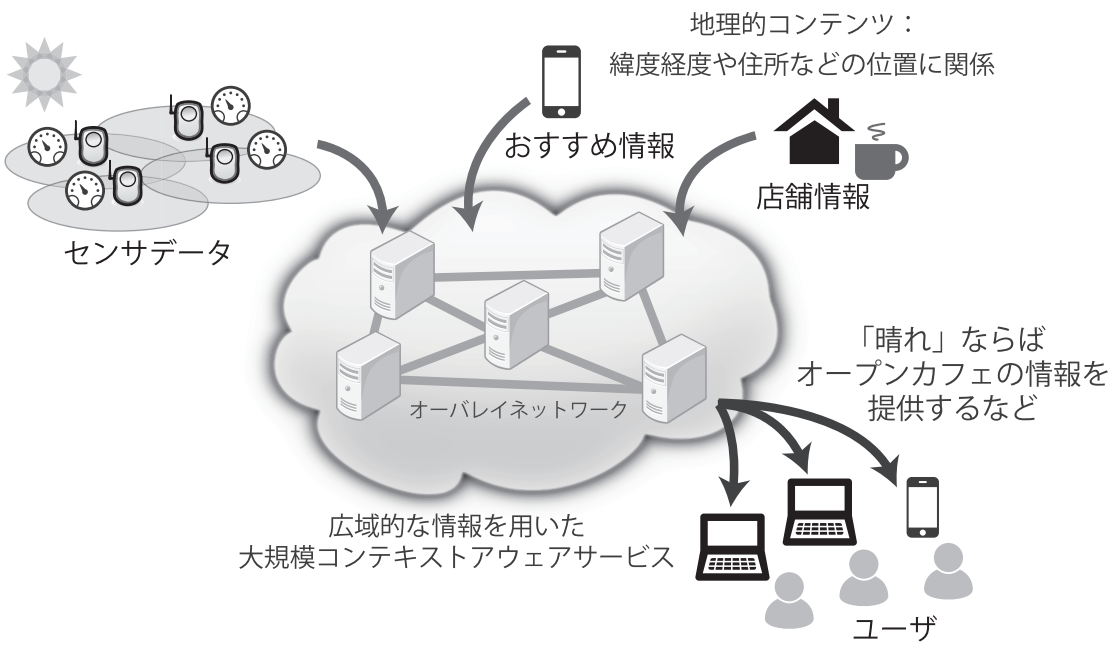


図 1.1: 大規模コンテキストアウェアサービス

ストとなる．そのため，現実世界におけるユーザ自身の位置や，観測値が有効となる地理的な範囲を持ったセンサデータ，ある店舗に関するコンテンツなど，地理的な位置や範囲にかかわるコンテンツの扱いが重要となる．これらのコンテンツを「地理的コンテンツ」と呼ぶ．本研究では，地理的コンテンツのユーザへの提供を中心とした大規模コンテキストアウェアサービスの実現法を検討する．

想定する大規模コンテキストアウェアサービスとしての地理的コンテンツ配信の例を図 1.1 に示す．想定サービスでは，地理的コンテンツやセンサデータの分散管理，コンテキスト導出，コンテンツ配信などの処理を分担するコンピュータ群がオーバレイネットワークを構築する．地理的コンテンツやセンサデータは，オーバレイネットワークを構築しているピアへインターネット経由で送信され，分散管理される．また，ユーザは自身の現在地や地図アプリケーション上での操作によって地理的な範囲などの条件を指定しておき，条件を満たす地理的コンテンツの配信を受ける．地理的コンテンツの配信を行うピアでは，分散管理されているセンサデータからユーザやコンテンツのコンテキストを導出し，導出されたコンテキストを照合することで，地理的コンテンツを関連するユーザへ配信する．地理的コンテンツのユーザへの配信は，センサデータの更新によるコンテキストの変化にともなって逐次行われる．

1.3 大規模コンテキストアウェアサービスに要求される基盤技術

1.2 節で述べた大規模コンテキストアウェアサービスを構築するためには，以下の 3 つの基盤技術が求められる．

- 地理的コンテンツやセンサデータの検索および収集

  大規模コンテキストアウェアサービスでは，ユーザの現在地など，位置に基づき，分散したピアに保持される地理的コンテンツの検索機能を実現できる必要がある．また，ある店舗のコンテキストはその位置周辺のセンサデータから導出するなど，地理的コンテンツに基づくコンテキストを導出する上では，地理的に対応するセンサデータを得る必要がある．センサは設置場所や観測値の有効範囲などの地理的な情報をもち，分散管理されているセンサデータを検索および収集できる必要がある．このとき，膨大な数のユーザや地理的コンテンツ，センサデータが存在する環境においても，地理的コンテンツやセンサデータの検索および収集を現実的な時間内に完了できる効率的な基盤技術が要求される．

- センサデータからのコンテキスト導出

  大規模コンテキストアウェアサービスでは，広域にわたるあらゆる場所でのコンテキスト導出が求められる．しかし，想定環境では単一の組織が計画的にセンサを設置しているわけではないため，必要な場所にセンサが存在せず，センサデータの取得およびコンテキストの導出ができない状況が発生する．そのため，こうした，センサデータの取得ができない状況でコンテキストアウェアサービスを実現するための対処が必要となる．

- コンテキストに基づくコンテンツ配信

  大規模コンテキストアウェアサービスでは，大量のユーザが存在する状況でも，それぞれのユーザのコンテキストに適合するコンテンツを現実的な時間内に処理し，配信する必要がある．すなわち，膨大な数のユーザやコンテンツのコンテキストを互いに照合したうえで，条件を満たすコンテンツを適切なユーザへ効率的に配信する基盤技術が要求される．

従来のコンテキストアウェアサービスを実現する要素技術では，これらの要求が十分に扱われておらず，いまだ基盤技術として確立していない．本論文では，大規模コンテキストアウェアサービス構築基盤の確立を目的として，3 つの基盤技術に対して以降の章でそれぞれ新たな手法を提案している．

1.4 本論文の概要

本論文の概要は以下のとおりである．

まず，地理的コンテンツやセンサデータの検索および収集を効率的に行うための基盤技術として，地理的オーバレイネットワークを用いたコンテンツ分散管理手法を提案する．提案手法は，既存のさまざまな地理的オーバレイネットワークに適用可能とするため，地理的コンテンツの管理を地理的オーバレイネットワーク層と地理的コンテンツ管理層の 2 層に分離して実現するとともに，地理的コンテンツをその位置の最近傍ピアに管理させることにより地理的局所性を実現する．また，ピアや地理的コンテンツの情報を収集しながら反復的に探索することで，効率的な地理的コンテンツ検索を実現する．

次に，センサデータの取得ができない状況に対処しつつ，センサデータからコンテキストを導出し，コンテキストアウェアサービスを実現可能とする基盤技術として，センサデータの意味的解釈に基づく分散センサ情報管理アーキテクチャを提案する．提案アーキテクチャは，センサデータを扱う層である Raw Layer，センサデータの補間処理を行う Abstract Layer，ルールに基づきコンテキスト導出処理を行う Logical Layer からなり，これらの 3 層の連携によってセンサデータに付与されたメタデータおよびコンテキスト導出のためのルールに基づき任意の地点のコンテキスト導出を行う．また，Abstract Layer における補間処理の結果や Logical Layer におけるコンテキスト導出結果の再利用を可能にする仮想センサを導入することにより，コンテキスト導出処理の負荷低減を実現する．

さらに，大量のユーザがサービスに参加している状況のもと，コンテキストに基づく地理的コンテンツを効率的に配信可能とするための基盤技術として，Rete アルゴリズムを用いた地理的な情報に基づくコンテンツ配信手法を提案する．提案手法は，ルール処理を効率化する Rete アルゴリズムを分散環境で実現するとともに，コンテンツやセンサデータを地理的な情報に基づいてブロックに分割して分散管理することにより，ユーザに対する効率的な地理的コンテンツ配信を実現する．

1.5 本論文の構成

本論文は 5 章で構成され，それぞれの内容は次のとおりである．

まず，第 2 章では，地理的オーバレイネットワークを用いたコンテンツ分散管理手法を提案する．また，提案する地理的コンテンツ分散管理手法のシミュレーション評価結果を示し，その有効性について検証する．第 3 章では，センサデータの意味的解釈に基づく分散センサ情報管理アーキテクチャを提案する．また，提案する分散センサ情報管理アーキテクチャのシミュレーション評価結果を示し，その有効性について検証する．第 4 章では，Rete アルゴリズムを用いた地理的な情報に基づくコンテンツ配信手法を提案する．また，提案するコンテンツ配信手法のシミュレーション評価結果を示し，その有効性について検証する．第 5 章では，本論文の成果を要約したのち，今後の研究課題について述べ，本論文のまとめとする．

なお，第 2 章は文献 [31, 32, 33, 34]，第 3 章は文献 [29, 30, 36]，第 4 章は文献 [35] で公表した結果に基づき論述する．

第2章

地理的オーバレイネットワークを用いたコンテンツ分散管理手法

2.1 まえがき

広域にわたる大規模コンテキストアウェアサービスでは，作成者の現在地や店舗の住所など，対象とする多くの情報は何らかの位置に関連する．また，センサは実世界に設置され，センサデータも設置場所や観測値の有効範囲などの地理的な情報をもつ．以降，地理的な情報に関連するこのようなコンテンツを地理的コンテンツと呼ぶ．

携帯電話などの GPS 機能を搭載したモバイル端末の普及により，地理的コンテンツを扱う位置情報サービス（Location-Based Service; LBS）や地理情報システム（Geographic Information System; GIS）について，さまざまな研究開発が行われている．例えば，久永ら [19] は，歩行者 ITS（Intelligent Transport Systems）において，携帯端末向け 3 次元地図表示方式を提案している．また，Bao ら [3] は，指定された地理的範囲に基づいて友人やニュースソースから発信された情報をユーザへ配信する GeoFeed システムを提案している．さらに，経度緯度情報であるジオタグが付けられた写真を地図上に表示し，観光マップとして活用するような研究も行われている [4,63]．このように，地理的コンテンツは人々の実生活を支援する情報として重要な役割を果たすため，その活用についてこれまで盛んに研究開発が進められている．

ユビキタス環境においては，膨大な数の端末や機器が存在し，それらが自律的に地理的コンテンツを発信することになる．地理的コンテンツは，コンテンツが関連する位置から近

い場所から多く発信・取得され，発信者と利用者がネットワーク的に近い接続をできる場合も多いと考えられるが，クライアント・サーバアーキテクチャでは，地理的コンテンツの位置やそれを必要とする利用者の位置に関係なく，必ず特定のサーバにアクセスしてコンテンツを登録したり取得したりしなければならない．また，地理的コンテンツの有効活用を考えた場合，さまざまな組織が開発する多様なサービスから地理的コンテンツを共有して活用できることが望ましいが，単一の組織が地理的コンテンツを管理する形態ではそのような有効なコンテンツ共有が困難である．このようなクライアント・サーバアーキテクチャにおける問題を解決するため，1.2 節で述べたように P2P アーキテクチャを基盤とした地理的コンテンツの分散管理手法について，これまで多くの研究が行われてきた [13,42,56,57].

しかし，既存の P2P 型地理的コンテンツ管理手法では，地理的コンテンツ管理専用のオーバレイネットワーク構造を構成する必要がある．また，地理的コンテンツの担当ピアが実際のピアが存在する位置とは無関係に決まるため，近隣の位置依存コンテンツであっても，地理的に遠くにあるピアが担当する場合があるなど，検索のために発生するトラヒックなどの面でも問題がある．

そこで本章では，ピアが自身の位置情報に基づいて構成する地理的オーバレイネットワーク層と，地理的コンテンツを管理する地理的コンテンツ管理層を分離し，任意の地理的オーバレイネットワーク上で構成可能な地理的コンテンツ分散管理手法を提案する．提案手法では，地理的コンテンツをその位置の最近傍ピアに管理させることにより地理的局所性を実現し，効率的な地理的コンテンツ検索を実現する．

以下，2.2 節では地理的な情報を扱う既存 P2P 技術について述べる．2.3 節では提案する地理的コンテンツ分散管理手法について述べる．2.4 節では 2.3 節の手法で分散管理されている地理的コンテンツに対する検索手法について論じ，2.5 節でシミュレーションによる評価について述べる．最後に 2.6 節で本章をまとめる．

2.2 地理的な情報を扱う既存 P2P 技術

2.2.1 地理的オーバレイネットワーク構築手法

位置情報に関連する既存 P2P 技術として，ピアの位置情報に基づく地理的オーバレイネットワーク構築手法が提案されている．

LL-Net [25] は，領域全体を格子状に階層的に分割して考え，各階層レベルにおいて隣接

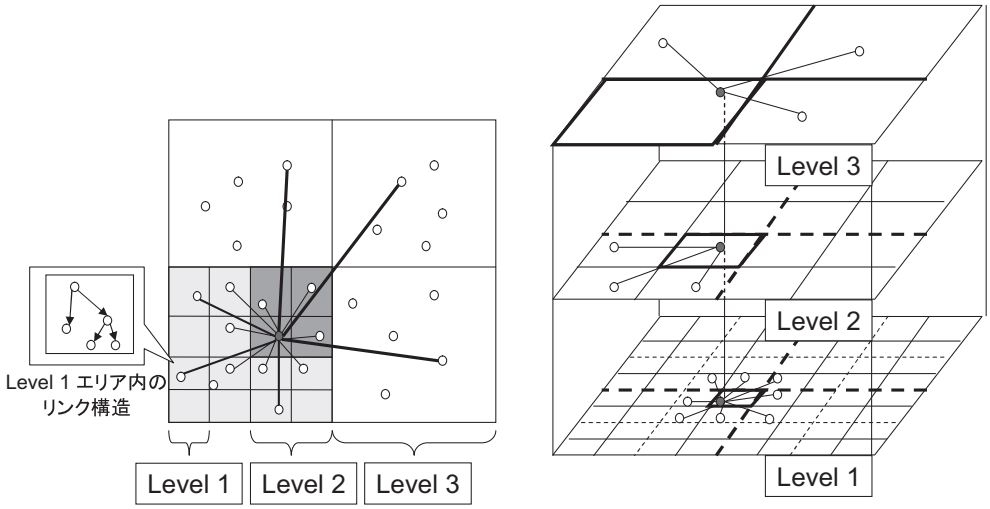


図 2.1: LL-Net

する領域にリンクを構成することにより遠隔の領域にも効率的に検索要求を転送できるようにした地理的オーバレイネットワークである．具体的には，図 2.1 (a) のように，領域全体を格子状に分割し，最小エリアと定義する．各ピアはまず，同じ最小エリア内に存在するピア間で木構造のネットワークを構築する．また，各ピアは隣接する最小エリア内のピアへも 1 本ずつ，最大で 8 本のリンクを構築する．図 2.1 (b) のように，各エリアは 4 つのエリアごとに階層的なレベルとして扱われ，レベル N では次のレベル $N+1$ を形成する 3 つのエリア内のピアへのリンクを構築する．このように階層的に構築されたリンクを用いることで，任意のエリアへのホップ数を $O(\log_4 n)$（n は最小エリアの数）に抑えている．

文献 [1,20,49] では，ボロノイ領域（Voronoi Regions）に基づくドロネーオーバレイネットワーク構築手法が提案されている．ボロノイ領域とは，図 2.2 (a) のような 2 次元平面上の複数の点（母点）に対して，各点とその周囲の点から得られる垂直二等分線を境界とする領域である．図 2.2 (b) のように，ボロノイ領域が隣接する母点を結んだ辺はドロネー辺と呼ばれ，ドロネー辺で構成される三角形はドロネー三角形と呼ばれる．ドロネーオーバレイネットワーク構築手法では，各ピアの位置をボロノイ領域の母点とし，ドロネー辺で示されるリンクを構築することで，地理的に近いピアとのリンクを構築する．また，文献 [40,50,62] では階層的にドロネーオーバレイネットワークを構築する手法を提案しており，ピアを探索する際のホップ数を抑えている．

地理的オーバレイネットワークでは，指定した地理的な範囲に含まれるピアへクエリを

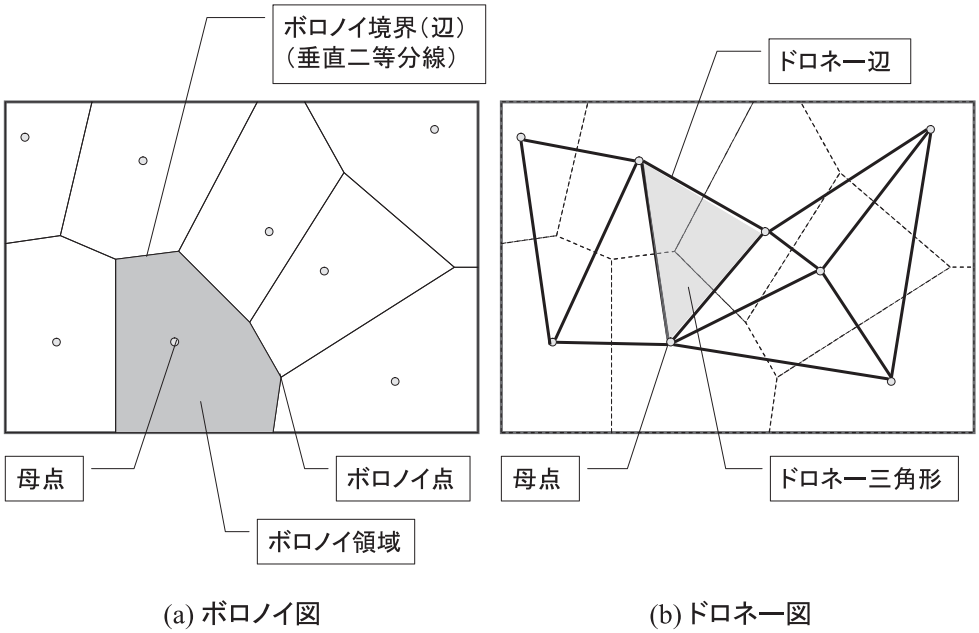


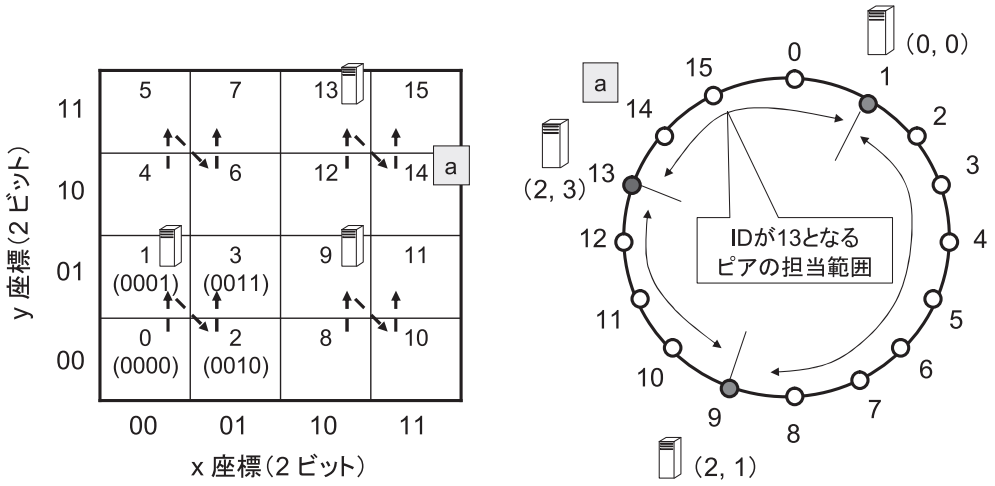
図 2.2: ボロノイ領域およびドロネー図

転送することで，ピアの範囲探索（Range Search）が可能である．また，指定した地点を基準に範囲探索を行うことで，ピアの最近傍探索（Nearest Neighbor Search）も可能である．しかし，地理的オーバレイネットワーク構築手法は，ピアを探索するための手法であるため，ピアが自身の位置と異なる位置に関する地理的コンテンツを保持している場合，地理的コンテンツの位置に基づく検索には対応できない．地理的オーバレイネットワークを用いて地理的コンテンツを分散管理するための方法として，地理的コンテンツをそれぞれ 1 つのピアとして仮想的に地理的オーバレイネットワークを構築することが考えられる．しかし，コンテンツの数はピアの数と比べて膨大であることが想定されるため，オーバレイネットワークが肥大化する．また，1 つのコンピュータが離脱すると複数ピアが同時にオーバレイネットワークから消失するため，オーバレイネットワークの維持のために大きなコストが必要となる．

2.2.2 P2P 型地理的コンテンツ管理手法

既存の P2P 型地理的コンテンツ管理手法として，Mill [42] や ZNet [57] などが提案されている．

Mill [42] では，2 次元座標を 1 次元の値に変換することで，地理的コンテンツの登録や



(a) Z-ordering による2次元情報の1次元化　　(b) ピアの担当範囲

図 2.3: Mill

検索の手順を単純化している．(x, y) で表される 2 次元座標を 1 次元の値に変換するには，図 2.3 (a) に示すように，空間充填曲線の Z-ordering [51] を用いている．Mill では，各ピアの位置を Z-ordering により 1 次元の値に変換し，その値を Key としてコンピュータ間に DHT（Distributed Hash Table）の Chord [59] に類似したオーバレイネットワークを構築する．各ピアは担当する Key の範囲をもち，コンテンツの管理や検索は，図 2.3 (b) のように，地理的コンテンツの位置が変換された Key に基づいて行われる．例えば，位置 $(3, 2)$ に関する地理的コンテンツ a を登録する場合を考えると，2 ビット表記された座標 $(11, 10)$ は 1 次元の Key では 14 となる．したがって，Key が 14 以前で最も近い，Key が 13 のコンピュータがその地理的コンテンツを管理する．

ZNet [57] も Mill と同様，2 次元座標を 1 次元の値に変換することで地理的コンテンツの登録や検索の手順を単純化している．ZNet では 2 次元座標の 1 次元の値への変換には同様に Z-ordering を用いており，1 次元の値を Key として構築するオーバレイネットワークには Skip Graph [2] を用いている．Skip Graph は Chord と同様に Key の順に並べて円形のリンクを構築し，各ピアが確率的な階層構造のリンクも構築する．コンテンツの管理や検索も Mill と同様，地理的コンテンツの位置が変換された Key に基づいて行われる．

地理的コンテンツの分散管理を目的としたこれらの既存手法はいずれも，2 次元座標を 1 次元の Key に変換し，各ピアにコンテンツの担当範囲を決めつつ，地理的コンテンツを分散管理する．そのため，2 次元座標を 1 次元の Key に変換したうえで担当ピアを探索

する必要があり，それぞれ独自のオーバレイネットワークを構築する必要がある．また，2次元座標を1次元の Key に変換する場合，地理的な近接性が失われる．そのため，近隣の地理的コンテンツであっても，地理的に遠隔のピアが担当する場合がある．地理的コンテンツの検索においても，指定する地理的な領域は断片化された複数の ID の範囲に変換され，それぞれがクエリとして地理的に拡散されることになるが，広域にわたる情報を扱う環境においては，クエリがさまざまな地域に拡散されることは望ましくない．

2.3 地理的コンテンツ分散管理手法

2.3.1 コンテンツ管理の階層化

2.2 節で述べたように，ピアの位置に基づく地理的オーバレイネットワークはピアの位置に基づくピア探索には対応するが，そのままでは地理的コンテンツの検索には対応できない．また，地理的コンテンツの管理を目的とする既存 P2P 型地理的コンテンツ管理手法では，特定のオーバレイネットワークの構築が必要となる問題がある．

これらの問題を解決するため，ピアの位置に基づいて構成される任意の地理的オーバレイネットワークを基盤とし，その上位層において地理的コンテンツ管理を行う手法を提案する．つまり，下位の地理的オーバレイネットワークがもつピア探索機能を基盤として，地理的コンテンツの格納と検索を行う地理的コンテンツ管理層をその上位層で実現する．このような構成とすることにより，上位の地理的コンテンツ管理層は，特定の地理的オーバレイネットワークに依存せず，2.2.1 節で述べた手法以外にもさまざまな地理的オーバレイネットワーク構築手法へ適用可能となる．

提案する地理的コンテンツ分散管理手法の概略を図 2.4 に示す．地理的オーバレイネットワーク層については，既存技術や地図アプリーケションとの連携などを考慮し，多くの地理的オーバレイネットワークが対応する，矩形による探索範囲の指定が可能なものを想定する．また，地理的コンテンツの保持と管理は，地理的オーバレイネットワーク層の機能で実現される最近傍探索を利用し，その位置情報に最も近い位置に存在するピアに担当させることとする．担当ピアは地理的コンテンツの位置とその周辺ピアの位置の情報のみで決定でき，既存の地理的コンテンツ管理手法のように専用のオーバレイネットワークを必要としない．そのため，矩形による範囲探索が可能な地理的オーバレイネットワークならば，提案する地理的コンテンツ管理層を適用できる．

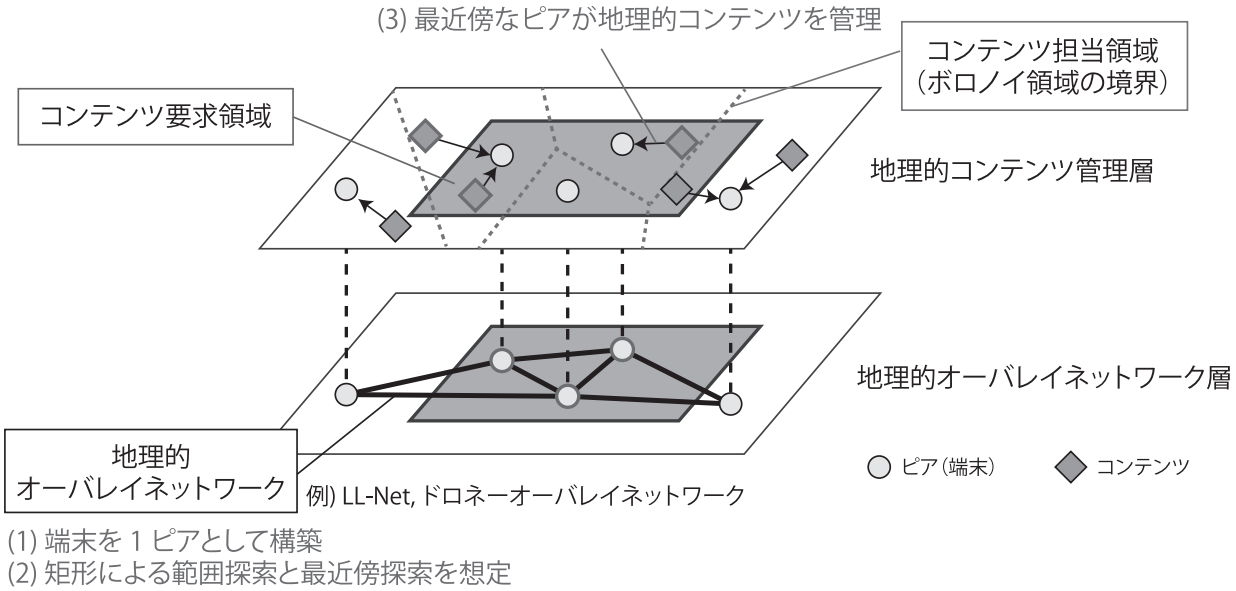


図 2.4: 提案する地理的コンテンツ分散管理手法の概略

2.3.2 地理的コンテンツの登録

地理的コンテンツの登録は，前節で述べたように，登録する地理的コンテンツの最近傍ピアに登録する．最近傍ピアの発見は，地理的オーバレイネットワーク層において最近傍ピアの探索機能が提供されていればそれを用いればよいが，最近傍ピア探索機能が提供されていない場合は，範囲指定によるピア探索機能を用いて以下で述べる方法により実現する．

まず，位置 (x, y) に関する地理的コンテンツを登録する場合，図 2.5 (a) のように，位置 (x, y) から設定された水平幅 w，垂直幅 w の範囲のピアを地理的オーバレイネットワーク層において探索する．ここで，範囲内に該当するピアが存在しなかった場合，探索範囲をあらかじめ設定した分だけ拡大し，再探索する．この周囲の探索は，探索範囲にピアが 1 つ以上発見されるまで行う．

次に，発見されたピアの中から位置 (x, y) の最近傍ピアを仮決定し，その直線距離 d を求める．このとき，図 2.5 (a) のように，最近傍として仮決定したピア（図のピア 1）との距離 d が $\frac{w}{2} < d$ の場合，さらに近いピアが探索範囲外に存在する可能性がある．その場合，図 2.5 (b) のように，位置 (x, y) を中心に，水平幅と垂直幅が $2d$ の範囲に存在するピアを，地理的オーバレイネットワーク層で再探索する．再探索の結果，最近傍として仮決定したピアよりも (x, y) に近い位置に存在するピアが発見された場合，そのピアを最近傍ピアとし，そのようなピアが発見されなかった場合は仮決定したピアを最近傍ピアとする．

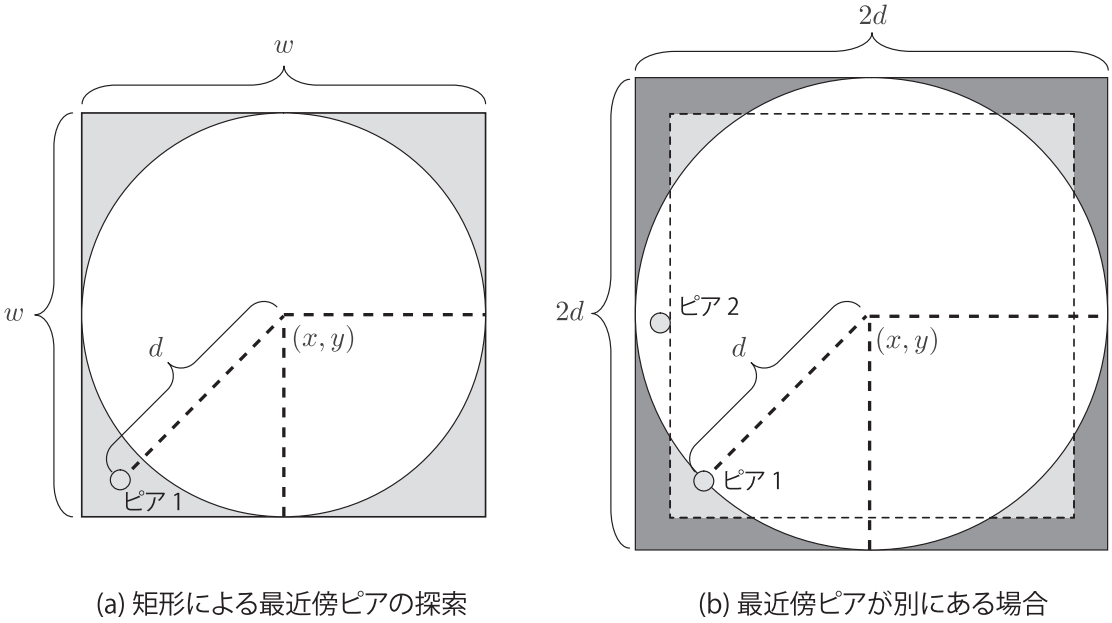


図 2.5: 範囲探索による最近傍ピア探索

例えば，図 2.5 (b) はピア 1 よりも (x, y) に近いピア 2 が発見された例であり，この場合はピア 2 が最近傍ピアとなる．

このようにして最近傍ピアが決定されれば，そのピアに対してコンテンツ登録要求を送信する．

2.4 探索価値に基づく地理的コンテンツ検索手法

2.4.1 4 エリア探索手法

地理的コンテンツをその位置情報に最近傍のピアに登録した場合，図 2.6 (a) のように，各ピアに登録される地理的コンテンツの存在可能範囲（以下，コンテンツ担当領域と呼ぶ）は，各ピアの位置を母点としたボロノイ領域で表される．ボロノイ領域とは，それぞれの点（母点）とその周囲の点で得られた垂直二等分線を境界とする領域である．

ここで，図 2.6 (a) のようなピアの配置において，地理的コンテンツがそれぞれの最近傍ピアに登録されており，この図の中央に示した矩形の領域内の地理的コンテンツを検索したい場合を考える．以降，検索したい地理的コンテンツの領域をコンテンツ要求領域と呼ぶ．このコンテンツ要求領域を D_0 とした場合，D_0 内に存在するピアのみの探索では，D_0

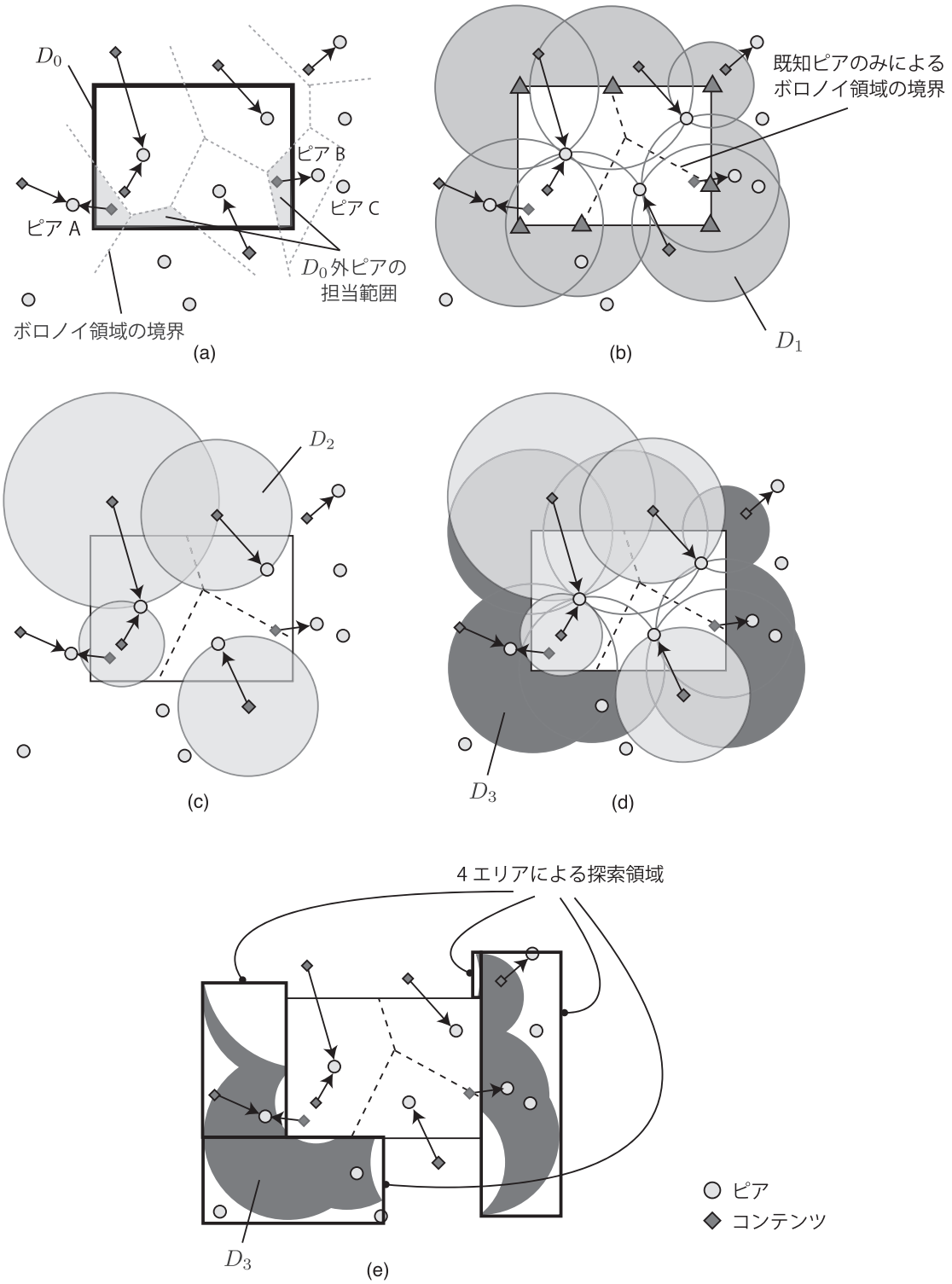


図 2.6: 提案手法におけるピア探索範囲

内の地理的コンテンツを D_0 の外のピアが管理している可能性があるため，対象コンテンツの取得漏れが発生する可能性がある．図 2.6 (a) の例では，D_0 外の左右に存在するピア A, B がそれぞれ対象コンテンツを管理しているが，D_0 内に存在するピアのみを探索した場合はピア A, B を発見できず，結果的に対象コンテンツを取得できない．この対象コンテンツの取得漏れは，コンテンツ要求領域とピアによるコンテンツ担当領域のずれと，探索済み範囲のピアによる不完全なボロノイ領域しか把握できないことで生じる．図 2.6 (a) の例では，コンテンツ要求領域内の正確なボロノイ領域は 6 つのピアで構成されるが，コンテンツ要求領域内のみの探索では，そのうちの 3 つのピアからの不完全なボロノイ領域しか得られない．そのため，対象コンテンツをすべて取得するには，対象コンテンツの担当ピアが存在しうる領域（以降，担当ピア存在可能領域と呼ぶ）も探索する必要がある．

まず，D_0 内に存在するピアの位置情報だけを考えた場合，担当ピア存在可能領域は，D_0 のピアのボロノイ担当領域の境界線と D_0 の交点，さらに D_0 の 4 頂点をそれぞれ中心にし，最近傍ピアまでの距離を半径とした正円の集合で表わされる．図 2.6 (b) の例では，正円の中心は三角で示した地点で，担当ピアが存在しうる範囲は D_1 で示した領域である．一方，D_0 内に存在するピアが保持している地理的コンテンツの情報も考えた場合，各地理的コンテンツはその位置に最近傍のピアがそのコンテンツを担当するため，探索されたピアがもつコンテンツの位置を中心にし，そのピアまでの距離を半径とした正円の集合には他ピアが存在しないことがわかる．例えば，図 2.6 (c) の D_2 で示した領域に存在するピアは，D_0 内のコンテンツを保持することはない．以上により，担当ピア存在可能領域は図 2.6 (d) の D_3 に示すような領域となり，以下の式で表される．

$$D_3 = D_1 - D_0 - D_2 \tag{2.1}$$

D_3 内に存在するすべてのピアを探索することで，D_0 内のすべての地理的コンテンツを取得できる．ピアの探索を地理的オーバレイネットワークが対応する矩形で行うには，D_3 を包含する矩形領域を探索すればよい．しかし，D_3 を 1 つの矩形で包含する領域では，探索済みの D_0 内に存在するピアへもクエリが転送される．不要なピアへのクエリの転送を抑えるには，図 2.6 (e) に示したように，D_0 を基準に D_3 を上下左右のグループに分類し，4 つの矩形領域でピアを探索する方法が考えられる．この 4 つの矩形領域で探索する方法を **4 エリア探索手法**と呼ぶ．1 つの矩形領域で探索する場合と比べて，4 エリア探索手法では矩形領域が増え，クエリごとのオーバヘッドが増える．しかし，重複する D_0 内の探索や地理的コンテンツの取得を除外できる．

2.4.2 反復探索手法

4 エリア探索手法では，担当ピア存在可能領域 D_3 内に存在しないピアも探索範囲に含まれてしまう可能性がある．また，D_3 内に存在するピアであっても，そのピアと D_0 の間に他のピアが存在する場合，そのピアは実際にはコンテンツ要求領域内のコンテンツを保持し得ない場合がある．例えば，図 2.6 (a) のピア C の場合は，コンテンツ要求領域の間にピア B が存在するため，ピア C はコンテンツ要求領域内に地理的コンテンツの担当領域をもたない．そのため，一般に D_0 からの距離が遠いピアが対象コンテンツを管理する可能性は大きく下がる．つまり，D_3 に存在する全ピアが実際にコンテンツ要求領域内の地理的コンテンツを担当しているとは限らないため，D_3 のすべてを一度で探索すると実際にはコンテンツ要求領域内の地理的コンテンツを保持していないピアも探索してしまうことになり，これは結果的に無駄なトラヒックを発生させることとなる．

そこで，既知の情報から判別される不要な範囲の探索を抑制しつつ，要求される地理的コンテンツを検索する反復探索手法を提案する．反復探索手法では，D_0 内に存在するピアを地理的オーバレイネットワークの機能を用いて探索した後，それらのピアから保持コンテンツの情報をまず取得し，担当ピア存在可能領域 D_3 を算出する．その後，得られた D_3 の領域を細分化し，それぞれ未知な地点における担当ピアの存在確率を既知のピアやコンテンツの配置，分布密度などから計算する．さらに，計算された存在確率とコンテンツ担当領域を用いることで，探索時に取得される対象コンテンツ数の推定値を探索価値として計算し，価値の高い領域から優先的に探索する．その探索によって得られたピアの位置情報や，それらのピアが保持する地理的コンテンツの位置情報を用いて，担当ピア存在可能領域を逐次更新しながら反復的に探索する．これにより，不要な領域の探索を抑制する．

探索価値の計算

反復探索手法では，まず，コンテンツ要求領域 D_0 のピアとその保持コンテンツの情報を取得し，担当ピア存在可能領域 D_3 を算出する．次に，未探索の地点におけるピアの存在確率を近似的に計算するため，図 2.7 (a) のように，D_0 を水平・垂直方向に任意の分割数 m で矩形のセルに分割し，同様のセルを D_3 を包含するように配置する．その後，D_3 における各セルを探索する価値をそれぞれ計算し，それらの価値に基づいて次に探索する領域を決定する．

各セルの探索価値は各セルを探索した際に取得されるコンテンツ数の推定値で与え，既

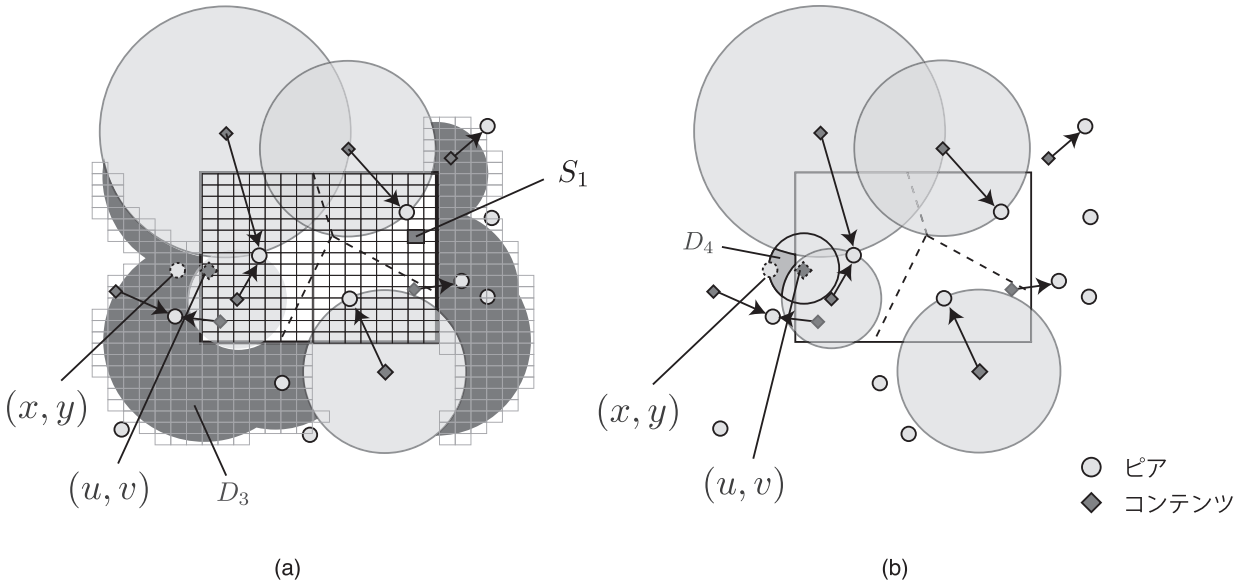


図 2.7: 探索価値の計算

知の情報に基づいたピアやコンテンツの存在密度，対象セルにピアが存在する確率などを用いて計算する．ここで，ピアの存在密度 N_p は，探索済み領域のピア数を探索面積で割った「単位面積あたりの平均ピア数」で求める．また，コンテンツの存在密度 N_c は，探索済み領域のピアが保持するコンテンツの総数を探索面積で割ることで「単位面積あたりのピアが保持する平均コンテンツ数」を計算し，この値を「単位面積あたりの平均コンテンツ数」とする．探索済み領域にあるピアやコンテンツの存在密度は，探索済みの領域周辺の未探索の領域も探索済み領域内と同じであると仮定し，N_p と N_c を近似的に用いる．

未探索な 1 セルにピアが存在しない確率は，時間や距離などの一定区間で事象が発生する数の分布であるポアソン分布にしたがって計算する．まず，一般的なポアソン分布の確率関数は以下の式で表される．

$$f(x) = \frac{\lambda^x e^{-\lambda}}{x!} \tag{2.2}$$

ここで，面積 S_1 の小さな 1 セルにピアが存在する確率を計算する場合，λ は 1 セルにおける平均ピア数となる．前述のとおり，探索済み領域の局所的なピア存在密度 N_p を未探索領域におけるピア存在密度は同じであると仮定しているため，1 セルの面積を S_1 とすると $\lambda = N_p S_1$ で表される．また，セル内にピアが存在しない確率は $x = 0$ のときで，以下の式で表される．

$$f(0) = e^{-N_p S_1} \tag{2.3}$$

したがって，逆に 1 セルに少なくとも 1 ピア存在する確率は $1 - e^{-N_p S_1}$ で表される．

D_3 において中心が (x, y) の特定のセルの探索価値は，そのセル内に存在するピアを探索した場合に得られる地理的コンテンツ数の期待値 $E_c(x, y)$ として，以下の流れで計算する．

1. コンテンツ担当領域に含まれる確率の計算

   図 2.7 (a) のように，コンテンツ要求領域 D_0 における任意のセルを $D_0(u, v)$ $(1 \leq u \leq m, 1 \leq v \leq m)$ で表すものとする．このとき， $D_0(u, v)$ 内に存在する地理的コンテンツを (x, y) のピアが担当するのは，$D_0(u, v)$ を中心にし，(x, y) までの距離を半径とした正円に他ピアが存在しない最近傍となる場合である．したがって，(x, y) のピアが $D_0(u, v)$ をコンテンツ担当領域に含む確率 $P(u, v)$ は，その正円から探索済みである D_0 およびピアが存在しない D_2 を除外した領域 D_4 内に他のピアが存在しない確率と (x, y) にピアが存在する確率の積になる．したがって，領域 D_4 の面積を $S(u, v)$ とすると，$P(u, v)$ は以下の式で表される．

$$P(u, v) = \begin{cases} e^{-N_p S(u,v)}(1 - e^{-N_p S_1}) & (D_4 \text{ 内に他ピアが発見されていない場合}) \\ 0 & (D_4 \text{ 内に他ピアがすでに発見されている場合}) \end{cases} \tag{2.4}$$

2. 取得できる地理的コンテンツ数の期待値の計算

   $D_0(u, v)$ は $P(u, v)$ の確率でコンテンツ担当領域になり，取得できる地理的コンテンツ数の期待値はコンテンツ存在密度 N_c と 1 セルの面積 S_1 の積 $N_c S_1$ で表される．一方，$1 - P(u, v)$ の確率でコンテンツ担当領域外になり，この場合の取得できる地理的コンテンツ数の期待値は 0 になる．したがって，(x, y) のピアの $D_0(u, v)$ における取得できる地理的コンテンツ数の期待値 $E_1(u, v)$ は以下の式で表される．

$$\begin{aligned} E_1(u, v) &= P(u, v)(N_c S_1) + (1 - P(u, v)) \times 0 \\ &= (1 - e^{-N_p S_1}) N_c S_1 e^{-N_p S(u,v)} \end{aligned} \tag{2.5}$$

3. セルの探索価値の計算

   (x, y) を中心とするセルの探索価値 $E_c(x, y)$ は，コンテンツ要求領域 D_0 の任意のセル $D_0(u, v)$ で取得できる地理的コンテンツ数の期待値 $E_1(u, v)$ を用いた以下の式で表される．

```
Require:
  D_0: コンテンツ要求エリアを m × m 個に分割したセルの集合
  peers: 探索されたピアの情報
  contents: 取得されたコンテンツの情報
  searchedArea: 探索済み範囲
  D_3: 探索済み範囲を D_0 と同様の大きさに分割したセルの集合
Ensure:
  各セルの探索価値を算出した D_3
  ---------------------------------------------------------------------------
  N_p ← peers.length/calculateSize(searchedArea); // 既知の情報から，ピアの存在密度を算出
  N_c ← contents.length/calculateSize(searchedArea); // 既知の情報から，コンテンツの存在密
  度を算出
  D_2 ← calculateContentCircles(peers, contents); // 各コンテンツを中心に，最近傍ピアまでの
  距離を半径とした円の集合を算出
  for i ← 0 to D_3.length do
    for j ← 0 to D_0.length do
      D_4 ← calculateCircle(D_0[j], D_3[i]); // D_0[j] を中心に，D_3[i] までの距離を半径とした円
      を算出
      isNearest ← true;
      for k ← 0 to peers.length do
        if contains(D_4, peers[k].location) then
          isNearest ← false;
          break;
        end if
      end for
      if isNearest then
        // D_4 に他のピアが存在しない場合，D_0[j] で取得されるコンテンツ数の期待値を算出
        D_4 ← subtract(D_4, searchedArea); // 探索済み範囲を除外
        D_4 ← subtract(D_4, D_2); // コンテンツを中心とした円の範囲を除外
        P ← pow(e, −1∗N_p∗calculateSize(D_4))∗(1−pow(e, −1∗N_p∗calculateSize(D_3[i])));
        // D_3[i] が D_0[j] のコンテンツを担当する確率を算出
        D_3[i].value ← D_3[i].value + P ∗ N_c ∗ calculateSize(D_0[j]); // D_0[j] でコンテンツ
        数の期待値を D_3[i] の探索価値として追加
      end if
    end for
  end for

  return D_3;
```

図 2.8: 探索価値計算処理 *calculateAreaValue*() の擬似コード

$$
\begin{aligned}
E_c(x,y) &= \sum_{u=1}^{m}\sum_{v=1}^{m} E_1(u,v) \\
&= (1-e^{-N_pS_1})N_cS_1\sum_{u=1}^{m}\sum_{v=1}^{m} e^{-N_pS(u,v)}
\end{aligned}
\tag{2.6}
$$

各セルの探索価値を計算する処理 *calculateAreaValue*() の擬似コードを図 2.8 に示す．

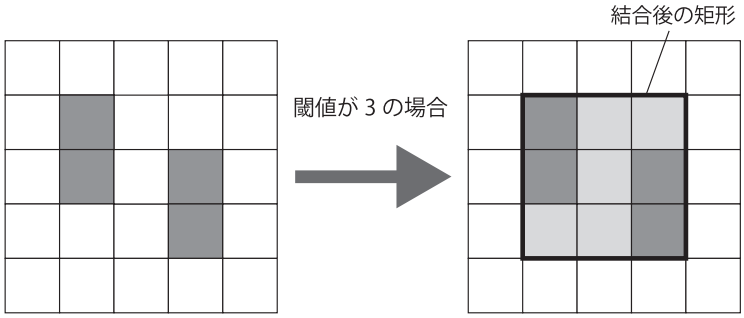


図 2.9: セルからの矩形への整形

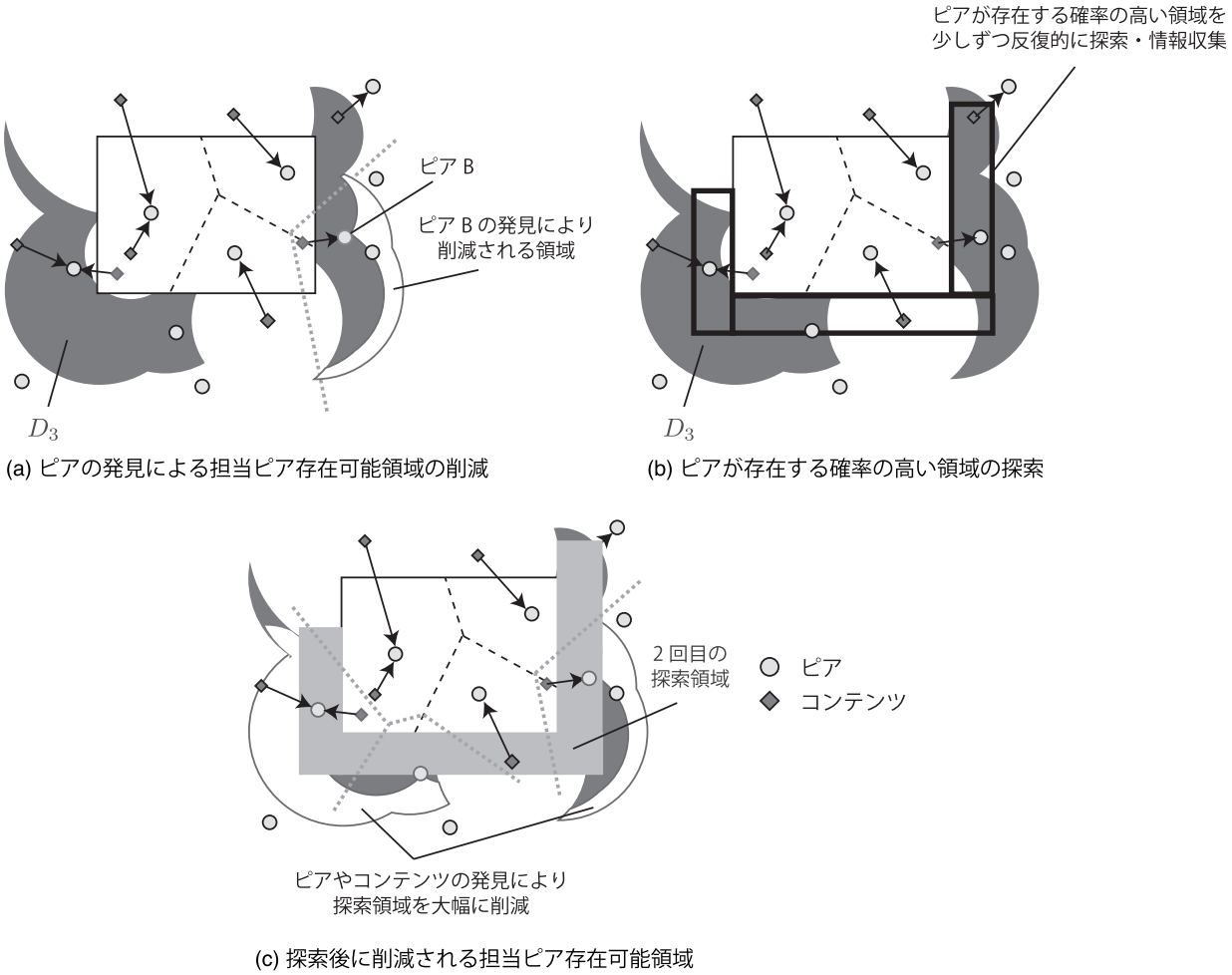
探索対象セルの矩形範囲への整形

2.4.2 節で計算した各セルの探索価値 $E_c(x, y)$ に基づき，矩形領域によってピアを探索する．このとき，各セルごとの矩形領域で探索すると，探索価値が 0 の不要な領域は全く含まれない．しかし，矩形領域が多くなり，それぞれの矩形領域をクエリで送信すると，それぞれのオーバヘッドにより探索時のメッセージが増加する．このメッセージの増加については，既知のピアまではクエリを直接送信し，探索範囲までのホップ数を小さく抑えるとしても，トラヒックへ影響を与える可能性がある．単純に矩形領域の数を削減するには，隣接しているセル同士を結合すれば同様に不要な領域は全く含まれない．しかし，探索対象となるセルの集合が 1 つの矩形領域になる可能性は低く，複数の矩形領域に分割されることになる．前述のセル単位で探索する場合と比べて矩形領域は少ないが，探索対象のセルが分散していると，矩形領域が多くなってしまう．

そこで，反復探索手法では，矩形領域の数を削減するため，本来は探索対象ではないセルも含めて矩形領域に整形する．ただし，矩形領域を少なくするために多くのセルを大きな矩形領域へ整形すると，不要な領域も探索に含まれ，無関係なピアへのクエリ送信が増加する．そのため，反復探索手法ではマンハッタン距離を閾値 l（$0 \leq l$）として指定しておき，その距離以下のセルを結合することで調整可能とする．例えば，$l = 1$ の場合は隣接するセルのみを結合し，複数の矩形の探索範囲へ整形する．$l = 3$ の場合において，対象のセルから矩形領域へ整形する例を図 2.9 に示す．

探索対象セルの反復的な探索

2.4.1 節で述べたとおり，反復探索手法では無関係なピアへのメッセージ送信を削減するため，対象となる全セルを一度で探索するのではなく，探索価値に基づいて決定した矩形



(a) ピアの発見による担当ピア存在可能領域の削減

(b) ピアが存在する確率の高い領域の探索

(c) 探索後に削減される担当ピア存在可能領域

図 2.10: 複数回による反復探索

領域の探索と情報収集を繰り返す．例えば図 2.10 (a) では，ピア B やその保持コンテンツを発見できれば，その右側の白色の部分が担当ピア存在可能領域から除外される．コンテンツ要求領域に距離が近く，対象コンテンツを担当する確率が高い図 2.10 (b) のような領域を 2 回目に探索すれば，発見されるピアやその保持コンテンツから，図 2.10 (c) のように大幅な担当ピア存在可能領域を削減できる．

各探索における領域については，探索価値の高いセルから順に前述の方法で矩形領域へ整形してゆき，未取得コンテンツに対する再現率の推定値が一定割合を越えた時点での矩形領域とする．ここで指定する再現率は，閾値 r（$0 \leq r \leq 1.0$）として設定しておく．未取得コンテンツの期待値 E とすると，E は全セルのコンテンツ数の推定値の合計となり，(x, y) を中心とするセルの探索価値 $E_c(x, y)$ を用いた以下の式で表される．

$$E = \sum_{(x,y)\in D_3} E_c(x,y) \tag{2.7}$$

ある時点で次の探索領域と決定された矩形領域を探索することによって得られる地理的コンテンツ数の期待値を E' とすると，$E' \geq rE$ となるまで，探索価値の高いセルから順に結合と矩形領域への整形を行い，次の探索領域に追加する．例えば $r = 0.5$ の場合は，未取得コンテンツのうちで 50% を取得すると推定される領域である．$r = 1.0$ の場合は，すべてのセルを対象として矩形の探索領域へ整形する．

複数の矩形で得られた探索領域を決定した後は，その領域内のピアを地理的オーバレイネットワーク層において探索し，新たに発見したピアやその保持コンテンツを既知の情報として更新する．その情報に基づき担当ピア存在可能領域を再計算し，担当ピア存在可能領域がなくなるまで，探索価値の計算，次の探索領域の決定，および探索処理を反復的に行う．ただし，反復的に探索する回数が増加すると，対象コンテンツをすべて取得するまでの時間が長くなる．したがって，許容される処理時間を考慮し，最初の数回は価値の高いセルを優先的に探索し，その後に閾値 $r = 1.0$ として探索することで，任意の反復回数以下で探索処理を終了する方法で対応する．このようにコンテンツ取得時間をある程度に抑えつつも，反復探索手法では最初の数回の探索で多くの情報を取得し，不要だと判明した範囲を探索対象から除外してゆくことで，一度に担当ピア存在可能領域を探索する場合と比べてトラヒックを削減できる．

以上の反復的なピア探索によるコンテンツ検索機能 *searchContents*() について，擬似コードを図 2.11 に示す．図 2.11 において，*calculateAreaValue*() は図 2.8 の擬似コードが対応する．

4 エリア探索手法および反復探索手法による探索範囲の例を図 2.12 に示す．図 2.12 にはピアと地理的コンテンツの配置を示しており，直線は各ピアが管理している地理的コンテンツを表している．また，図中で中央の矩形がコンテンツ要求領域を，図 2.12 (a) が担当ピア存在可能領域とそれを分割したセルを示しており，図 2.12 (b), (c) はそれぞれ 4 エリア探索手法，反復探索手法における探索領域の一例である．4 エリア探索手法である図 2.12 (b) は，太線で囲まれた上下左右 4 つの探索領域を生成する．また，反復探索手法である図 2.12 (c) では，探索領域は前述したセル間の距離 l に基づいて矩形領域へ変換されている．図 2.12 (c) は探索回によって領域を濃淡で色分けしており，この例ではコンテンツ要求領域外の探索は 2 回である．探索領域に含まれていないセルは他の領域を探索することで探索不要と判断され，それらが除外されることで，整形される矩形領域もさらに

Require:
D_0: コンテンツ要求エリアを $m \times m$ 個に分割したセルの集合
$maxSearchCount$: 反復探索における探索上限回数
l: 探索対象セルの矩形への整形において，離れたセル同士を結合するかの閾値となるマンハッタン距離
r: 反復探索において，各探索で取得しようとする未取得コンテンツの割合
Ensure:
D_0 に関するコンテンツ

$peers \leftarrow queryPeersFromRectangle(D_0)$; // オーバレイネットワークによりピアを探索
$contents \leftarrow getContents(peers)$; // 探索されたピアからコンテンツを取得
$searchedArea \leftarrow D_0$; // 探索済み範囲を D_0 に初期化
for $i \leftarrow 1$ to $maxSearchCount$ **do**
  $D_3 \leftarrow calculateTargetArea(D_0, peers, contents, searchedArea)$; // ピアが存在しうる範囲のセルを算出
  if $calculateSize(D_3) = 0$ **then**
    $break$; // ピアが存在しうる範囲がなければ，コンテンツ検索を終了
  else if $i = maxSearchCount - 1$ **then**
    $r \leftarrow 1.0$; // 探索上限回数に達していれば，次の探索ですべてのコンテンツを取得
  end if
  $D_3 \leftarrow calculateAreaValue(D_0, peers, contents, searchedArea, D_3)$; // 各セルの探索価値を計算
  $nextSearchArea \leftarrow calculateNextSearchArea(D_3, l, r)$; // コンテンツ再現率の推定値が r を越えるまで，探索価値の高いセルから順に探索範囲への追加と距離 l 以下のセルの結合を行い，次に探索する複数の矩形範囲を算出
  $newPeers \leftarrow queryPeersFromRectangle(nextSearchArea)$;
  $newContents \leftarrow getContents(newPeers)$;
  $searchedArea \leftarrow addArea(searchedArea, nextSearchArea)$; // 探索済み範囲を更新
  $peers \leftarrow addPeers(peers, newPeers)$; // 探索されたピアの情報を更新
  $contents \leftarrow addContents(contents, newContents)$; // 取得されたコンテンツの情報を更新
end for

return $filterContents(contents, D_0)$ // 取得されたコンテンツのうち，D_0 に関するものを抽出

図 2.11: コンテンツ検索機能 *searchContents*() の擬似コード

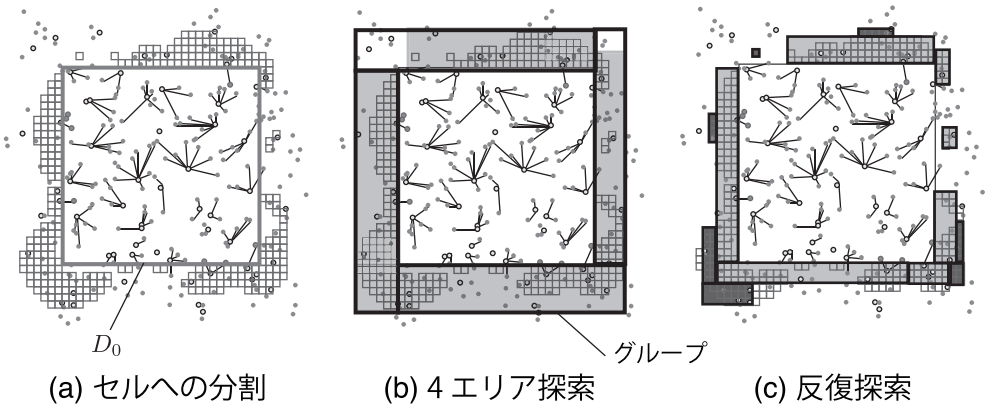


図 2.12: 探索範囲の例

小さくなっている．

2.5 評価

2.5.1 反復探索手法の設定値による比較結果

2.4 節で提案した反復探索手法では，一辺のセル分割数 m，未取得コンテンツに対する再現率 r，結合するセル間の距離 l の 3 つの設定値を与える．設定値による反復探索手法の性能について，シミュレーションによる評価を行った．未取得コンテンツに対する再現率 r については，提案手法ではコンテンツ要求領域周辺の探索が特に重要であり，ある程度の情報を収集した後の探索による影響は小さい．そのため，本シミュレーションでは，最初のコンテンツ要求領域を探索した後，2 回目の探索を r で与え，3 回目の探索では 1.0 とし，多くとも合計 3 回の探索で対象コンテンツを取得することを想定した．

シミュレーションの設定項目を表 2.1 に示す．ピアの存在密度は面積 1 あたりに存在する平均ピア数であり，全体領域にはこの密度でピアが存在する．本シミュレーションではピアの存在密度を 10，40 に設定し，全体領域には 2,560 ピア，10,240 ピアが存在する環境でそれぞれの結果を計測した．ピアの分布は，ピア存在密度にしたがい，無作為かつ一様にピアを配置した場合と，現実環境を想定し，ピアの配置に偏りがある場合で計測した．ピアの配置に偏りがある場合は図 2.13 のように，複数の正規分布にしたがってピアの疎密を一定間隔で変化させた．図 2.13 の色の濃淡はピアの疎密を示しており，それぞれの正規

表 2.1: 設定値の評価におけるシミュレーション環境

地理的オーバレイネットワーク	LL-Net
全体領域	16×16
コンテンツ要求領域	領域 1.0×1.0 で位置は無作為に決定
ピアの存在密度	10, 40
ピアの分布	一様分布，正規分布
地理的コンテンツの存在密度	40, 160
地理的コンテンツの分布	一様分布

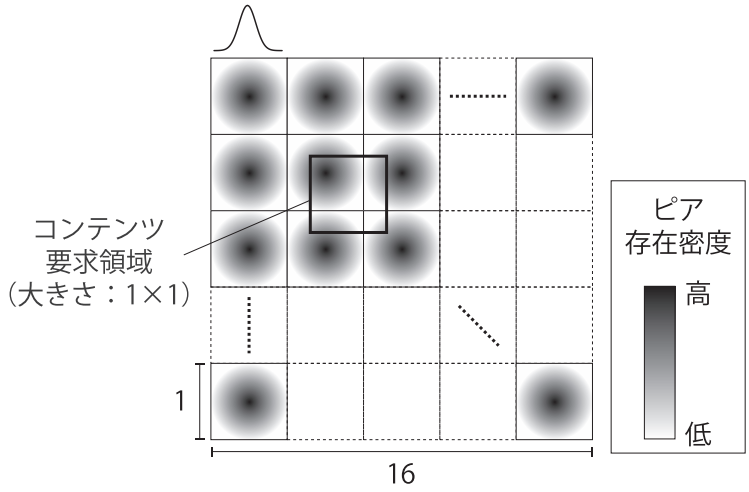


図 2.13: ピアの分布に偏りがある環境

分布は分散 $\delta^2 = 0.5$ とした．また，コンテンツ要求領域にはいずれかの正規分布の頂点が常に含まれ，ピア探索範囲が複数の分布に影響される環境を想定した．シミュレーションはそれぞれの設定値で 20 回ずつ実行し，その平均を算出した．また，変化させない場合の各設定値は，$m = 20$，$r = 0.8$，$l = 6$ とした．

各設定値による比較結果を図 2.14 から図 2.17 に示す．図 2.14 は，セル分割数 m を $m = 5, 10, 15, 20$ の 4 段階に変化させた際の探索面積の推移を表している．図 2.14 より，基本的に m を大きくするほど探索面積は小さくなるが，本シミュレーション環境では m が約 20 の段階で探索面積がほぼ収束した．一方，探索価値の計算量については，コンテンツ要求領域 D_0 上のセル数は m^2 で，D_3 上のセル数も同数の m^2 である場合を仮に考えると，D_3 上の各セルで D_0 のセル数の計算を行うため，合計で m^4 となる．前述の探索面積の減少量が収束したのに対し，m を大きくした場合の計算量の増加が大きいため，以降のシミュレーション評価ではセル分割数を $m = 20$ に設定することにした．

図 2.15 は，未取得コンテンツに対する再現率 r を $r = 0.2, 0.4, 0.6, 0.8, 1.0$ の 5 段階に変化させた際の探索面積の推移を表している．図 2.15 より，ピアの存在密度が 10 の場合は $r = 0.8$ で探索面積が最小になったが，ピアの存在密度が 40 の場合は $r = 0.6$ で探索面積が最小になった．そのため，ピアの存在密度が高いほど，探索面積の合計が最小となる再現率 r の値が小さくなると考えられる．以降のシミュレーション評価では再現率を $r = 0.8$ に固定したが，提案手法で r を調整する際，ピアの存在密度に基づいて決定することが必要である．

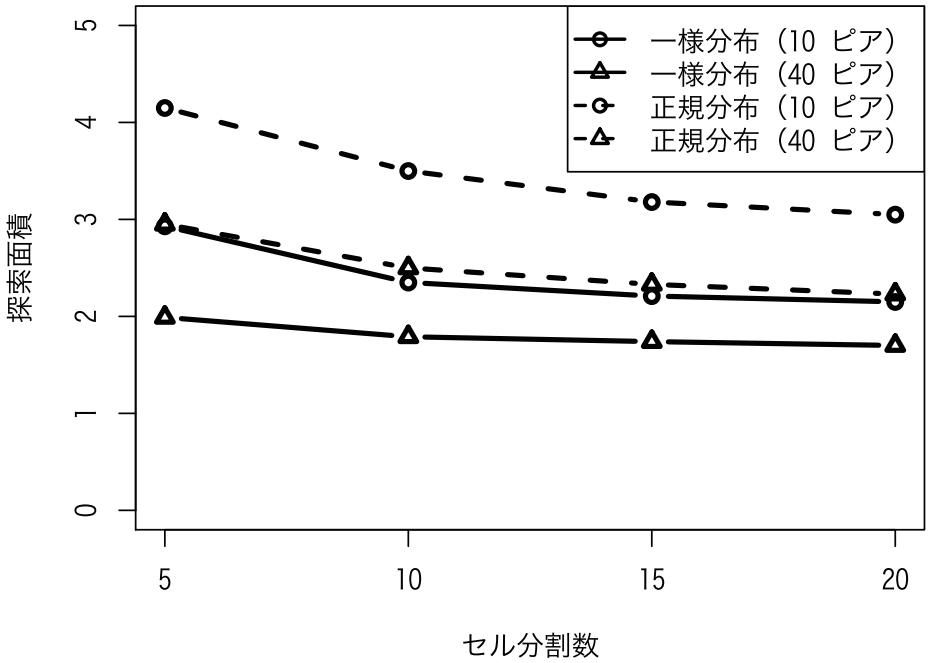


図 2.14: セル分割数による探索面積の変化

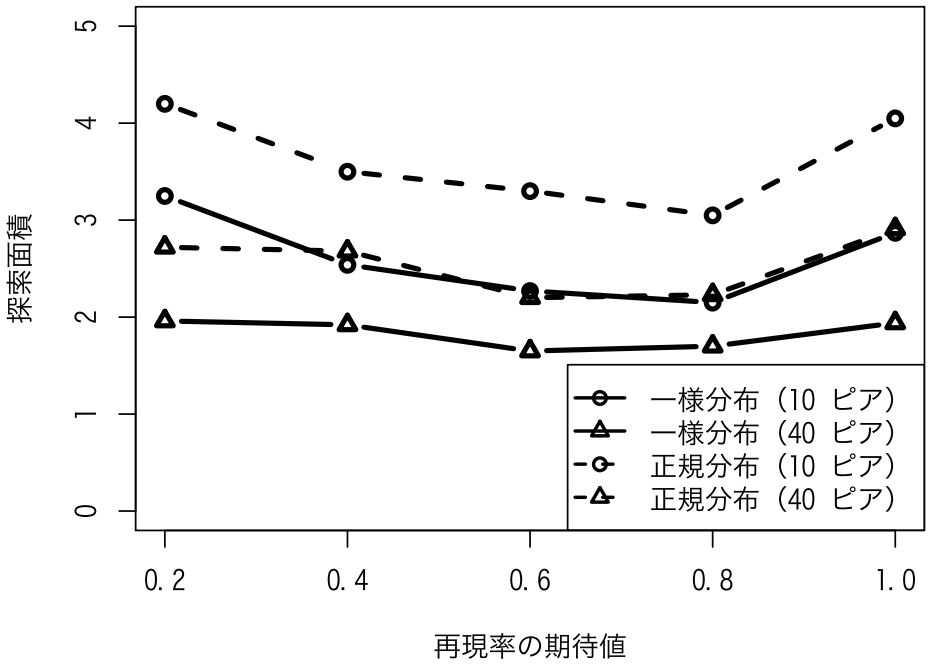


図 2.15: 再現率による探索面積の変化

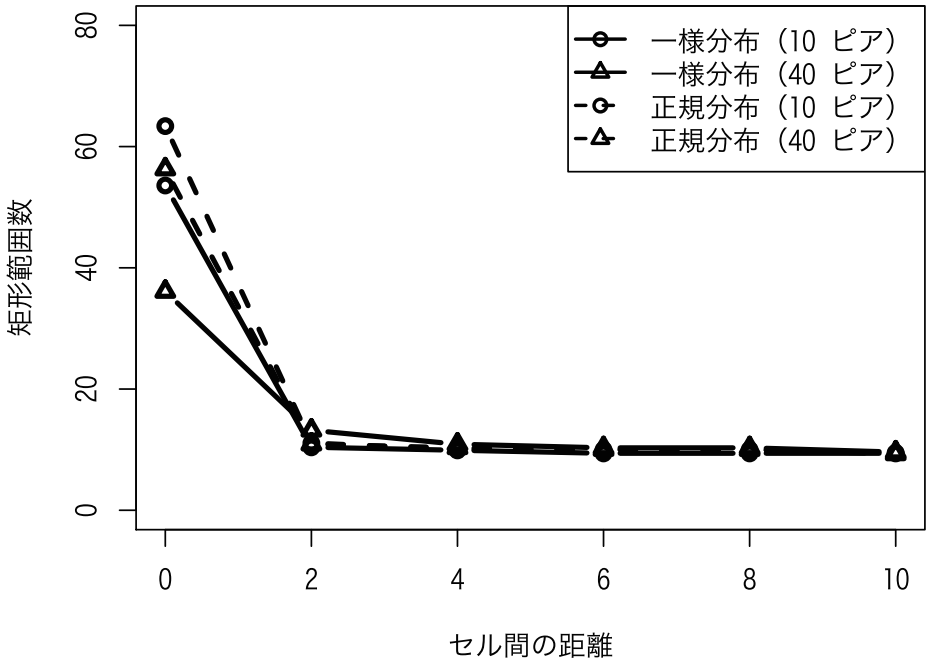


図 2.16: セル間の距離による矩形範囲数の変化

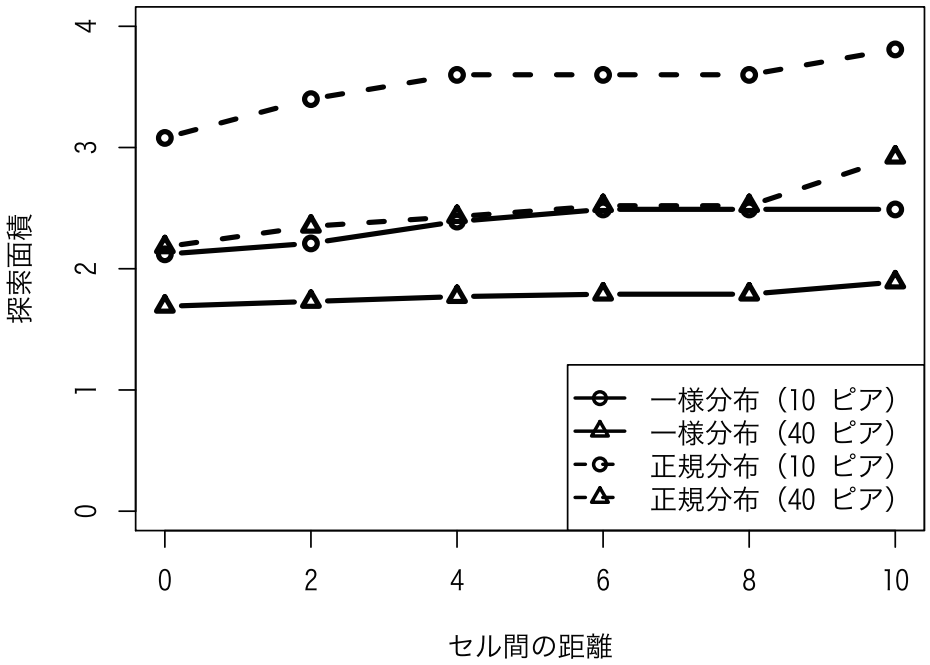


図 2.17: セル間の距離による探索面積の変化

図 2.16 および 図 2.17 は，セル間の距離 l による探索領域の矩形数および面積の推移を表している．l を大きく設定すれば探索領域である矩形の数が減少し，送信元ピアからのクエリによるオーバヘッドが小さくなるが，探索面積は大きくなり，不要な範囲のピアを探索することになる．一方，l を小さく設定すれば多くの矩形領域を探索することになり，送信元ピアからのクエリによるオーバヘッドが大きくなるが，探索面積は小さくなり，不要な領域のピアを探索対象から除外される．図 2.16 より，反復探索では合計で約 10 個の矩形領域により探索している．また，2.4.1 節の 4 エリア探索では，最初のコンテンツ要求領域 D_0 と 2 回目の 4 エリアによる合計 5 個である．一方，図 2.17 より，反復探索では $l = 6$ 周辺で探索面積が一時的に一定になっている．以上の結果より，本シミューション環境では $l = 6$ 周辺が矩形数と探索面積のバランスがよいと考え，セル間の距離を $l = 6$ に設定した．

2.5.2 探索手法による比較結果

提案する地理的コンテンツ検索手法の評価のため，シミュレーションによる評価を行った．シミュレーションの設定項目を表 2.2 に示す．本シミュレーションでは，地理的オー

表 2.2: 探索手法の評価におけるシミュレーション環境

地理的オーバレイネットワーク	ドロネーオーバレイネットワーク, LL-Net
全体領域	16 × 16
コンテンツ要求領域	領域 1.0 × 1.0 で位置は無作為に決定
ピアの存在密度	10 ∼ 40
ピアの分布	一様分布，正規分布
地理的コンテンツの存在密度	40, 80, 120, 160
地理的コンテンツの分布	一様分布
一辺のセル分割数 m	20
未取得コンテンツに対する再現率 r	1 回目: 0.8 2 回目: 1.0
結合するセルの距離 l	6

バレイネットワーク層として，地理的な近傍ノード間のリンクを基本とするドロネーオーバレイネットワーク [49] と，空間を区切った階層構造のリンクを基本とする LL-Net [25] を用いた．ピアの存在密度や分布が示す内容は前節と同様で，本シミュレーションではピアの存在密度を 10 ～ 40 まで 10 間隔で 4 段階に変化させた．全体領域には 2,560 ピア，5,120 ピア，7,680 ピア，10,240 ピアが存在する環境となる．シミュレーションはそれぞれのピアの存在密度で 20 回ずつ実行し，その平均を算出した．

反復探索手法の設定値は，前節の比較結果より，結合するセル間の距離 $l = 6$，一辺のセル分割数 $m = 20$，未取得コンテンツに対する再現率 $r = 0.8$ とした．反復的な探索回数については，前節と同様に多くとも合計 3 回の探索で対象コンテンツを取得することを想定し，最初のコンテンツ要求領域を探索した後，2 回目の探索では 0.8，3 回目の探索では 1.0 とした．

反復探索手法の比較手法については，2.4.1 節の 4 エリア探索手法と，2.3.1 節で述べた地理的コンテンツを仮想ピアとして地理的オーバレイネットワークを構築し，指定した領域の地理的コンテンツを探索可能とする手法を比較対象とした．

ピア存在密度を変化させた場合のメッセージ数について，比較結果を図 2.18 および図 2.19 に示す．図 2.18 および図 2.19 は，それぞれドロネーオーバレイネットワーク，LL-Net において探索時に発生するメッセージ数を表している．メッセージ数は探索領域に存在するピア数と要求元ピアから送信されるクエリの数に影響される．これらの評価結果より，ピアの存在密度が 10 と低い環境では，要求元から送信されるクエリの数が少ない 4 エリア探索手法のメッセージ数は反復探索手法よりも少し多い程度であるが，ピアの存在密度が高くなり，探索されるピアの数が増加するにしたがって，反復探索手法のメッセージ数との差が大きくなる．ピアの分布が正規分布の場合は反復探索手法および 4 エリア探索手法ともにメッセージ数が増加しているが，その増加量からも，反復探索手法はメッセージ数の増加を抑制していることがわかる．地理的コンテンツを 1 ピアとした場合は，反復探索手法や 4 エリア探索手法のようにコンテンツ要求領域以外のピアを探索する必要はない．しかし，本シミュレーションでは地理的コンテンツの数をコンピュータの 4 倍に設定しており，オーバレイネットワークの規模が大きくなるため，最も多くのメッセージ数が必要となっている．現実世界においても，地理的コンテンツの数はコンピュータの数に対して明らかに多いため，コンテンツを 1 ピアとした手法は適していないと考えられる．

図 2.20 および図 2.21 は，それぞれドロネーオーバレイネットワーク，LL-Net において

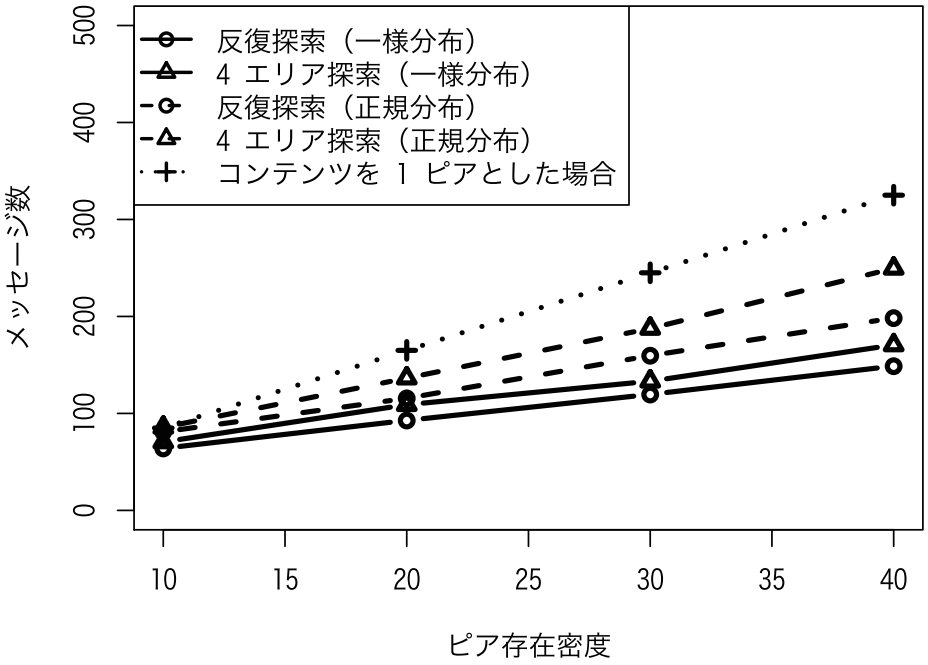


図 2.18: ドロネーオーバレイネットワークにおけるメッセージ数の変化

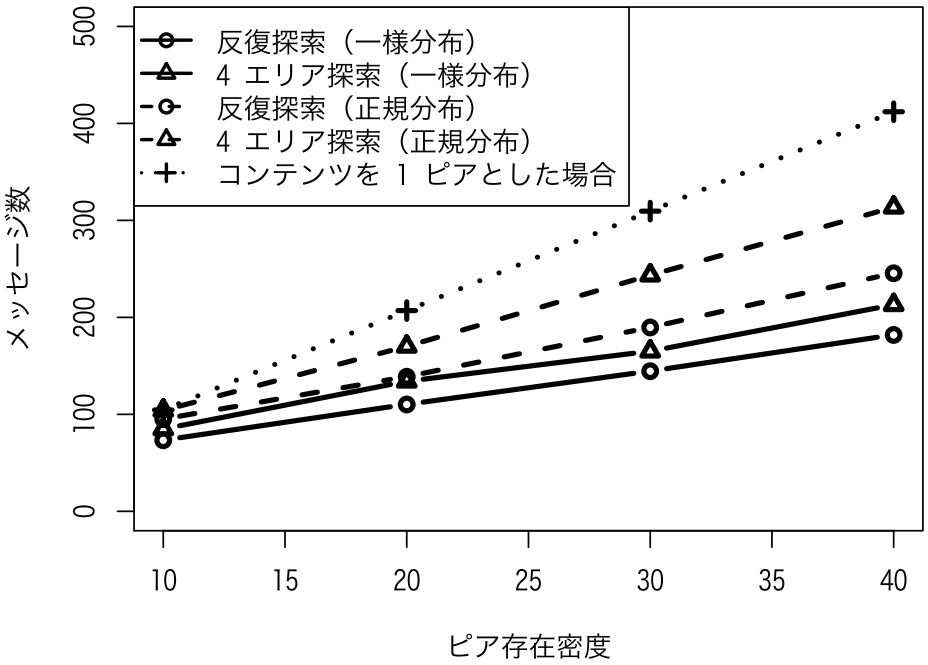


図 2.19: LL-Net におけるメッセージ数の変化

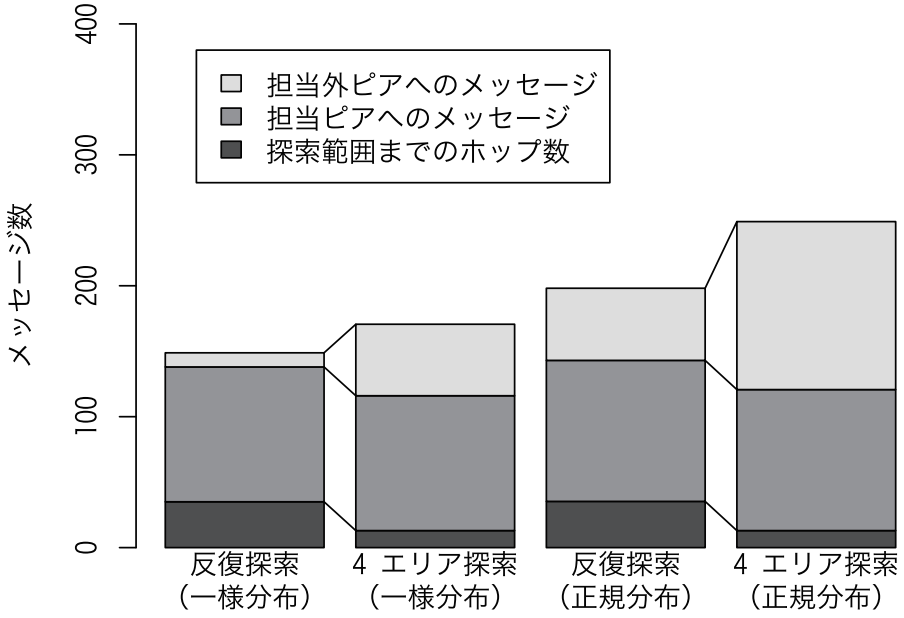


図 2.20: ドロネーオーバレイネットワークにおけるメッセージの内訳

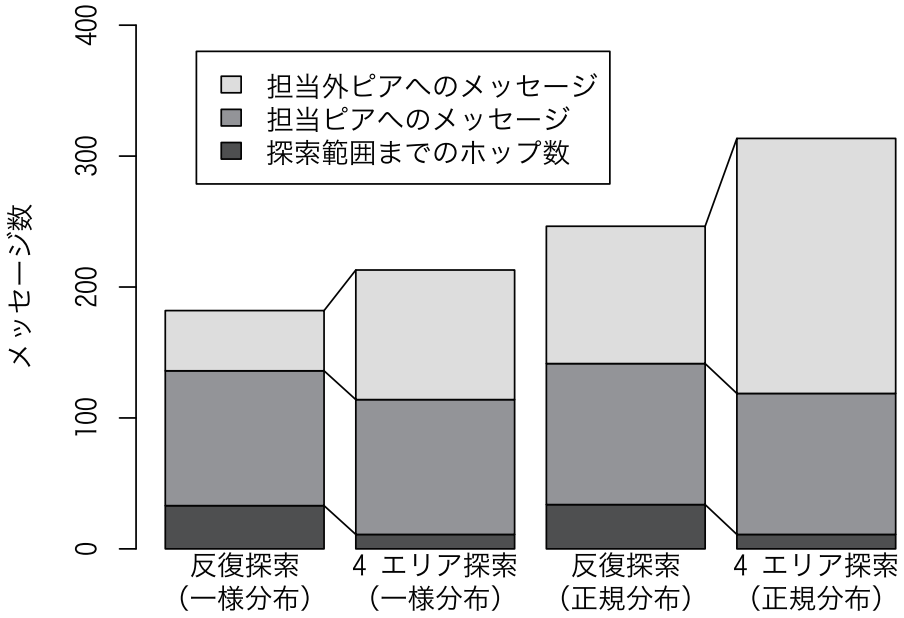


図 2.21: LL-Net におけるメッセージの内訳

ピアの存在密度が 40 の場合のメッセージの内訳で，要求元ピアから探索領域までのホップ数，コンテンツ要求領域を担当しているピアへ送信されたメッセージ数，コンテンツ要求領域を担当していないピアへ送信されたメッセージ数を表している．これらの結果より，反復探索手法はコンテンツ要求領域を担当していない無関係なピアへのメッセージ送信を抑制していることがわかる．また，ピアの分布が正規分布の環境においては，4 エリア探索手法と比べ，担当外ピアへのメッセージ数の増加がさらに抑制されていることがわかる．そのため，特にピアの分布に偏りがある実環境においては，反復探索手法がより有効であると考えられる．

2.6 むすび

本章では，地理的オーバレイネットワーク層と地理的コンテンツ管理層の 2 層からなる地理的コンテンツ分散管理手法を提案した．また，提案手法での地理的コンテンツの検索においては，既知のピアやコンテンツの配置，分布密度などから担当ピアの存在確率を計算し，その確率とボロノイ領域で表されるコンテンツ担当領域によりピア探索範囲を決定するアルゴリズムを提案した．さらに，地理的コンテンツの検索手法の有効性の評価として，シミュレーション評価結果を示した．本評価結果により，提案する反復探索手法は，適切な範囲のみのピアを探索することで，ピア数が多い環境や分布に偏りがある環境においても，コンピュータやネットワークへの負荷を抑制しつつ，必要とする地理的コンテンツを取得できることを確認した．

第3章

センサデータの意味的解釈に基づく分散センサ情報管理アーキテクチャ

3.1 まえがき

ユーザの実生活を支援するようなサービスでは，ユーザの状況や提供する情報・サービスに関連する状況を考慮し，適切かつ有益なサービスをユーザに提供することが重要である．そのようなサービスを効率的に構築するためには，ユーザが持ち歩くモバイル端末に内蔵されたセンサや，ユビキタス環境に設置された各種のセンサから得られる情報を有効活用できるような基盤技術が必要不可欠である．

センサは，状況を得たい地点すべてに存在するわけではない．例えばあるオープンカフェ周辺の状況を得たい場合に，そのオープンカフェの地点に必要な種類のセンサが存在するとは限らない．したがって，周辺に存在するセンサからうまくセンサデータを補間する必要がある．センサが十分に存在しない環境において，指定地点の推定値を得る方法については，周囲の値からの推定誤差分散を利用するクリギング法をセンサネットワーク内で分散的に行う方法 [48] や，センサネットワークごとに DB ピアと呼ばれる端末を設置し，センサデータの集約および補間を行う方法 [23] などが提案されている．しかし，これらの手法は同種のセンサデータを補間するための手法であり，それだけでは異種のセンサデータを活用して必要な状況情報を推定することはできない．

また，任意の地点のセンサデータが得られたとしても，サービスごとにそのセンサ値の解釈を行うのはサービス構築効率の観点で非効率的であるため，サービス構築のためのプ

ラットフォームとして，センサデータを直接返すだけではなく，それを意味的に解釈した状況情報を返すような機能があれば有用である．これまで，センサデータを用いたコンテキストアウェアサービスを実現するためのコンテキスト導出手法として，Ni ら [47] はセマンティックセンサネットワークを実現するためのフレームワークを提案している．また広田ら [18, 21] は，物体の情報や物体同士の関係，物体と環境との関係を論理表現に基づく推論規則によって推定する，セマンティックセンサネットワークを提案している．センサデータを推論によって状況解釈する研究 [14] も行われており，複数の情報源の関連を定義するためにオントロジを使うアプローチもある [6, 46]．センサデータにメタデータなどを付与することでシステムが情報を把握し，ルールなどに従って状況を判断可能なこれら意味的解釈に関する研究は，よりユーザに適するコンテキストアウェアサービスの実現のために重要であると考えられる．

しかし，これら既存手法はセンサデータの収集やメタデータの管理，意味的に解釈する処理を集中的に行っており，膨大な情報やユーザが存在する環境では特定のコンピュータやネットワークに負荷が集中する．また，コンテキストアウェアサービスが大規模になるにつれて，同様の状況にある複数ユーザが同様の要求を行う場合が増加する．この場合，導出される結果が同じにもかかわらず，冗長なセンサデータの収集やコンテキスト導出が行われてしまうため，非効率的である．

そこで本章では，多様な大規模コンテキストアウェアサービスに対して効果的に状況情報を提供できる分散センサ情報管理アーキテクチャを提案する．

以下，3.2 節では提案する分散センサ情報管理アーキテクチャについて説明し，3.3 節でシミュレーションによる評価について述べる．最後に 3.4 節で本章をまとめる．

3.2 分散センサ情報管理アーキテクチャの提案

3.2.1 アプローチ

ここでは，大規模コンテキストアウェアサービスにおいて，センサをピアとした地理的オーバレイネットワークを構築し，センサデータの意味的解釈を行うことでセンサ情報を分散管理するアーキテクチャおよびクエリ処理手法を提案する．

まず，指定するセンサが存在せず状況判断が行えない問題に対して，周辺に存在する他のセンサから情報を収集して意味的に把握し，演繹処理により状況判断を行う方法を提案

する．センサ情報の意味的な把握には，センサデータに付与されたメタデータを用いる．メタデータは対象とするデータについての情報が記述され．情報検索を効率化するために作成および付与されたデータである．提案するアーキテクチャでは，観測値の種類や観測時刻，観測場所，単位などがメタデータとしてセンサデータに付与されているものとする．また，必要となる種類のセンサが周辺にも存在しない場合においては，異なる種類のセンサデータを組み合わせて意味的に解釈することで，当該地域の状況に関する情報を導出することを考える．

次に，センサデータにおける重複処理の問題に対しては，演算結果をもとに仮想センサを設置し，その仮想センサに処理結果を保持させる手法を提案する．処理結果を保持して再利用することで処理回数が減少し，センサやネットワークへの負荷を軽減することができる．

3.2.2 センサ情報管理の階層化

各階層における処理

提案する分散センサ情報管理アーキテクチャでは，コンテキスト導出に必要なセンサデータを意味的に把握することと，要求される場所のセンサデータを得ることの 2 つの機能が必要となる．これらの 2 つにセンサデータを収集する機能を加え，それぞれを明確に分離する．提案する分散センサ情報管理アーキテクチャは，センサデータをそのまま扱う Raw Layer（RL），センサデータを空間補間する Abstract Layer（AL），空間補間されたデータから演繹処理により意味的解釈を行う Logical Layer（LL）の 3 階層で構成される．以下，各層について説明する．

- Raw Layer (RL)

  センサデータをそのまま扱う層である．RL では雨量センサや温度センサなどの異種センサが多数分散して配置されており，各センサは周囲の環境を監視し，観測値を保管する．この層のセンサを Raw Sensor と呼ぶ．また，各 Raw Sensor は自身の観測値の種類，観測場所，観測値という情報をもつ．例えば，位置 $(2, 3)$ に設置されている雨量センサが雨量 23（ml/時間）を測定した場合,「観測値の種類: 雨量，位置 $(2, 3)$，値: 23ml」というセンサデータを保持する．センサデータは，位置と観測値の種類を指定して検索できる．

- Abstract Layer (AL)

  地理的な距離をもとにセンサデータを補間した情報を提供する層であり，上位の LL に対して任意地点のセンサデータを提供する層である．この層では，LL からのセンサデータ取得要求に対して，要求された位置のセンサデータを取得し，応答する．また，要求する位置に Raw Sensor が存在しない場合は，周囲に存在する Raw Sensor 情報を利用することで，その位置に欠ける Raw Sensor 情報を補間する．この補間処理は，その位置からある一定距離の領域に存在する Raw Sensor 値から特定の計算モデルに基づいてその位置の推定値を計算することにより実現する．提案アーキテクチャでは，最近傍法や距離による加重平均，クリギング法などの空間補間による計算モデルを想定する．また，推定値を計算するために収集対象とするセンサデータの距離は，計算モデルによって定義されるものを用いる．センサが十分に存在しない環境において，指定地点の推定値を得る方法については，クリギング法をセンサネットワーク内で分散的に行う方法 [48] や，センサデータの集約および補間を行う方法 [8,23] などが既に提案されている．推定値の計算にはこうした既存手法の適用も可能と考える．

- Logical Layer (LL)

  ユーザからのクエリの意味的解釈を行い，概念を定義したルールを用いた演繹によって，必要となるセンサの種類を導出する．これらの関連するセンサデータを AL から取得し，状況を判断することで，クエリに応答する．AL，LL の組み合わせによって，複数の推定値に基づくルールを用いた状況判断が可能となる．このとき，ルールを用いることによって，単一種類のセンサのみならず，複数の異種センサの推定値を用いた状況判断が行える．例えば雨という状況を推定する際，雨量センサが存在しなければ気圧センサを用いるといったことが可能であり，単一種類のセンサのみを用いた場合と比べ，センサ不在の問題をより広範囲で解決できる．これらのルールについては，あらかじめこの層の処理モジュールとしてセンサに組み込まれていることを想定する．

クエリに対する処理の例

提案アーキテクチャでは，図 3.1 のような木構造のルールを想定する．図 3.1 は，遊びに訪れるのによい場所かどうかの「$SuitablePlace(x, y)$」に関するルールである．「$SuitablePlace(x, y)$」というクエリを受信した場合，まず木構造のルールにより LL で $SuitablePlace$ の意味を

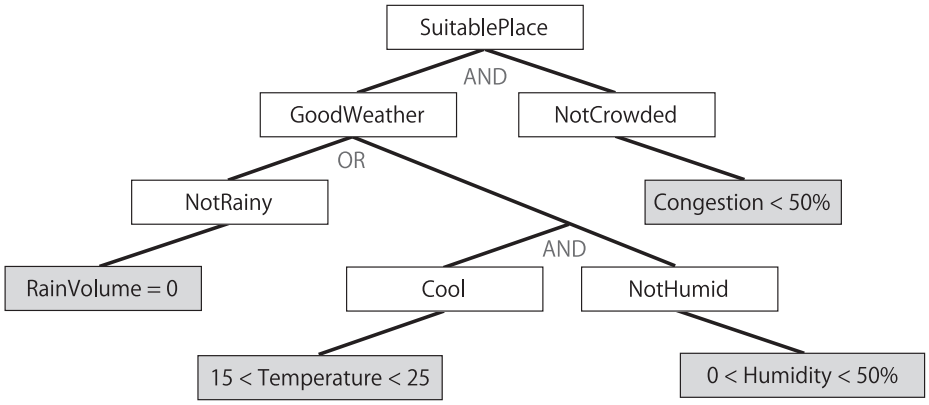


図 3.1: ルールの例

解釈し，必要であると導出されたセンサデータを AL に要求する．AL ではこの LL からの要求に対し，必要となる地理的な範囲のセンサデータを RL で収集し，当該位置におけるセンサの観測値を補間処理により推定する．得られた推定値を AL がセンサデータとして LL に返し，LL はそのセンサデータを用いて判断することでクエリに応答する．図 3.1 の例では，クエリに応答するために必要となるセンサデータは色付けした「*RainVolume*」,「*Temperature*」「*Humidity*」「*Congetion*」の 4 箇所で示している．

階層化された分散センサ情報管理アーキテクチャおよびクエリに対する処理の具体例を図 3.2 に示す．図 3.2 では図 3.1 のルールを用いている.「*SuitablePlace*$(2,3)$」というクエリに対しては,「よい場所（*SuitablePlace*）」とは「よい天気（*GoodWeather*）かつ混雑していない（*NotCrowded*）」，そして「よい天気」とは「雨が降っていない（*NotRainy*）」または「涼しい（*Cool*）かつ蒸し暑くない（*NotHumid*）」であると LL において導出される．つまり,「よい場所」に関連する単純な意味的状態は「雨ではない」,「涼しい」,「蒸し暑くない」,「混雑していない」という 4 つの状態であると解釈し，各状態に関するセンサデータを AL に要求する．必要となるセンサデータは,「*RainVolume*」,「*Temperature*」「*Humidity*」「*Congetion*」の 4 種類である．

- ルール 1:

  $SuitablePlace \leftarrow GoodWeather$ and $NotCrowded$

- ルール 2:

  $GoodWeather \leftarrow NotRainy$ or $(Cool$ and $NotHumid)$

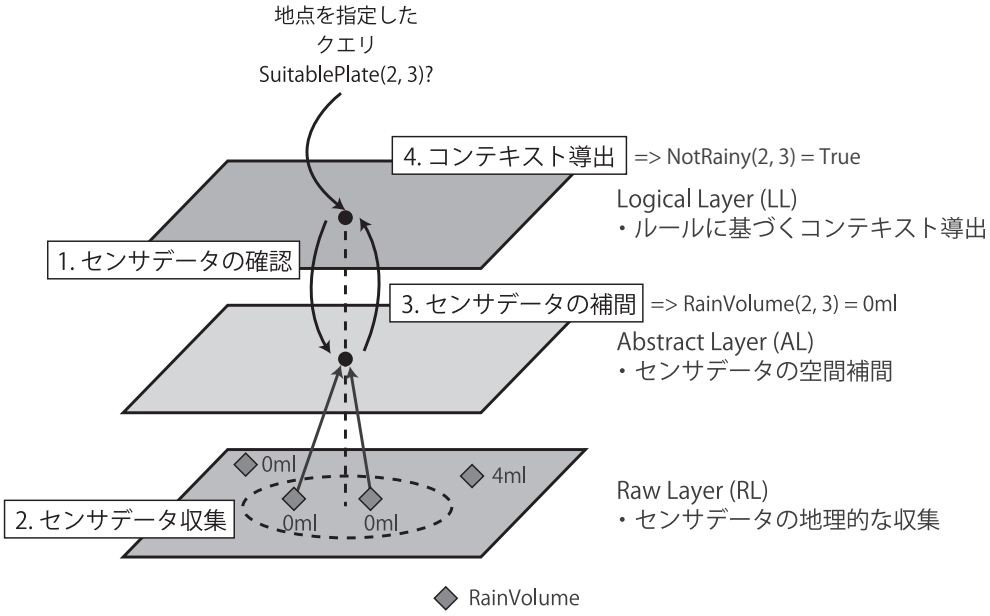


図 3.2: 階層化されたセンサ管理の構造

AL では，LL から要求されたセンサデータを取得する．図 3.2 に示した *RainVolume* の場合を例として述べる．まず *RainVolume* は雨量センサと関連し，位置 $(2, 3)$ の雨量センサの観測値を探索する．位置 $(2, 3)$ には雨量センサが存在しないため，対象の位置からユーザやアプリケーションが決めた範囲，例えば半径 100m の領域内の RL に存在する雨量センサの観測値を収集し，雨量センサ用の計算モデルによって観測値の推定値を算出する．図 3.2 では収集した観測値はすべて 0 であるため，位置 $(2, 3)$ における推定値も 0 で，推定された観測値をセンサデータとして LL に応答する．同様に *Temperature*，*Humidity*，*Congetion* という単純な意味的状態に関するセンサデータも取得し，LL に応答する．LL では，取得されたセンサデータから $NotRainy = \mathrm{True}$，$NotCrowded = \mathrm{True}$，$Cool = \mathrm{False}$，$NotHumid = \mathrm{True}$ などの単純な意味的状態を導出し，ルールに基づいて演繹を行う．まず，前述のルール 2 より，$GoodWeather = \mathrm{True}$ の結果が導出される．その後，前述のルール 1 に基づいて演繹を行い，$SuitablePlace = \mathrm{True}$ の最終結果をユーザに応答する．

3.2.3 処理結果の共有

仮想センサの生成

3.2.2 節で提案したセンサ情報分散管理アーキテクチャでは，複数ユーザが類似するクエリを送信すると，結果が同じにもかかわらず，RL におけるセンサデータ収集処理や AL における補間処理，LL におけるルール処理などの冗長な処理が行われてしまう場合がある．

そこで，任意のクエリの処理結果に対して仮想センサを生成し，類似するクエリが発生した場合に仮想センサが応答することでこれらの処理を抑制する方法を提案する．仮想センサを効率的に使用するためには，その仮想センサの情報を容易に取得できることが必要である．そのため，クエリに基づいて発見される可能性を考慮し，元のクエリの位置から地理的に一番近い位置にある Raw Sensor 上に仮想センサを生成する．AL の処理結果をキャッシュとして提供する仮想センサを AVS (Abstract Virtual Sensor)，LL の処理結果をキャッシュとして提供する仮想センサを LVS (Logical Virtual Sensor) と呼び，クエリに対して応答可能な AVS や LVS があれば，各層の処理を行わずに仮想センサの結果を利用する．

図 3.2 の例により生成される仮想センサを図 3.3 に示す．

LL や AL におけるクエリに対して，仮想センサは要求する位置にそのセンサが存在しているように応答する．さらに，LVS は対象とする状況，位置，値，有効半径を属性としてもち，AVS は値の種類，位置．値，有効半径をもつ．有効半径は仮想センサの位置を中心とした円形の領域を表すものとし，この領域内において，仮想センサの値の信頼性が高いという意味をもつ．したがって，クエリの指定した位置が有効範囲に含まれれば，その仮想センサが応答するものとする．

このとき，仮想センサが応答する値は必ずしも正確ではなく，AL においては AVS の値を用いず Raw Sensor を用いた補間処理で得られる値と AVS から得られる値の間に誤差が生じたり，LL においては LVS を用いずにルールから導出した結果と LVS から得られる結果との間に違いが生じたりする場合がある．例として，図 3.3 で生成された仮想センサが応答するクエリの例を図 3.4 に示す．図 3.4 では，別のクエリ 1，2 ともに図 3.3 で生成された仮想センサの有効範囲内で，$NotRainy = \mathrm{True}$ と応答される．しかし，RL で Raw Sensor のセンサデータを収集し，AL で補間すると，クエリ 2 の位置の値は 0 ではなく，LL で $NotRainy = \mathrm{False}$ となる．ここで，図 3.4 に示した LL 上の点線は，仮想センサを使用せずに任意の地点で周辺 Raw Sensor からのセンサデータ収集および AL での補間を

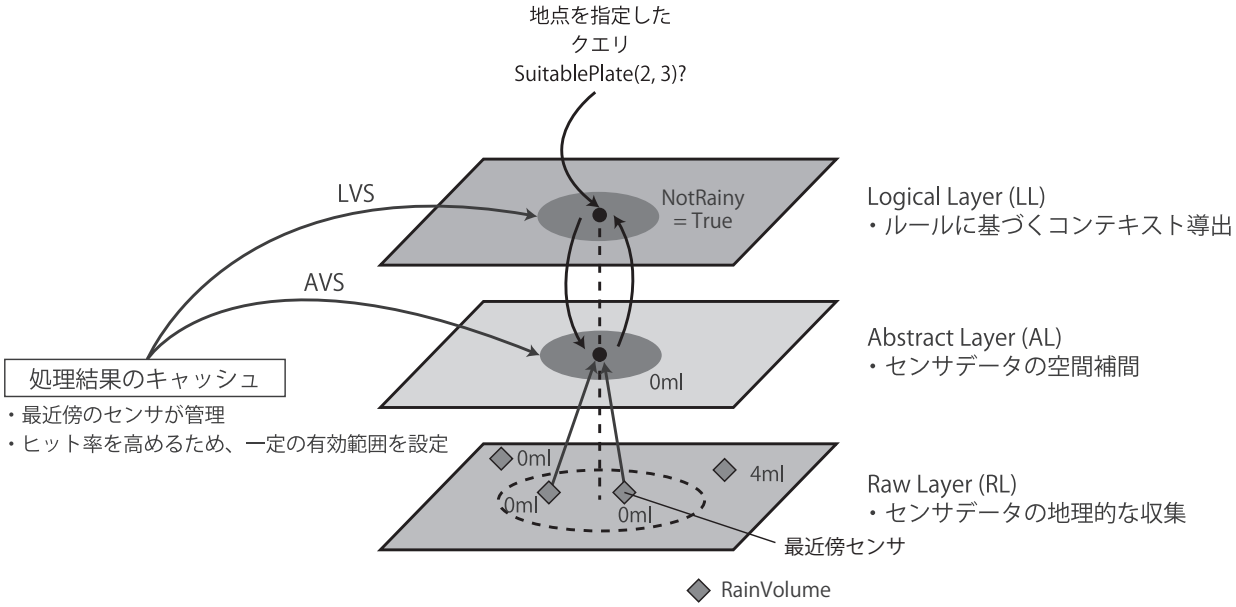


図 3.3: 仮想センサの設置

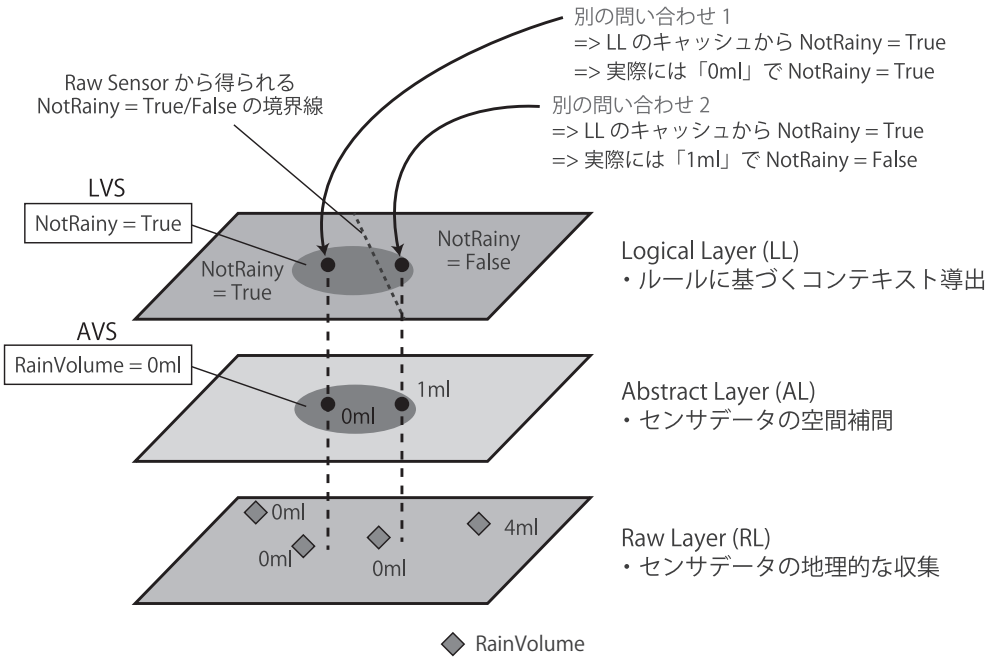


図 3.4: 仮想センサの応答

行うことで得られる，真の演繹結果の境界線を表している．このように仮想センサによる処理結果の共有は，Raw Sensor のセンサデータ収集を抑制できる一方，誤った結果が応答される可能性がある．したがって，仮想センサの有効範囲を大きくすればより多くの処理やメッセージ数を削減できるが，誤った結果を返す可能性も高くなるため，適切な有効範囲を設定する必要がある．

仮想センサの更新

仮想センサの値の精度を維持するには，Raw Sensor の値の変化を反映する必要がある．そのため，LVS は自身に関連する AVS の更新情報を取得する．このとき，物理的に LVS が存在するセンサ，つまり LVS を生成した Raw Sensor と，関連する AVS が存在する Raw Sensor が異なる場合，LVS が AVS から更新情報を頻繁に取得すると大量のトラヒックが生じてしまう．また，各 LVS においても，AVS から取得した情報をもとに再演繹処理を行うことになる．

しかし，AVS に更新がない場合や，一部の AVS に更新はあるが LVS の再演繹結果には影響しない場合がある．そこで，更新処理による負荷の増大を抑制するため，LVS の結果に影響する AVS を抽出し，それらの AVS の値が更新した場合のみ LVS に通知させる．LVS の結果を変化させる AVS の抽出は，LL のルールに基づいて導出する．

知識に基づく推論手法は，前向き推論と後ろ向き推論がある．前向き推論とは，ある事実が成立するかを調べる際，既知の事実をもとにルールから新しい事実を増やしながら調べる手法である．一方，後ろ向き推論は，成立するかを調べたい事実に対して，成立するために必要な事実をルールから調べながら進む手法である．提案アーキテクチャでは特定の事実が成立しているかを問い合わせるため，後ろ向き推論手法が適切である．

後ろ向き推論では，結果（結論）をルートノードとして，推論を行いながら推論木を作成する．まず，結果が成立するための条件をルートノードの子ノードとして追加する．それぞれの子ノードに対しても，それらが成立するための条件を抽出し，対象ノードの子ノードとしてさらに追加する．この作業を繰り返し，最終的に導出されるすべての条件が事実として成立すれば，ルートノードの結果が成立する．

図 3.5 はルールを表す推論木の表記例である．この推論木では，すべてのノードが真（True）と偽 （False）の値をもち，推論規則に従った論理演算によって成り立つ．ルール「$A \leftarrow B$ or C」では，B または C が真であれば A も真である．このルールを推論木で表

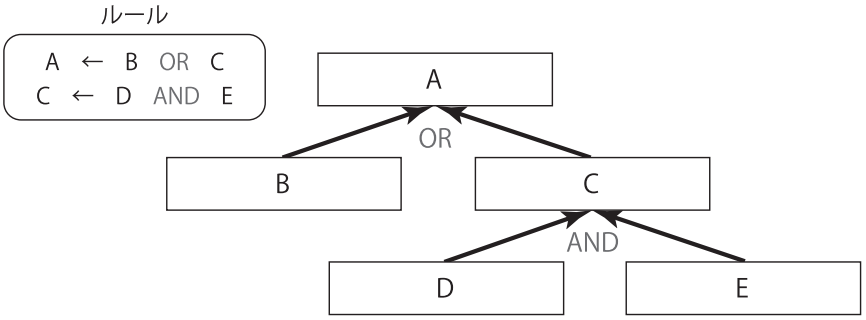


図 3.5: ルールと推論木の表記例

すと，B および C は ノード A の子ノードになる．また，ノード B と C は論理和 OR の関係をもつ．ルール「$C \leftarrow D$ and E」も同様に親子関係となり，ノード D と E は論理積 AND の関係をもつ．

図 3.5 に示す推論木において，葉ノードに事実を挿入した際の導出結果を図 3.6 に示す．図 3.6 の (a) ～ (d) はそれぞれ，葉ノード B，D の事実の組合せが（True, True），(True, False），(False, True），(False, False）の場合である．葉ノード E については，すべての場合で True である．このように，推論木では葉ノードの状態によってルートノードの結果が決定される．提案アーキテクチャでは，葉ノードは「$RainVolueme = 0$」のセンサデータから導出される「$NotRainy$」などの単純な意味的状態となる．

ここで，推論木における葉ノードの変化が上位ノードにどのように影響するかを検証する．例えば，図 3.6 (a) と (c) の比較すると，True であるノード C と OR で結ばれたノード B が変化しても，ノード A は常に True のままである．一方，図 3.6 (c) と (d) を比較すると，ノード E と AND で結ばれたノード D が True から False へ変化すればノード C が False となる．ノード C は False であるノード B と OR で結ばれており，ノード C が False となることで，ノード A も False となる．このように，対象とするノードが True であれば，下位に存在する True の葉ノードの変化のみが対象ノードの結果に影響する．同様に，対象ノードが False であれば，False の葉ノードの変化のみが対象ノードの結果に影響する．つまり，特定のノードに下位ノードの更新を反映するには，対象とするノードの下位に存在する葉ノードのうち，現在の自身の状態と異なるノードのみの変化を監視すればよい．

上述の考え方に基づき，提案アーキテクチャでは仮想センサを更新する仕組みを導入する．LVS の結果と関連する AVS は，それぞれ推論木におけるルートノードと葉ノードに

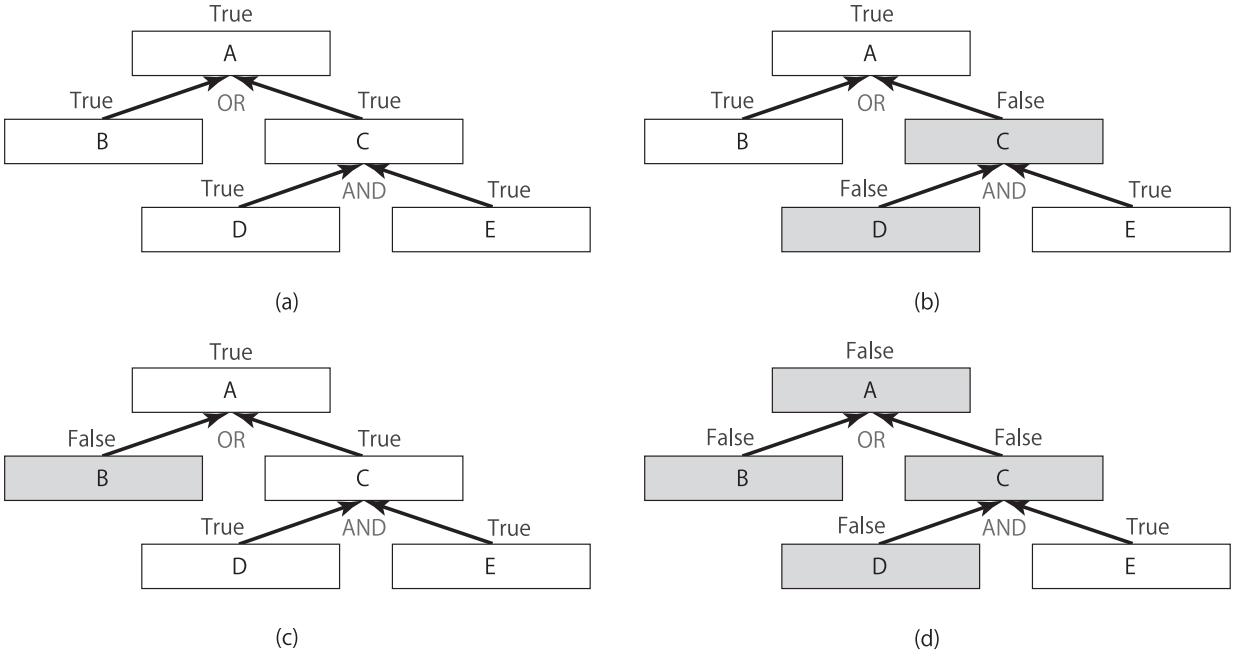


図 3.6: 仮想センサの更新

相当する．そのため，LVS の結果の変化を管理するには，LVS を変化させられる AVS の変化のみを監視する．具体的な更新処理は以下のように行う．

- LVS は自身が生成された際，監視する必要のある AVS を指定する．
  - LVS の結果が False であれば，関連する AVS の中で False の結果をもつ AVS を監視対象として指定する．
  - LVS の結果が True であれば，関連する AVS の中で True の結果をもつ AVS を監視対象として指定する．
- AVS は定期的に Raw Layer からセンサデータを収得し，自身で更新を行う．さらに，LVS に監視対象として指定された AVS は，自身の結果に変化があれば LVS に通知する．

以上の更新処理によって，LVS は関連する AVS のすべてから更新情報を取得する必要がなくなり，LVS と AVS 間に発生するトラヒックや再演繹処理を抑制できる．

仮想センサの破棄

仮想センサを生成しすぎると，前節で述べた仮想センサの更新によるトラヒックが増大する問題が発生する．そのため，あまり使用されない仮想センサや同じ情報をもつ仮想センサなどの不要な仮想センサを破棄する必要がある．この仮想センサの破棄を実現するため，各仮想センサに TTL（Time To Live; 生存時間）を指定する．TTL の操作は以下の手順で行い，あまりアクセスされない仮想センサを破棄できる．

1. アクセス頻度が大きいほど TTL を大きく指定する．この手法では一定間隔におけるアクセス数に基づいて，TTL を再指定する．つまり，平均アクセス頻度が大きいほど，仮想センサの存在時間が長くなる．例えば，1 時間の平均アクセス数 n を計測しておき，TTL になった時点で以下のように TTL を再指定する．
   - $10 \leq n$: TTL = 1 時間
   - $5 \leq n \leq 10$: TTL = 30 分
   - $n \leq 5$: 消滅
2. 各頻度の TTL の指定時間はあらかじめ設定しておき，仮想センサにクエリが届いた際，TTL をその時間に再指定する．つまり，最後にアクセスされた時刻からその指定時間アクセスされなければ仮想センサを破棄する．

3.3 実装および評価

3.3.1 実装

提案する分散センサ情報管理アーキテクチャに基づく状況判断システムの実装を PIAX (Peer-to-peer Interactive Agent eXtentions) [65] 上で行った．PIAX は，オーバレイネットワーク上に分散エージェント環境を構築可能な Java ベースの P2P プラットフォームシステムである．PIAX におけるエージェントとは，P2P ネットワーク上に自律分散し，相互に連携可能なソフトウェアモジュールである．PIAX では，エージェントの連携によりサービスを構成することで，エージェントの変更のみで，扱う情報資源や属性の形式，処理アルゴリズムの変更を柔軟に行うことができる．また，構築したシステムでは，LL 層におけ

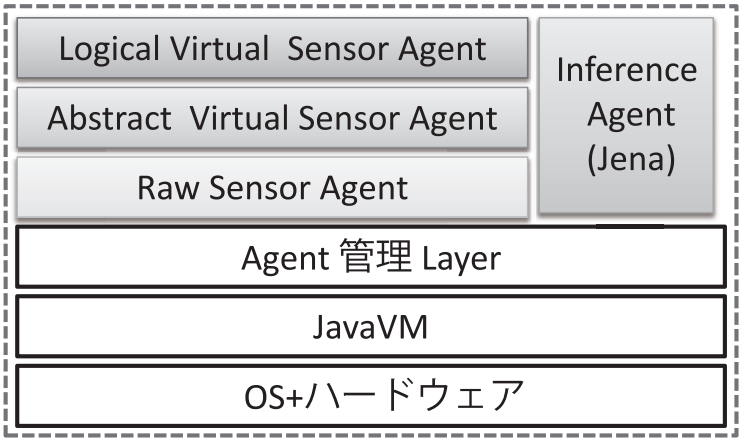


図 3.7: 実装システムの構成

る演繹処理を行うためのモジュールとして，Java によるセマンティックウェブアプリケーション開発が可能な JenaAPI[1] を採用した．構築したシステムの構成を図 3.7 に示す．

図 3.7 の下の 3 階層は PIAX の構成を示している．ここで，Agent 管理 Layer の上層は Agent 層であり，この層では Logical Virtual Sensor Agent (LVS Agent), Abstract Virtual Sensor Agent (AVS Agent), Raw Sensor Agent (RS Agent) と Inference Agent がある．LVS Agent, AVS Agent, RS Agent は，それぞれ LVS, AVS, Raw Sensor の処理を担当する．Inference Agent はユーザのクエリに対して応答するエージェントで，LVS Agent や AVS Agent, RS Agent への問い合わせや LL における演繹処理，AL における空間補間を行う．LVS Agent, AVS Agent, RS Agent は，Inference Agent からの問い合わせに対して，該当する情報をもっていれば応答する．また，AL においてセンサデータを補間する計算モデルについては，指定地点を含む三角形の頂点となる 3 つの近隣センサデータを取得し，各頂点からの距離による加重平均で推定値を計算するモデルを実装した．

ユーザのクエリに対する Inference Agent の処理の流れは以下のとおりである．

1. クエリの受信

   ユーザからのクエリを受信すると，指定位置を有効範囲に含むかを LVS Agent に問い合わせる．LVS Agent からの応答がある場合はその結果をユーザに応答する．一方，応答がない場合は処理 2 を行う．

2. LL における演繹処理

[1] http://jena.sourceforge.net/

クエリの内容に対して LL のルールに基づいて演繹処理を行い，対象となる AVS Agent が指定位置を有効範囲に含むかを問い合わせる．AVS Agent からの応答があれば，その結果をルールと照合し，導出された結果をユーザに応答する．その際，導出された結果から LVS Agent を生成する．一方，対象となる AVS Agent からの応答がない場合，処理 3 を行う．

3. AL における空間補間

   指定位置から一定範囲内に関連する RS Agent に問い合わせ，RL において取得したセンサデータを特定の計算モデルに基づいて空間補間する．その結果を用いて AVS Agent を生成し，処理 2 に移行する．

3.3.2 仮想センサの効果の評価

実装した状況判断システムを用いて，提案手法の有効性を評価するシミュレーションを行った．本シミュレーションでは，多くの異種センサが配置されている環境において，仮想センサの有効範囲の大きさによる効果を比較評価した．本シミュレーションの設定項目を表 3.1 に示す．地理的オーバレイネットワークは LL-Net を用い，Raw Sensor を 1 ピアとして構築した．Raw Sensor は，雨量センサと湿度センサ，温度センサ，混雑センサの 4 種類のセンサを想定する．それらのセンサの合計数は 1,000 に設定し，各センサの種類および配置は無作為に決定される．

表 3.1: シミュレーション環境

項目	値
全体領域	10km × 5km
Raw Sensor 数	1,000
Raw Sensor の配置	無作為に決定
RL でのセンサ探索範囲	500m
累計クエリ数	2,000
仮想センサの有効範囲	0m, 100m, 200m, 500m
計測値	メッセージ数，正答率

```
Logical Layer Rule
 [Point:(?M xAxis ?x)(?M yAxis ?y) ->
  [rule0: (?M Status SuitablePlace) <-
   (?M GoodWeather True),
   (?M NotCrowded True)]
  [rule1: (?M GoodWeather True) <- (?M tenki NotRainy)]
  [rule3: (?M GoodWeather True) <-
   (?M tenki NotHumid),
   (?M tenki Cool)]
  [rule4: (?M tenki NotHumid) <- checkStatus(?x,?y,NotHumid)]
  [rule5: (?M tenki NotRainy) <- checkStatus(?x,?y,NotRainy)]
  [rule6: (?M tenki Cool) <- checkStatus(?x,?y,Cool)]
  [rule7: (?M NotCrowded True) <- checkStatus(?x,?y,NotCrowded)]
  [rule8: (?M status Cool) <-
   getSensorData(temperatureSensor,?x,?y,?data),
   smaller(?data,25),greater(?data,15)]
  [rule9: (?M status NotRainy) <-
   getSensorData(rainSensor,?x,?y,?data),
   smaller(?data,0)]
  [rule10: (?M status NotHumid) <-
   getSensorData(humidSensor,?x,?y,?data),
   smaller(?data,50)]
  [rule11: (?M status NotCrowded) <-
   getSensorData(personCountSensor,?x,?y,?data),
   smaller(?data,50)]
 ]
```

図 3.8: Jena で使用するルール例

クエリについては，全体領域内における無作為な位置 (x, y) に対して,「$SuitablePlace(x, y)$」を発行することとした．$SuitablePlace(x, y)$ クエリに応答する導出ルールは図 3.1 で示すとおりとする．実装システムで用いた Jena において図 3.1 のルールに対応する，任意の地点 M (x, y) に関するルールの例を図 3.8 に示す．

本シミュレーションでは，仮想センサが存在しない状態から開始し．ユーザから送信されるクエリの応答処理を行った後，必要に応じて仮想センサを生成する．つまり，ユーザからのクエリに対する処理回数が増加するのに伴い，新たに設置される仮想センサの数も増加する．本シミュレーションでは 2,000 個のクエリを順次に与え，仮想センサの有効範囲が 0m，100m，200m，500m の環境において，クエリを処理する際に LL-Net 内で発生したメッセージ数の累計およびクエリに対する応答の正答率を精度としてそれぞれ計測した．仮想センサの有効範囲が 0m の場合，仮想センサの地点に関するクエリに対してのみ，

その仮想センサの値が有効となる．本シミュレーションではクエリの位置は全体領域内で無作為に決定しているため，過去に同じ地点のクエリが送信され，仮想センサが生成されている地点に再度クエリが送信される確率は低い．つまり，仮想センサの有効範囲が 0m の結果は，仮想センサを利用しない場合とほぼ等しい．LL-Net 内で発生するメッセージ数については，ユーザが要求する位置に複数の仮想センサの有効範囲が重なる場合，複数の仮想センサから応答メッセージが送信される．1 クエリに対して複数の仮想センサから応答を得た場合，各仮想センサの応答を 1 メッセージとして計測した．また，クエリに応答する仮想センサが存在しない場合，クエリに応答するために複数の Raw Sensor からセンサデータを収集する．この場合は各 Raw Sensor からの応答を 1 メッセージとして計測した．正答率については，RL におけるセンサデータのみから導出される結果を正解とし，仮想センサも含めて導出される結果が正しいかをクエリごとに計測した．

仮想センサの有効範囲の変化に対するトラヒックの変化を図 3.9 に示す．グラフの横軸はクエリ数で，縦軸は発生したメッセージの累計である．図 3.9 より，仮想センサの有効範囲が大きくなるほど，仮想センサを設置しない環境よりも多くのメッセージを削減できていることがわかる．有効範囲が 500m の場合は仮想センサを設置しない場合と比較して，20% 以下になっている．同様に仮想センサの有効範囲の変化に対する正答率の変化を図 3.10 に示す．グラフの横軸はクエリ数で，縦軸は導出された結果の正答率を表す．図 3.10 より，仮想センサの有効半径が一定の状況ならば，ユーザからのクエリが増加するのに伴い，正答率が減少していることがわかる．これは，クエリが増加するほど仮想センサの生成数も増加し，有効範囲をもった仮想センサが誤った値をクエリに応答する確率が高くなるためである．ただし，本シミュレーション環境においては，有効半径が 500m の場合でも，90% 以上の正答率を維持している．

本シミュレーション結果により，上記条件のもとでは仮想センサを生成することで，高い正答率を維持しつつトラヒックの発生を大きく抑制できることを確認した．したがって，提案する分散センサ情報管理アーキテクチャおよび仮想センサを用いたセンサデータのキャッシュ手法により，正しいコンテキストを高い確率で導出しつつ，コンテキストアウェアサービスにおいてセンサデータからコンテキストを導出するための処理やトラヒックを抑制できると考えられる．

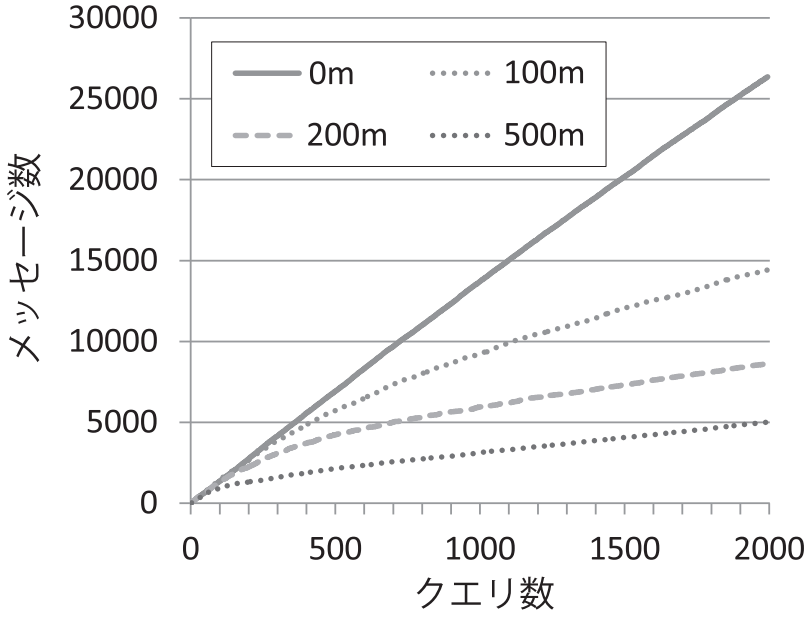


図 3.9: 有効範囲の大きさによるメッセージ数の変化

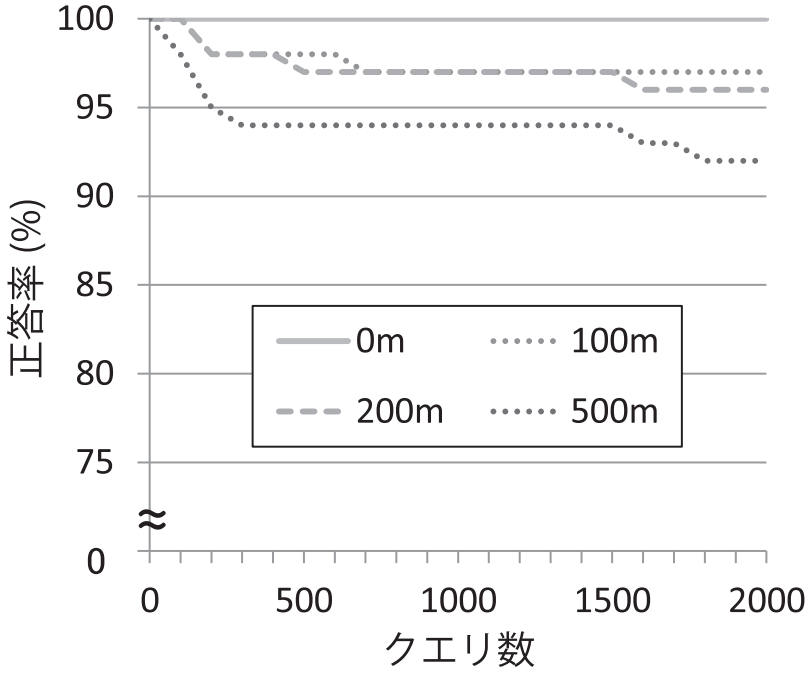


図 3.10: 有効範囲の大きさによる正答率の変化

3.4 むすび

本章では，大規模コンテキストアウェアサービスに対して効果的に状況情報を提供できる分散センサ情報管理アーキテクチャを提案した．提案アーキテクチャでは，必要なセンサ情報を取得できない問題に対して，周辺に存在する異種のセンサ情報から必要な情報を導出する手法を提案した．情報の導出は，Raw Layer（RL），Abstract Layer（AL），Logical Layer（LL）の 3 層でセンサ情報を階層的に管理し，センサデータを LL 層で意味的に解釈することで行われる．また，同様の要求に対してもセンサデータ収集やコンテキスト導出などの冗長な処理を行う問題に対しては，要求に対する処理結果を共有するための仮想センサを設置し，過去の結果を再利用することで処理を抑制する手法を提案した．提案するセンサ情報分散管理アーキテクチャを実装し，仮想センサによって負荷を削減する手法の有効性をシミュレーションにより評価した．本シミュレーション結果により，高い正答率を維持しつつトラヒックの発生を大きく抑制できることを確認した．提案するセンサ情報分散管理アーキテクチャおよび仮想センサ設置手法により，コンテンキストアウェアサービスにおいてセンサデータからコンテキストを導出するための処理やトラヒックを抑制できると考えられる．

第4章

Rete アルゴリズムを用いた地理的な情報に基づくコンテンツ配信手法

4.1 まえがき

第 3 章では，センサデータをルールに基づいて意味的に解釈し提供する分散センサ情報管理アーキテクチャを提案した．これにより，ユーザやサービスからの要求に対して任意地点のコンテキストを提供することが可能である．しかし，得られる実世界のコンテキストに応じてコンテンツを配信するような大規模コンテキストアウェアサービスを提案アーキテクチャを用いて構築する場合，配信するコンテンツと実世界のコンテキストとのマッチング，つまり，配信するコンテンツが実世界のコンテキストにおいて適切なコンテンツであるかどうかの判別処理は，構築するサービス側で実現しなければならない．一般に，センサから得られる実世界のコンテキストやユーザコンテキストに応じてコンテンツを配信するようなサービスは大規模コンテキストアウェアサービスの典型例であり，このようなサービスの構築をより容易にするためには，実世界のコンテキストやユーザコンテキストと配信するコンテンツとのマッチングを実現するような基盤技術の開発が望まれる．

このようなコンテキストとコンテンツのマッチング処理もまた，IF-THEN 形式のルールで表現することが可能である．したがって，IF-THEN 形式で与えられたルールを効率的に処理できるような基盤技術が必要となる．これまで，IF-THEN ルールに基づく推論を行うプロダクションシステム [39] が開発されており，その推論エンジンにおけるルールの効率的な処理手法として，Rete アルゴリズム [11] が提案されている．また，Rete アル

ゴリズムを分散環境での Publish/Subscribe 型システム [10] に適用しようという既存研究もある [58].

しかし，大規模コンテキストアウェアサービスで使用されるルールには，コンテンツの位置情報とセンサの位置情報の近接性の条件や，ユーザが要求する地点あるいは地理的範囲とコンテンツの位置情報の包含性の条件など，地理的な条件照合が含まれることが多く，Rete アルゴリズムを単純に適用するだけでは不十分である．したがって，地理的な条件照合も含め，効率的にルールを処理できる基盤技術が必要である．

そこで本章では，コンテンツやセンサデータを地理的な情報に基づいて分散管理しつつ，IF-THEN 形式のルールに従ってコンテキスト導出や有効なコンテンツの配信を行う手法を提案する．提案手法では領域全体を地理的なブロックに分割し，ルールを Rete アルゴリズムに基づいて処理する際，対象とするブロック単位で処理を P2P ネットワーク上で分散させる．さらに，Rete アルゴリズムにおける処理を中間結果も含めてすべて分散管理することで，同様のクエリに対する処理を削減する．

以下，4.2 節ではルール処理のための既存技術について述べる．4.3 節では提案するコンテンツ配信手法について説明し，4.4 節でシミュレーションによる評価について述べる．最後に 4.5 節で本章をまとめる．

4.2 ルール処理のための既存手法および分散環境に適用した関連研究

本節では，データ群（事実群）をルール群に基づき処理するプロダクションシステムでの代表的なルール処理手法である Rete アルゴリズムについて解説する．また，Rete アルゴリズムを分散環境に適用した関連研究について述べる．

4.2.1 Rete アルゴリズム

プロダクションシステムは，“IF-THEN” 形式のルール群によって入力データ群を処理するシステムである [39]．プロダクションシステムでは，事実をルールの IF 部と照合し，すべての条件が満たされたルールの THEN 部を実行する．プロダクションシステムは，ルールを記憶するプロダクションメモリ（PM），ワーキングメモリエレメント（WME）と呼

ばれるデータを記憶するワーキングメモリ（WM），推論エンジンと呼ばれる制御プログラムから構成される．

プロダクションシステムでは条件照合の高速化に関するさまざまな研究が行われており，Forgy によって提案された Rete アルゴリズム [11] はその代表的な条件照合アルゴリズムである．Rete アルゴリズムは，Rete ネットワークと呼ばれるグラフ構造をルールに基づいて作成する．Rete ネットワークでは，入力データをルールの条件によってフィルタリングするノードを α ノード，β ノードと呼ぶ．α ノードは，1 つの親からの入力データを条件によってフィルタリングする．一方，β ノードは，2 つの親からの入力データを結合し，結合データを条件によってフィルタリングする．この β ノードにおける処理は “join” と呼ばれる．これら α ノード，β ノードで条件を満たしたデータは下流のノードへ送られる．Rete アルゴリズムは α ノードや β ノードでの条件照合において，条件を満たすデータあるいはデータの組合せを，それぞれ α メモリ，β メモリとして記憶しておき，データが追加されたり削除されたりした場合に必要となる条件照合処理を削減している．

この Rete アルゴリズムについて，図 4.1 に示したルール例を用いて説明する．このルールは,「『ユーザの要求領域に含まれる屋内のコンテンツ』もしくは『ユーザの要求領域と値が 0 の雨量センサの観測範囲に含まれる屋外のコンテンツ』をユーザに配信する」というルールである．このルールから生成される Rete ネットワークを図 4.2 に示す．ルールを満たすコンテンツの抽出は，図 4.2 の最上流に ルートノードで示したノードからユーザやコンテンツ，センサのデータを流すことで行われる．データはまず種類に応じて，下流の Object Type ノードと呼ばれる “user u”，“content c”，“sensordata s” へと振り分けて流される．同様に各ノードで示した条件を満たすデータのみをさらに下流のノードへ流すことで，最終的にルールを満たすデータの組みが図 4.2 の最下流にターミナルノードで示したノードに流される．図 4.2 のターミナルノードでは，左側に「ユーザの要求領域に含まれる屋内のコンテンツ」が，右側に「屋外かつ，ユーザの要求領域と値が 0 の雨量センサの観測範囲に含まれるコンテンツ」が流される．このとき，α ノード，β ノードには，処理結果がそれぞれ記憶されている．例えば,「屋内のコンテンツ」という条件を含むルールを新たに追加する場合，“c.status=INDOOR” で示した α ノードの下流にノードを追加することでこの α ノードがもつ α メモリの内容を取得でき，共通する条件に対する処理を抑制できる．

```
for user u, content c, sensordata s
if u.requestArea contains c.loc and
   (c.status = 'INDOOR' or
      (c.status = 'OUTDOOR' and
      s.type = 'RAINFALL' and
      s.observedArea contains c.loc and
      s.value = 0))
then
   deliver content c to user u
```

図 4.1: 擬似コードによるルールの例

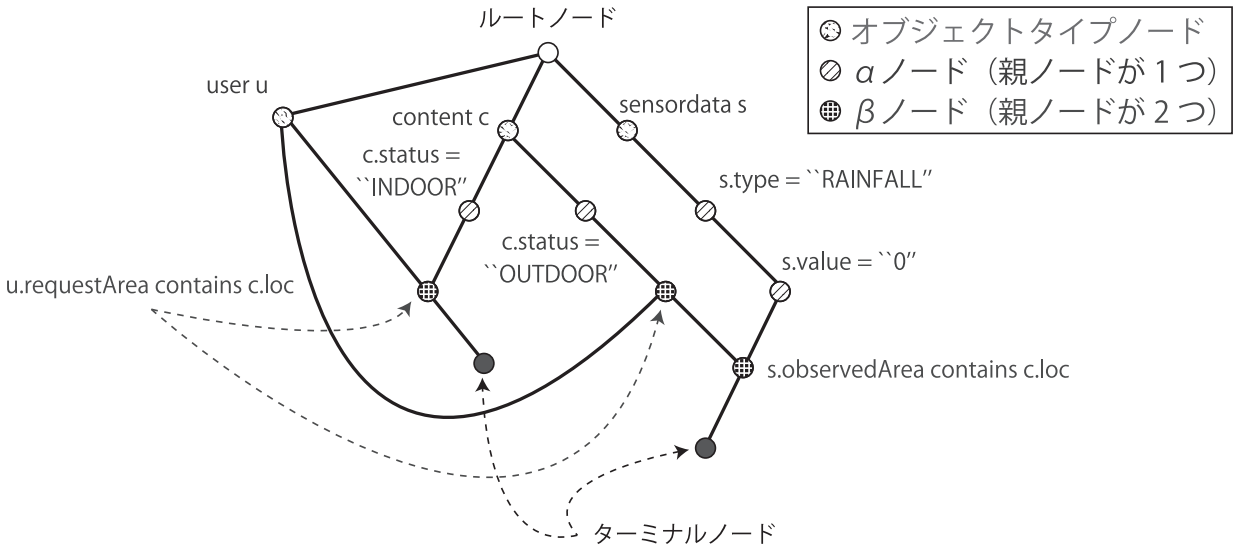


図 4.2: Rete ネットワークの例

Rete アルゴリズムは Drools[1] や JESS[2] など，多くのプロダクションシステム構築ツールやルールエンジンで用いられている．また，データ更新が頻繁に発生する環境を想定し，α メモリや β メモリを保持しない TREAT アルゴリズム [43] や，条件照合の計算コストが高いルールに対処するための手法 [38] が提案されている．また，条件照合を高速化するためのワーキングメモリのデータ構造も提案されている [45].

[1] http://www.jboss.org/drools/

[2] http://www.jessrules.com/

4.2.2 分散環境を想定した既存手法

Rete アルゴリズムを分散環境での Publish/Subscribe 型システム（Pub/Sub システム）[10] へ適用した手法として，DHT（Distributed Hash Table）の一種である Chord [59] を用いた手法が提案されている [58]．この手法では，発信するデータを（主語，述語，目的語）という 3-tuple で表現し，WME の主語，述語，目的語をそれぞれ Key として Chord で分散管理する．したがって，1 つの WME は Chord 上に 3 つ登録されることになる．このように WME を主語，述語，目的語のそれぞれに基づいて分散させることは，Rete ネットワークにおける α メモリを分散させることに相当する．Subscribe を行う購読者側は，ルールに基づき必要となる α メモリを担当するピアに購読要求を行う．つまり，ルール中の単純な照合条件に含まれる定数を Key とし，その Key を担当するピアに tuple template と呼ばれる照合条件を含む購読要求を行う．そのような照合条件はルール中に複数含まれることが一般的なため，1 つのルールに対して一般に複数のピアへの購読要求が必要となる．各 α メモリを担当するピアは購読者からの購読要求に対して，α メモリに存在する WME のうち登録された tuple template に適合する WME を購読者に送信する．購読者はこのようにしてルールの照合条件に適合する WME を購読先のピアから取得し，取得した WME を用いてルール処理を行う．特に，Rete ネットワークにおける β ノードの処理は，購読者側で行うこととなる．

この手法では，主語，述語，目的語を Key とするため，同じ Key をもつ WME は同じピアが担当することになり，多用される Key を担当するピアに負荷が集中するという問題がある．この問題に対し文献 [58] では，多用される Key を担当するピアの負荷を分散するため，WME が発行された日時を元に，それが発行された期間の情報を主語，述語，目的語それぞれに付加したものを Key として担当ピアを分散させる方法に言及している．しかし，この方法をとった場合，購読者側はさらに多くのピアに購読要求を行う必要が出てくるという問題が起こることが文献 [58] では述べられている．

また，文献 [58] は，多数の購読者が同一あるいは類似したルールをもとに購読要求を行うような環境は想定していない．このような環境を想定した場合，購読者間で共有できるのは α メモリだけであり，よりコストの高い処理，特に β ノードで行うべき処理が共有されず，非効率的である．

4.3 ルールおよび地理情報に基づく地理的コンテンツ配信手法

4.3.1 想定環境

本章では，センサデータを含む地理的コンテンツを複数のコンピュータ（ピア）から構成される P2P ネットワーク上で分散共有するとともに，ルールの処理をピア間で分担することにより，コンテキストアウェアサービスを構成するための手法を提案する．

コンテキストを導出するためのルール群は，あらかじめ作成されており，その処理を分担するすべてのピアで共有されているものとする．また，ルールの追加や削除は発生しないものとする．

まず，想定環境においてネットワークに接続するコンピュータを以下に定義する．

- ピア

  コンテキストアウェアサービスを実現するための情報の共有，処理を行うためのリソースとしてネットワーク接続された物理的コンピュータである．ピア間でオーバレイネットワークを構築し，情報の共有やルールに基づく導出処理を分散して行う．

- ユーザ端末

  ユーザが直接利用する端末であり，固定端末およびモバイル端末を想定する．ユーザが必要とするコンテンツの地理的な領域（例えば，ユーザの現在位置から半径 3km 以内など）が指定されるものとする．

- コンテンツ生成端末

  店舗に関する口コミ情報など，特定の位置に結び付いたコンテンツを生成し発信する端末である．また，生成コンテンツの管理を P2P ネットワーク上の担当ピアに依頼する．

- センサ

  設置された地域の気温や雨量などを計測し，センサデータを生成し発信する端末である．コンテンツと同様，センサデータの管理を P2P ネットワーク上の担当ピアに依頼する．

また，処理される地理的コンテンツやセンサデータを以下に定義する．

- 地理的コンテンツ

  ユーザに配信する対象となる情報である．本章では「位置 (x_1, y_1) にあるオープンカフェ」など，特定の位置情報を属性としてもつことを想定する．また，屋内（INDOOR），屋外（OUTDOOR）などの属性情報をもつ．例えば，建物内にあるレストランは屋内属性，オープンカフェは屋外属性となる．

- センサデータ

  各種のセンサが生成するデータであり，環境のコンテキストを導出するために用いる．センサの種類，センサが設置された場所を表す位置情報，計測されたセンサデータが有効となる地理的な範囲,「気温」「雨量」などの観測値を属性としてもつ．

4.3.2 地理的な情報に基づく分散処理

ルールの処理には 4.2.1 節で述べた Rete アルゴリズムを用いることで冗長な処理を削減できると考えられるが，それを多数のピアに分散処理させることを考えた場合，単純に Rete ネットワークの各ノードごとに担当ピアを割り当てると，Rete ネットワークのノード数のピアにしか処理を分担できない．また，Rete ネットワークでは各ノードに参照頻度の偏りがあり，特定の処理を担当するピアに負荷が集中する可能性がある．したがって，Rete ネットワークのノード単位ではなく，1 つのノードの処理をさらに細かく分割し複数のピアに担当させることが要求される．

想定環境においては，地理的コンテンツやセンサデータは位置情報をもつ．また，ユーザが指定する領域やセンサデータの有効範囲は地理的な範囲をもつ．IF-THEN ルールによるこれらのデータのマッチングには，地理的な包含関係，例えば「ユーザが要求する領域に含まれる地理的コンテンツ」や「センサデータの有効範囲に含まれる地理的コンテンツ」といった関係を条件とすることが多い．したがって，地理的近接性や包含関係という条件を効率的に処理できるような分散処理手法を考える必要がある．

そこで提案手法では，領域全体を複数のブロックに分割し，地理的な点や範囲をブロック単位でピアに分散管理させる．このように，ブロック単位で Rete ネットワークにおける各ノードの処理を分散させることにより，分割しない場合に比べて多くのピアで処理を分担できる．また，地理的近接性や包含関係の条件については，地理的コンテンツやセン

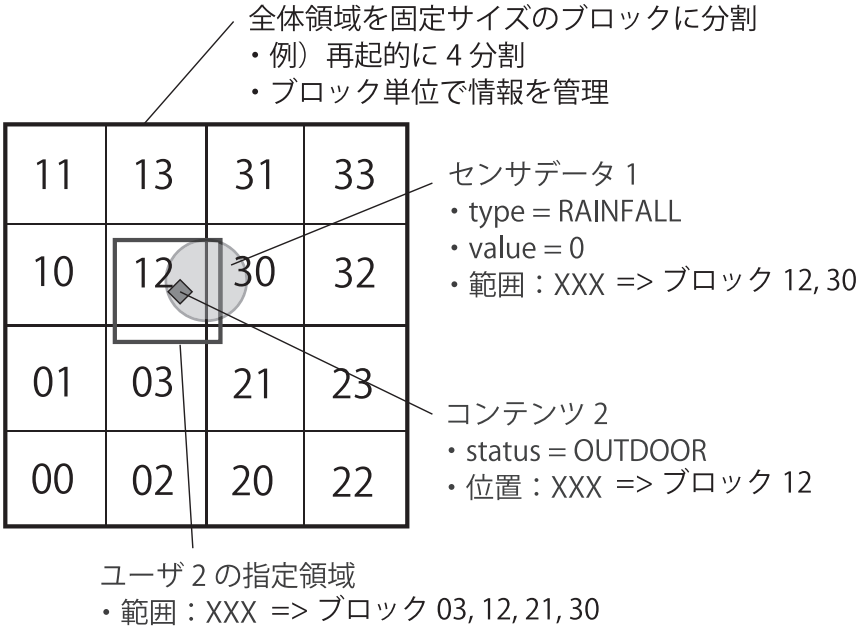


図 4.3: ブロックへの分割の例

サデータをブロック単位に分割することにより，それらの条件に適合しないようなデータ間のマッチング処理を大きく削減できる．

提案手法におけるブロックへの分割の例を図 4.3 に示す．図 4.3 の各ブロック中央の番号は各ブロックを識別するための ID（ブロック ID）である．図 4.3 の例では，コンテンツ 2 はブロック 12 内に存在するため，ブロック 12 を担当するピアにおいて処理される．一方，type が “RAINFALL” であるセンサデータ 1 の有効範囲はブロック 12，30 に重なっている．このように複数のブロックに重なるような領域をもつデータは，それらのすべてのブロックにおけるマッチング処理に必要となる可能性があるため，その領域に重なるブロックを担当する全てのピアにおいて処理されるようにそのデータを配信する．

地理的なブロックの分割数や大きさについては，ブロック数がピア数よりも多くなるようにさらに細分化することで，各ピアには担当すべき処理が細かく分担されることになる．また，複数のブロックの処理を 1 つのピアに担当させることにより，各ピアの負荷をより均等化することができる．ただし，図 4.3 のセンサデータ 1 の場合，ブロック 12，30 の担当ピアがそれぞれセンサデータ 1 の処理や結果の記憶を行うことになるが，さらにブロックサイズを小さくすると，さらに多くのピアがこのデータの処理や結果の記憶を行うことになってしまう．つまり，ブロックを細分化しすぎるとピア全体の処理やメモリ量が大き

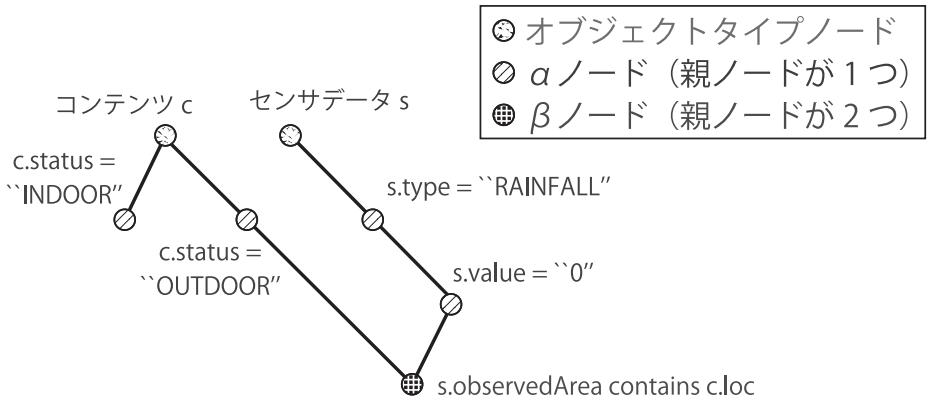


図 4.4: Rete ネットワークにおいて処理を分担させるノード

くなるため，負荷分散の度合いとのバランスを考え，ブロックの数や大きさを適切に設定する必要がある．適切なブロックの数や大きさは，コンテンツやセンサデータ，ピアの存在密度，有効範囲などによって決定できると考えられる．

提案手法では，Rete ネットワークの各ノードに ID を割り当て，上述のブロック ID と組み合わせることで Key とし，Key のハッシュ値に基づいて P2P ネットワーク上で担当ピアを割り当てる．ここで，処理を分担する Rete ネットワークのノードについて考えると，あるユーザのコンテンツ要求領域などのデータは，そのユーザ以外が参照する可能性は低く，P2P ネットワーク上で共有する必要性は低い．そのため，提案手法では，ユーザのコンテンツ要求領域と地理的コンテンツの位置情報の包含関係のマッチング処理は，Rete ネットワーク内でマッチング処理を行うかわりに，各ユーザ端末内で行うものとする．つまり，各ユーザ端末は，Rete ネットワークのターミナルノード（実際には，後述するようにターミナルノードの直近上位のノード）を担当するピアのうち，ユーザのコンテンツ要求領域に重なるブロックを担当するすべてのピアに接続し，それらのブロック内でルールに合致する地理的コンテンツの配信を要求することとする．

また，提案手法では Key に基づいて P2P ネットワーク上のピアを探索できるため，図 4.1 に示したルートノードおよびターミナルノードは不要である．そのため，ルートノードおよびターミナルノードはピアに担当させず，コンテンツ生成端末やセンサは，地理的コンテンツやセンサデータを対応する Object Type ノードに直接送信するものとする．また，ルール処理結果の配信を受けるユーザ端末は，ターミナルノードの直近上位のノードを担当するピアから配信を受けるものとする．したがって，図 4.1 のルール例に対しては，ピアに分担させる Rete ネットワークのノードは図 4.4 に示した各ノードになる．提案手法

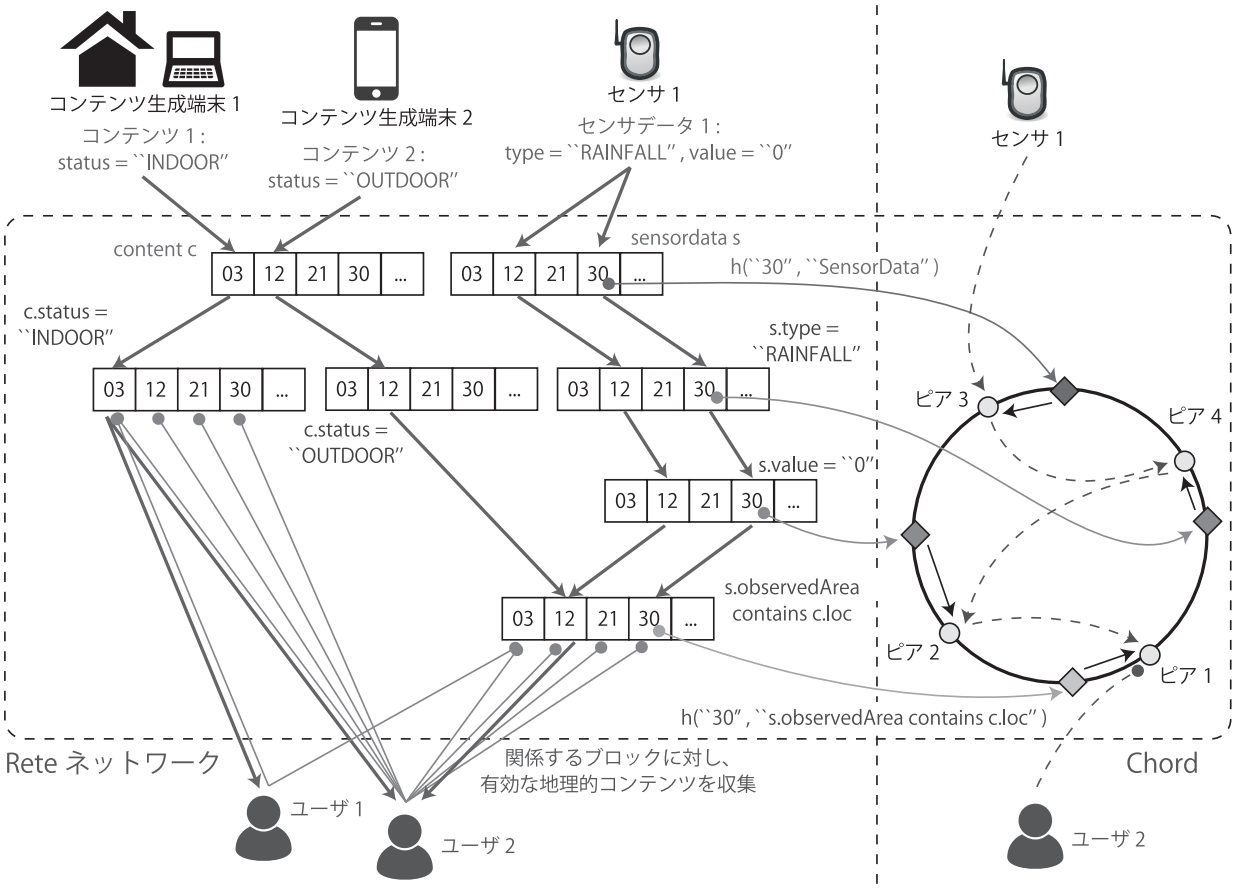


図 4.5: Rete ネットワークおよびブロックによる分散処理の例

では，図 4.4 のような Rete ネットワークのリンクを構築するために，各ピアは下流のノードおよびブロックの担当ピアをその Key に基づきオーバレイネットワーク上で探索し記憶する．また，文献 [58] の手法では α メモリだけが共有されていたが，提案手法では α メモリだけでなく β メモリも共有し，ユーザ端末は β ノードでフィルタリングされた結果を取得することで，不要なデータの送受信も抑制される．

図 4.4 の例に対して，提案手法で構築される Rete ネットワークの例を図 4.5 に示す．図 4.5 の例では，左側がルールから生成される Rete ネットワークを表している．複数のブロックに分割された “content c” や “sensordata s” などの各ノードは，図 4.4 の Rete ネットワークに対応している．図 4.5 の右側は，各ノードおよびブロックから計算されるハッシュ値の Chord における担当ピアへのマッピングの例を表している．

提案手法においては，Rete ネットワークのリンクはコンテンツもしくはセンサデータの登録にともなって構築する．図 4.5 のセンサ 1 の例ではまず，センサ 1 は自身の有効範囲

と重なる “sensordata s” ノードのブロック 12，30 に対して h(“30”, “sensordata s”) のようにハッシュ値をそれぞれ計算し，それらのハッシュ値を用いてオーバレイネットワーク上で担当ピアを探索する．発見されたブロック 30 の担当ピアは，センサデータが登録されると下流の “s.type = RAINFALL” ノードの同じブロック 30 に対して，同様に h(“30”, “s.type=RAINFALL”) のようにハッシュ値を計算し，そのハッシュ値の担当ピアをさらに探索し，リンクとして記録する．この処理を Rete ネットワークの末端ノードまでブロックごとに繰り返し，他のセンサデータやコンテンツでも同様の処理を行うことで，必要な Rete ネットワークのリンクが構築される．図 4.5 のセンサ 1 が含むブロック 30 の場合，以下のハッシュ値の担当ピアへ順にデータが流れる．

1. h(“30”, “sensordata s”)
2. h(“30”, “s.type=RAINFALL”)
3. h(“30”, “s.value=0”)
4. h(“30”, “s.observedArea contains c.loc”)

ユーザがコンテンツの配信を依頼する場合は，図 4.3 のユーザ 2 の場合，ユーザ 2 の指定領域に重なるブロック 03，12，21，30 の 4 つに関して，コンテンツの配信を依頼する．図 4.5 では依頼する対象は “c.status = INDOOR” のノードと “s.observedArea contains c.loc” のノードであり，あらかじめ知っている Rete ネットワークのノード ID およびブロック ID のハッシュ値によって対象となるブロックを担当するピアを探索し，発見されたピアにコンテンツの配信を依頼する．具体的には以下の 8 つのハッシュ値の担当ピアを探索する．

- h(“03”, “c.status=INDOOR”)
- h(“12”, “c.status=INDOOR”)
- h(“21”, “c.status=INDOOR”)
- h(“30”, “c.status=INDOOR”)
- h(“03”, “s.observedArea contains c.loc”)
- h(“12”, “s.observedArea contains c.loc”)

- h(“21”, “s.observedArea contains c.loc”)
- h(“30”, “s.observedArea contains c.loc”)

依頼されたピアは，自身まで流れてきたコンテンツのうち，条件に適合するコンテンツをユーザ 2 へ配信する．ユーザの指定領域が変化した場合には，コンテンツやセンサデータの配信依頼ブロックを逐次変更する．

4.4 評価

4.4.1 評価環境

4.3 節の提案手法の有効性を検証するため，シミュレーション評価を行った．本シミュレーションのシナリオおよびルールについては，図 4.2 の例を用いた．

シミュレーション環境を表 4.1 に示す．また，ユーザの指定領域やコンテンツ，センサの配置の例を図 4.6 に示す．図 4.6 では全体領域が 32km × 32km で，円がユーザの指定領域

表 4.1: 評価環境

全体領域	32km × 32km
ユーザの指定領域	半径 4km の円
コンテンツ数	512
コンテンツの位置	ランダム
コンテンツの種類の割合	屋内: 50%, 屋外: 50%
センサデータ数	512
センサデータの位置	ランダム
センサデータの有効範囲	半径 1.5km の円
センサデータの値の割合	0ml: 50%, 10ml: 50%
ピア数	32, 64, 96, 128
オーバレイネットワーク	Chord
ブロックの一辺の長さ	16km, 8km, 4km, 2km

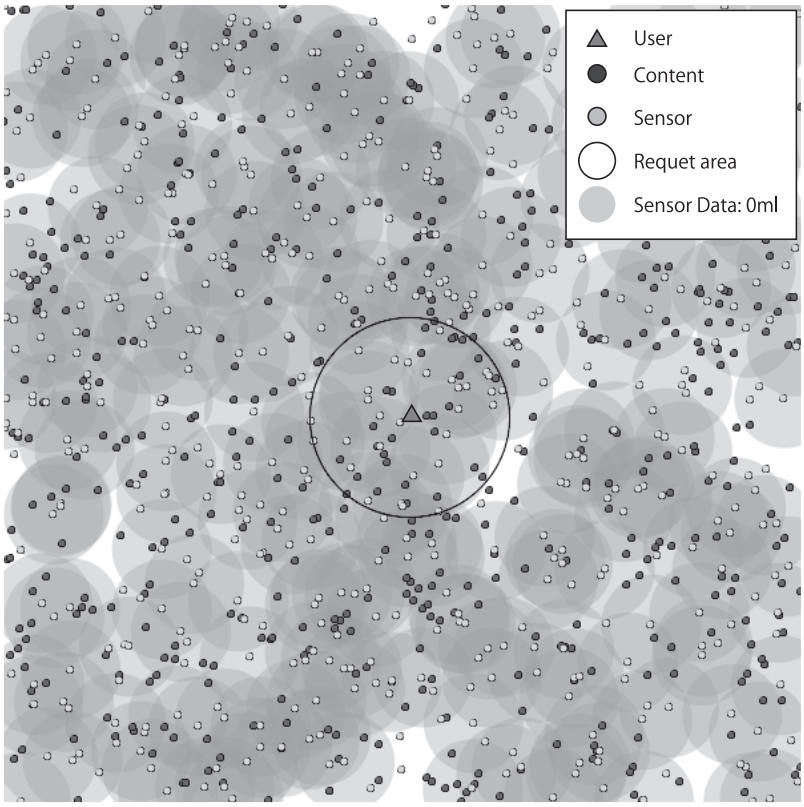


図 4.6: シミュレーションにおける配置の例

を，薄い点が雨量センサを，濃い点がコンテンツを表している．コンテンツおよび雨量センサは密度を 1km^2 あたり 0.5 個とし，場所は各試行でランダムに配置した．コンテンツには「屋内」「屋外」の 2 種類があり，割合は 50% ずつに設定した．また，雨量センサは半径 2km を計測値の有効範囲とした．図 4.6 の明るい範囲は「晴れ」である 0ml を，それ以外の雨量センサは「雨」である 10ml を観測値としており，本シミュレーションでは割合を 50% ずつに設定した．ユーザがコンテンツを要求する領域は中規模の都市程度の半径 4km とし，本シミュレーションでは 1 ユーザがランダムな場所のコンテンツを要求するとした．ピアを探索するためのオーバレイネットワークには Chord を用い，ピア数は 32，64，96，128 の場合で Chord における Key の担当範囲が偏らないように配置した．また，提案手法のブロックへの分割については，分割しない場合と一辺の長さが 16km，8km，4km，2km の固定サイズのブロックに分割した場合を行った．このとき，全体領域のブロック数はそれぞれ，1，4，16，64，256 となる．

本シミュレーションでは，評価項目として各ピアの通信負荷を比較した．実世界では受信と送信による負荷が等しいとは限らないが，本シミュレーションでは各ピアの通信負荷

はデータの受信数および送信数の和で定義した．負荷の対象とするデータについては，ピアが処理するコンテンツやセンサデータとした．

各試行では，まずコンテンツやセンサデータを発生させ，Rete アルゴリズムに基づいて各ピアで処理を行う．その後，ユーザがコンテンツ配信のクエリを担当ピアへ送信し，該当するデータを受信する．試行はそれぞれの環境で 50 回ずつ行い，シミュレーションの結果として負荷の平均値を算出した．

4.4.2 処理結果の共有における評価

本シミュレーションでは Rete アルゴリズムによる処理結果の共有の有効性を評価するため，コンテンツおよびセンサデータの数や位置を固定した状態でユーザ数を変化させた．本試行では，シミュレーション開始時にコンテンツおよびセンサデータを発生させ，Rete ネットワークを構築する．その後，ユーザを順次追加してクエリを送信し，ネットワーク全体におけるピアの通信負荷を 5 ユーザごとに比較した．各ユーザについては，半径 4km の円のランダムな領域を指定するものとした．ブロックへの分割は一辺の長さが 4km で，ピア数は 128 で行った．

比較手法については，Rete アルゴリズムの α メモリおよび β メモリを共有する提案手法に対し，「α メモリのみ共有する場合」「α メモリおよび β メモリを共有しない場合」と比較した．α メモリおよび β メモリを共有しない場合は図 4.2 の “content c” および “sensordata s” ノードにコンテンツやセンサデータを集約し，ブロック単位で担当ピアに割り当てる．この場合，各ピアの通信負荷は，Rete アルゴリズムを用いず，単純に指定領域に関連する地理的コンテンツやセンサデータがユーザのクエリのたびに収集される場合と同等となる．また，α メモリのみ共有する場合とは，文献 [58] で提案されている手法を適用した場合である．

ユーザ数を 0 から 20 まで追加した場合の通信負荷の累計を図 4.7 に示す．図 4.7 の横軸はユーザ数の累計で，縦軸は通信負荷の累計である．図 4.7 より，まずコンテンツおよびセンサデータが発生した時点では，Rete ネットワーク内をコンテンツやセンサデータが流れる．α ノードや β ノードのブロックを担当するピアでデータが送受信されるため，α メモリおよび β メモリを共有する場合が最も通信負荷が大きい．一方，α メモリおよび β メモリを共有しない場合は，通信負荷は “content c” もしくは “sensordata s” ノードのブロックを担当するピアが受信するデータ数のみとなる．しかし，ユーザを順次追加し，各

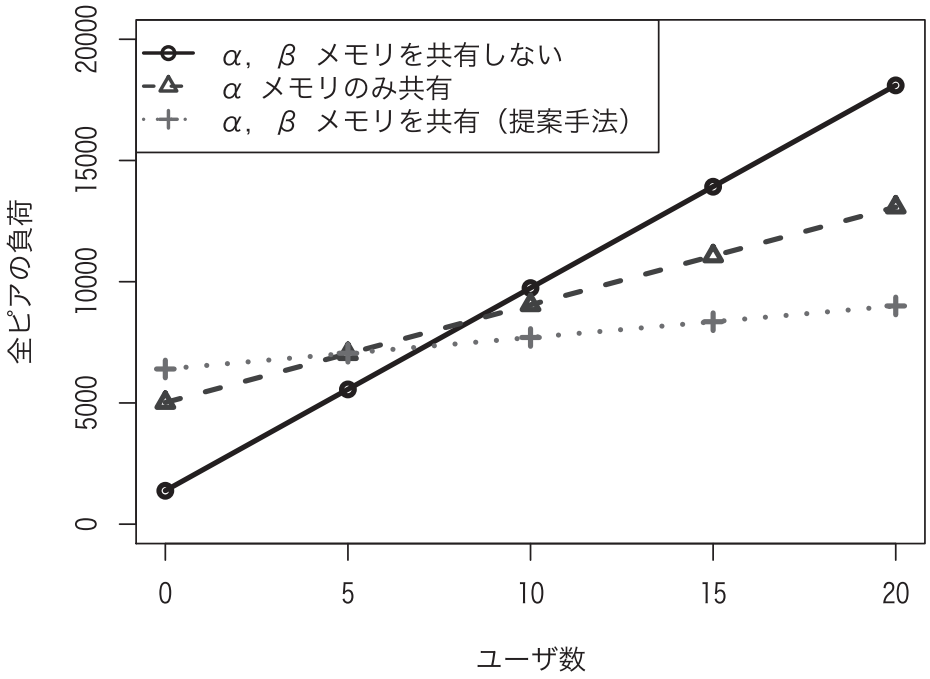


図 4.7: ユーザ数による全体の通信負荷（128 ピア，ブロックの一辺 4km）

ユーザが領域を指定して地理的コンテンツの配信を依頼するにしたがって，累積の通信負荷は α メモリおよび β メモリを共有しない場合が大きく増加する．コンテンツおよびセンサデータの数がそれぞれ 512 である本シミュレーションでは，α メモリおよび β メモリを共有する提案手法は，α メモリのみを共有する場合，α メモリおよび β メモリを共有しない場合ともに，ユーザ数が 10 の時点で累積の通信負荷が下回る．

それぞれの直線の傾きを示す要素については，まず α メモリおよび β メモリを共有する場合は，通信負荷はユーザと α ノードもしくは β ノード間のデータ送受信のみである．一方，α メモリのみを共有する場合は各ユーザが α ノードからデータを収集し，α メモリおよび β メモリを共有しない場合は “Content” および “SensorData” ノードから収集する．データの収集は，地理的に関連するセンサデータやルールによって最終的に無効と判断されるコンテンツに対しても行われる．特に，各ユーザが個別にセンサデータを収集する冗長な通信が全体の通信負荷に大きく影響している．

図 4.7 より，α メモリおよび β メモリを共有する提案手法は，ユーザやクエリの数の増加による影響が低く，スケーラビリティが高いといえる．しかし，本シミュレーションでは，コンテンツおよびセンサデータは最初に発生させた後，追加や更新などの変化はない．

α メモリおよび β メモリを共有する提案手法では，ユーザの増加による影響が低い一方，コンテンツやセンサデータの更新や追加によって，Rete ネットワークの各ノードのメモリに変更を反映するためのメッセージが流れる．Rete アルゴリズムは不要なノードへ変更のメッセージは流れないように設計されているが，文献 [38, 43] でも述べられているのと同様，更新頻度が高い環境では通信負荷が高くなると考えられる．

4.4.3 地理的な情報に基づく分散処理における評価

Chord のピア数が 128 の場合において，ブロックの大きさによる通信負荷を比較した結果を図 4.8 に示す．図 4.8 の横軸はブロックの一辺の長さである．縦軸は全ピアの通信負荷の合計を示しており，コンテンツやセンサデータの送受信ごとにそれぞれ示している．コンテンツおよびセンサデータに関係する送受信の各負荷には Rete ネットワークのリンクを構築する際のメッセージが含まれる．

図 4.8 より，ブロックの一辺の長さが半分，つまりブロック数が 4 倍になると，小さなブロック単位で扱うことで不要な地域へのクエリは削減されるが，ネットワーク全体の通信負荷は指数的に増加する．これは，センサデータの有効範囲が複数のブロックを含み，ブロックが小さくなると有効範囲に含まれるブロック数は大きくなるためである．一方，コンテンツ自体はブロックの大きさに関わらず関係するのは 1 ブロックのみである．しかし，初めに Rete ネットワークのリンクを同一ブロック内で構築し，その後は 1 ホップでメッセージを送信しているため，ブロックが細かくなると担当ピアを新たに発見するための通信が増加する．

各ピアの通信負荷の詳細について，ピア数が 32，64，96，128 の結果をそれぞれ図 4.9，4.10，4.11，4.12 に示す．図 4.9 から図 4.12 では，ピア数が 32 の場合のみ，測定値をプロットしている．各図の横軸はピアの ID を，縦軸は各ピアの通信負荷を示している．ピアの ID は負荷の小さいものから順番に並べており，右端のピアが通信負荷の最も高いピアを表している．

これらの結果より，地理的なブロックに分割しない場合は少数のピアしか Rete ネットワークのノードを担当しないため，コンテンツやセンサデータの配信，処理結果の管理などが特定のピアに集中する．そのため，いずれのピア数でも，ブロックに分割しない場合は通信負荷が上位 7 ピアにほぼ集中している．一方，ブロックに分割する場合，一辺が 16km では上位 28 ピアにほぼ集中しているが，8km 以下ではすべてのピアに通信負荷が分散さ

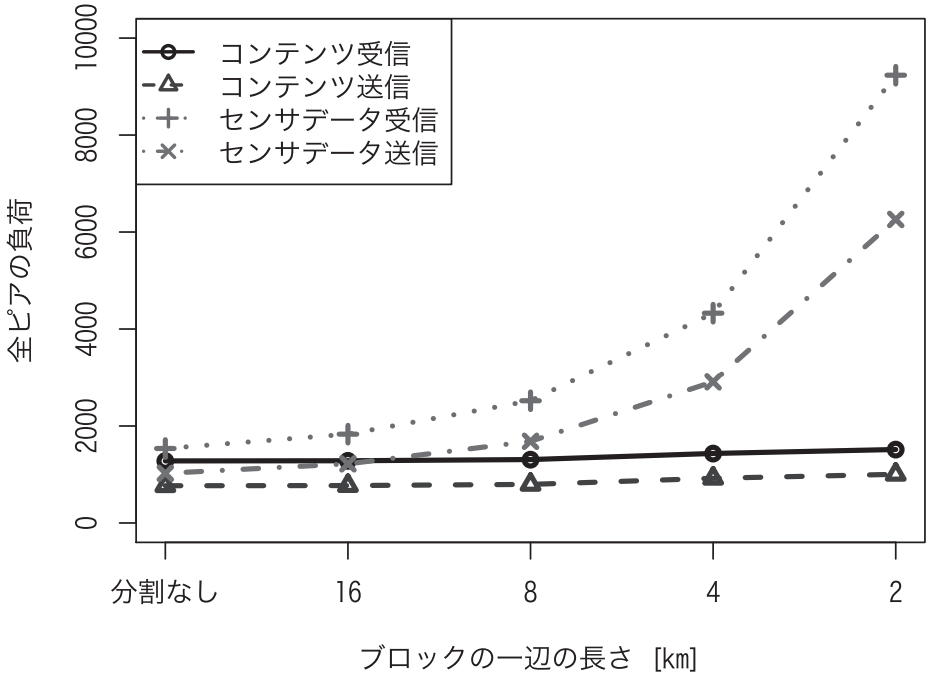


図 4.8: ブロックの大きさによる全体の通信負荷（128 ピア）

れている．通信負荷はブロックが細かくなるほど分散され，上位のピアの負荷も低下する．図 4.8 のようにブロックの分割数を大きくすると通信負荷が大きくなるため，分散による 1 ピアあたりの通信負荷の減少とのバランスを考慮し，ブロックの大きさを決める必要がある．ただし，ピア数が多くなると 1 ピアあたりの通信負荷も小さくなるため，ブロックの細分化によるオーバヘッドの影響は小さくなる．

各ピア数とブロックの大きさにおいて，通信負荷が最大となるピアの値を図 4.13 に示す．ブロックに分割しない場合は 1 ピアに最大で約 1,000 の通信負荷が集中するが，ブロックに分割することで，ピア数が 32 の環境ではブロックの一辺が 16km の場合で約 400 に，64，96，128 ピアの環境ではブロックの一辺が 8km の場合で約 200 に，それぞれ最低値として抑えられている．つまり，最も処理が集中してボトルネックとなるピアの通信負荷が最低になる点が，各ピア数の環境における最適なブロックの大きさといえる．また，図 4.9 などと同様，ピア数が多いほど，ブロックを細かく分割した際の最大負荷は減少している．

ピア数による通信負荷の比較として，各ピア数とブロックの大きさにおいて，1 ピアあたりの平均負荷を算出した結果を図 4.14 に示す．図 4.13 と同様，ピア数が少ないと 1 つのピアが多くのブロックを担当するため，1 ピアあたりの負荷が大きい．1 ピアあたりの

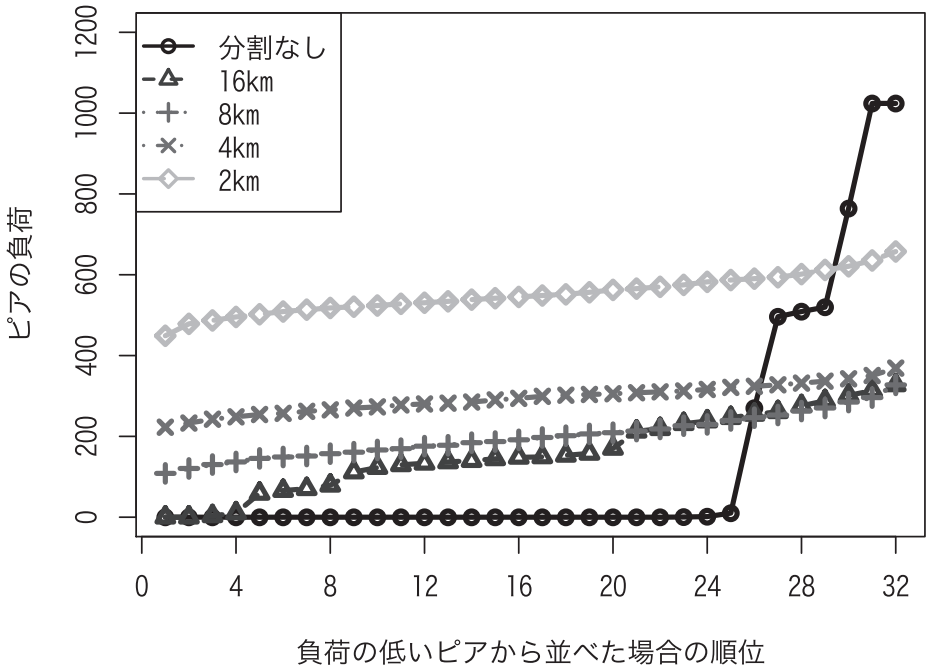


図 4.9: 32 ピアにおける通信負荷の内訳

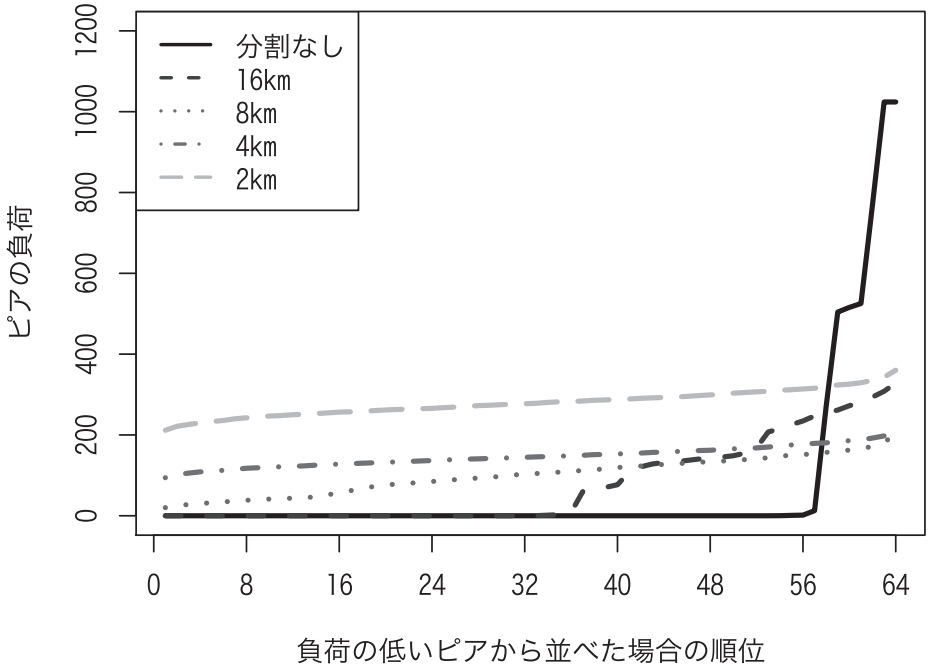


図 4.10: 64 ピアにおける通信負荷の内訳

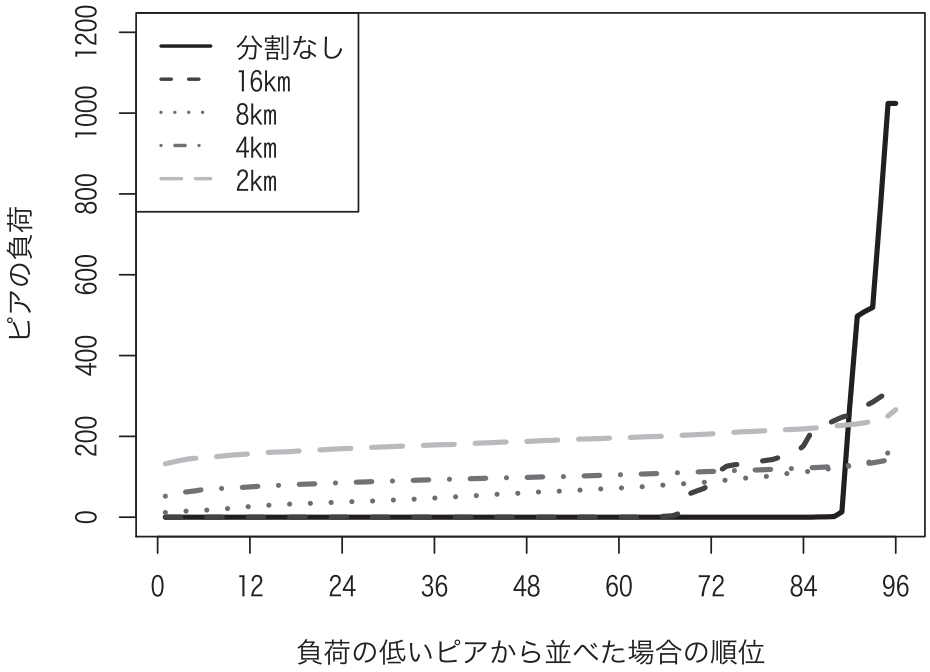


図 4.11: 96 ピアにおける通信負荷の内訳

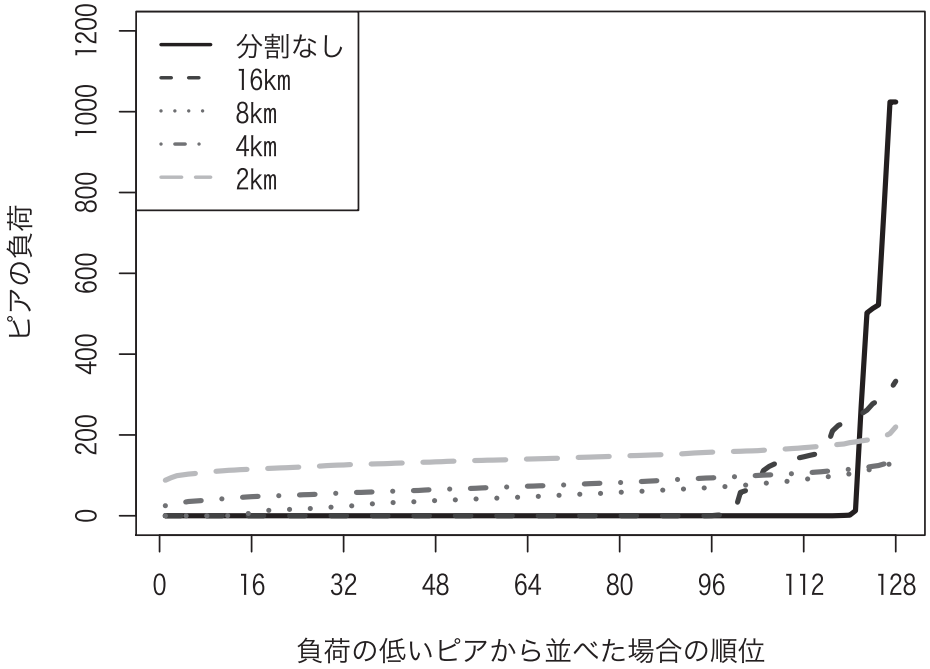


図 4.12: 128 ピアにおける通信負荷の内訳

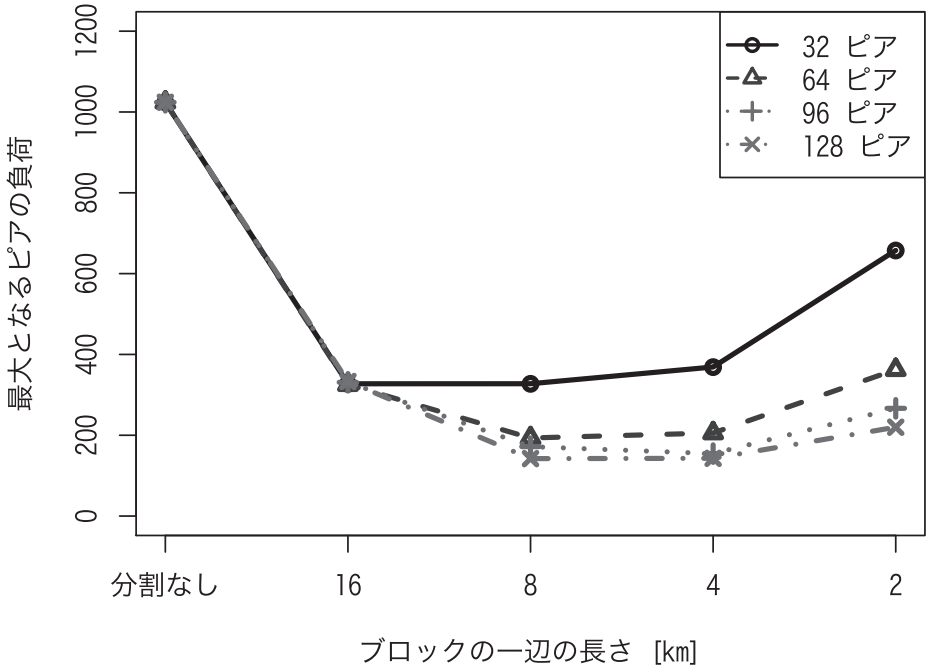


図 4.13: ブロックの大きさによる最大となるピアの負荷

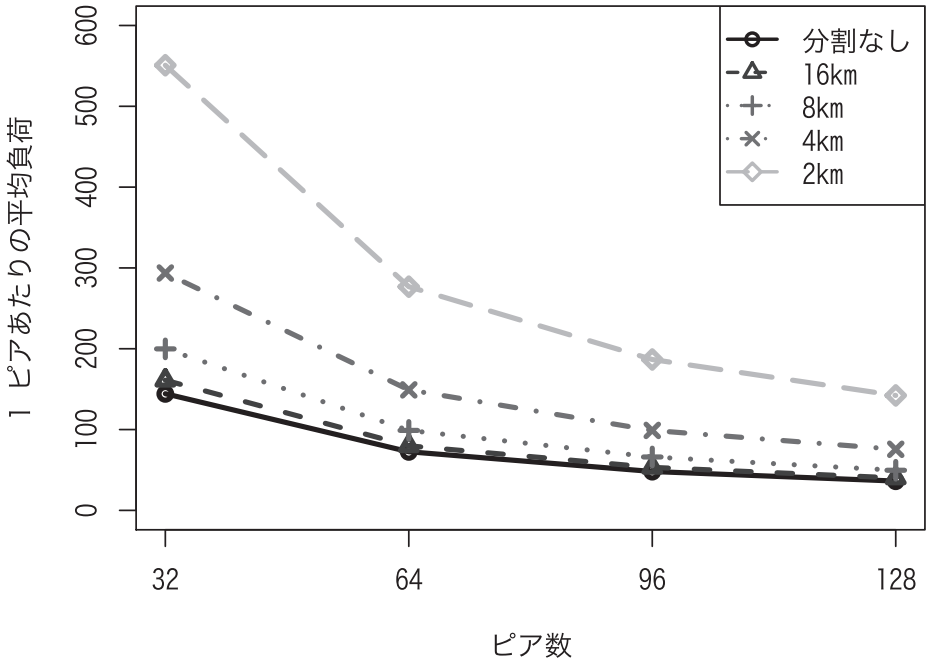


図 4.14: ピア数による 1 ピアあたりの平均負荷

平均負荷はピア数にほぼ反比例する．また，ブロックの大きさが小さいほど，ピア数の増加に対する 1 ピアあたりの平均負荷の低下の割合は大きい．ただし，ブロックの大きさを小さくすると全体の通信負荷は増加しているため，図 4.13 と同様，最適となるブロックの大きさがある．以上の結果から，ブロックを細かくすることによるオーバヘッドの増加と負荷分散による 1 ピアあたりの負荷の減少のバランスを考え，ブロックの大きさを決める必要があることがわかる．本シミュレーション環境では，以上のシミュレーション結果から総合的に判断すると，ブロックの大きさは 8km（16 ブロック）が最適であると考えられる．この最適なブロックの大きさは，ピア数が少なくなると大きくなり，多くなると小さくなる．

4.4.4 考察

まず提案手法の地理的なブロックへの分割については，図 4.8 より，過剰に分割すると通信負荷が大きく増加し，特にユーザの指定領域やセンサの有効範囲などの複数ブロックに関係する場合に影響が大きい．これは，センサの有効範囲が複数ブロックに跨がる場合に，Rete ネットワークの最上位の Object Type ノードにおいてそれらの複数ブロックを担当するピアにセンサデータを複製して送信していることも一つの原因である．実際には，最上位の Object Type ノードに対して複数ブロックにセンサデータを送信する必要はなく，センサデータとコンテンツの位置に基づく条件照合を行う β ノードに対して上位ノードからセンサデータを送信する際に，関係する複数ブロックにそれを複製して送信するだけでよいため，そのような改善方式について検討することが必要である．

4.5 むすび

本章では，コンテンツやセンサデータを地理的な情報に基づいて分散管理しつつ，IF-THEN 形式のルールに従ってコンテキスト導出や有効なコンテンツの配信を行う手法を提案した．提案手法では領域全体を地理的なブロックに分割し，ルールを Rete アルゴリズムに基づいて処理する際，対象とするブロック単位で処理を P2P ネットワーク上で分散させる．さらに，Rete アルゴリズムにおける処理を中間結果も含めてすべて共有することで，不要なデータの収集を抑制する．シミュレーションにより提案手法を評価し，地理的なブロック単位で多くのピアに処理を分散でき，処理結果を共有することで全体の通信負荷を削減できることを確認した．

第5章

結論

5.1 本論文のまとめ

本論文では，ユビキタス環境において大規模コンテキストアウェアサービスを構築するための 3 つの基盤技術に対し，それぞれ P2P アーキテクチャに基づき実現する手法について議論した．

まず，第 1 章では，ユビキタス環境における大規模コンテキストアウェアサービスの重要性について述べ，構築基盤として要求される基盤技術について述べた．

第 2 章では，位置情報に関連する地理的オーバレイネットワークを独立して構築し，さまざまな地理的オーバレイネットワークに適用可能な地理的コンテンツ分散管理手法を提案した．既知のピアやコンテンツの配置，分布密度などからコンテンツ担当ピアの存在確率を計算し，その確率とボロノイ領域で表されるコンテンツ担当領域によりピア探索範囲を決定する地理的コンテンツ検索手法を提案した．特に，不要な範囲のピア探索を抑制する手法として，ピアや地理的コンテンツの情報を収集しながら反復的に探索を行う手法を提案した．地理的コンテンツの検索手法の有効性の評価として，シミュレーションを行った．シミュレーション結果により，適切な範囲のみのピアを探索することで，ピア数が多い環境や分布に偏りがある環境においても，トラヒックの発生を抑制しつつ，必要とするコンテンツを取得できることを示した．

第 3 章では，分散環境においてコンテキストアウェアサービスを実現するための分散センサ情報管理アーキテクチャを提案した．提案アーキテクチャでは，センサが存在しないために必要なセンサ情報を取得できない問題に対して，周辺に存在する異種のセンサ情報

から必要な情報を導出する手法を提案した．また，同様の要求に対して冗長な処理が発生する問題に対して，処理結果を共有するための仮想センサを設置し，過去の結果を再利用することで処理を抑制する手法を提案した．さらに，提案するセンサ情報分散管理アーキテクチャの実装について述べ，仮想センサの有効性をシミュレーションにより評価した．シミュレーション結果により，高い正答率を維持しつつ，トラヒックの発生を抑制できることを示した．

第 4 章では，コンテンツやセンサデータを地理的な情報に基づいて分散管理しつつ，IF-THEN 形式のルールに従ってコンテキスト導出や有効なコンテンツの配信を行う手法を提案した．提案手法では領域全体を地理的なブロックに分割し，ルールを Rete アルゴリズムに基づいて処理する際，対象とするブロック単位で処理を P2P ネットワーク上で分散させる．さらに，Rete アルゴリズムにおける処理を中間結果も含めてすべて共有することで，不要なデータの収集を抑制する．シミュレーション評価により，提案手法は地理的なブロック単位で多くのピアに処理を分散でき，処理結果を共有することでトラヒックの発生を抑制できることを確認した．

本研究の成果によって，膨大なユーザや地理的コンテンツ，センサデータが存在するユビキタス環境において，P2P アーキテクチャに基づいた大規模コンテキストアウェアサービスを構築できるようになった．本論文では具体的なサービスの一例として主にコンテンツ配信を想定したが，他のさまざまなサービスにおいても，第 2 章の地理的コンテンツやセンサデータの収集，第 3 章のセンサデータを意味的に解釈したコンテキスト導出，第 4 章のコンテキストに基づいた処理の効率化の基盤技術は有効である．

5.2 今後の研究課題

本論文で述べた大規模コンテキストアウェアサービスは，地理的オーバレイネットワークを構築した多数のコンピュータ群による分散処理システムとして実現することを想定している．想定する分散処理システムでは，広域かつ膨大な数の地理的コンテンツやセンサデータを管理し，コンテキストアウェアサービスを要求するユーザに対して処理を行う．処理するユーザやデータが大規模化するにともない，分散処理システム自体も拡張されることが予想される．分散処理システム自体が拡張されることで，システムを構成するコンピュータが頻繁に参加や離脱することが考えられるが，本論文ではシステムを構築するコンピュータが追加や脱退などの動的に変化する場合や耐障害性について議論していない．た

だし，分散処理システムが構築する地理的オーバレイネットワークについては，ピアの追加や脱退，耐障害性などを考慮した構築手法が研究されている．本論文で提案した大規模コンテキストアウェアサービスのための基盤技術は，範囲探索が可能な地理的オーバレイネットワークならば適用できる．そのため，そのような耐障害性を考慮した地理的オーバレイネットワークにおける検索手法や推論手法の検討は今後の課題である．

観測データが連続的に流れるセンサデータに関しては，センサデータストリーミング配信に対する注目が高まっている．センサデータストリーム配信では，配信先は自身の処理能力や用途によって異なる周期で収集することが考えられる．我々のグループでは，複数の配信先が異なる周期で収集する環境を想定し，配信元の負荷を分散する手法を提案している [22, 27, 28, 37]．このようなセンサデータデータストリーム配信手法を用いることで，コンテキストアウェアサービスで重要となる，膨大な数のセンサからリアルタイム性の高いデータを収集する際の負荷を抑制できると考えられる．特に，こうしたデータストリーム配信手法と Rete アルゴリズムに基づく推論機構は相性が良いと考えられ，これらを連携動作させる推論機構の実現も今後の課題である．

また，本論文では，ルールの具体的な生成や管理の方法については扱っていない．広域かつ膨大な数のユーザやデータが存在する環境では，必要とされるルールも多種多様である．第 3 章および第 4 章で述べた手法で用いるルールに関して，ユーザやコンピュータがもつ独自のルールをどのように生成し管理するかについて，今後検討していく必要がある．本論文で述べた大規模コンテキストアウェアサービスでは，それらの多種多様なルールを用いることで，よりユーザに個別化された利便性の高いサービスを提供することが期待される．

本論文では，各提案手法の有効性はシミュレーションにより評価している．現実に大規模コンテキストアウェアサービスとして運用するには，システムの実装および実環境における評価が必要である．特に，大規模コンテキストアウェアサービスでは多種多様なユーザやコンテンツ，センサデータ，ルールが存在し，それらが動的に更新されるため，システムへ及ぼす影響を検証することは大規模コンテキストアウェアサービスの実現には不可欠である．

謝辞

本研究全般に関して，懇切なる御指導と惜しみない御助言を頂きました大阪大学サイバーメディアセンター 下條真司教授に謹んで御礼申し上げます.

本論文をまとめるにあたり，大変有益な御指導と御助言を多数賜りました大阪大学大学院情報科学研究科マルチメディア工学専攻 西尾章治郎教授，大阪大学大学院工学研究科 春本要准教授に心より感謝申し上げます.

本研究を推進するにあたり，直接の御指導，御助言，御討論を頂きました独立行政法人情報通信研究機構 寺西裕一氏に衷心より感謝申し上げます.

講義，学生生活を通じて，学問に取り組む姿勢をご教授頂きました大阪大学大学院情報科学研究科マルチメディア工学専攻 薦田憲久教授，藤原融教授，細田耕教授に厚く感謝申し上げます.

本研究において，多大なる御指導と御助言をご教授頂きました日本電信電話株式会社 竹内亨氏に厚く感謝申し上げます.

本研究において，共に研究を進め，多大なる御協力を頂きました今治市立今治小学校 三原慶彦氏，株式会社パシフィックビジネスコンサルティング Ly Lam Ngoc Bich 氏に深く感謝申し上げます.

本研究において，多大なる御助言，御協力，御支援を頂きました大阪大学サイバーメディアセンター 馬場健一准教授，伊達進准教授，義久智樹准教授，小島一秀講師，東田学助教，阿部洋丈助教，石芳正特任研究員，大阪大学大学院情報科学研究科マルチメディア工学専攻 原隆浩准教授，神崎映光助教，大阪大学大学院基礎工学研究科生体工学領域 野崎一徳特任講師，兵庫医療大学共通教育センター 加藤精一准教授，京都産業大学コンピュータ理工学部 秋山豊和准教授，株式会社ビービーアール 貫定秀典氏，吉田幹氏に深謝致します.

筆者の所属する研究グループにおいて，有益な御助言を頂いた大日本印刷株式会社 大谷隆三氏，株式会社博報堂 DY メディアパートナーズ 篠田裕之氏，独立行政法人情報通信研

究機構 山中広明氏，株式会社日立製作所 熊丸恵太氏，株式会社アイ・エム・ジェイ 小西佑治氏，田中博和氏，キヤノン株式会社 高橋健太郎氏，株式会社エヌ・ティ・ティ・データ 藤原謙太郎氏，小東浩一氏，株式会社デンソー 白木徹氏，ソニー株式会社 四之宮潤氏，大阪大学大学院情報科学研究科マルチメディア工学専攻 澤村雄介氏に深く御礼申し上げます．

本研究を進めるにあたり，多くの御討論や御助言を頂きました大阪大学大学院情報科学研究科マルチメディア工学専攻 下條研究室および西尾研究室の諸氏に心より感謝申し上げます．

最後に，研究生活を送る上で，暖かい御支援と多大なる御理解を頂いた両親を始めとする家族に心からの感謝と御礼を申し上げます．

参考文献

[1] F. Araujo and L. Rodrigues: GeoPeer: A Location-Aware Peer-to-Peer System, in *Proceedings of the 3rd IEEE International Symposium on Network Computing and Applications (NCA 2004)*, pp. 30–46 (Aug. 2004).

[2] J. Aspnes and G. Shah: Skip Graphs, *ACM Transactions on Algorithms (TALG)*, Vol. 3, No. 4 (37) (Nov. 2007).

[3] J. Bao, M. F. Mokbel, and C.-Y. Chow: GeoFeed: A Location Aware News Feed System, in *Proceedings of the 28th IEEE International Conference on Data Engineering (ICDE 2012)*, pp. 54–65 (Apr. 2012).

[4] D. Carboni, S. Sanna, and P. Zanarini: GeoPix: Image Retrieval on the Geo Web, from Camera Click to Mouse Click, in *Proceedings of the 8th International Conference on Mobile Human Computer Interaction (MobileHCI 2006)*, pp. 169–172 (Sept. 2006).

[5] G. Chen and D. Kotz: A Survey of Context-Aware Mobile Computing Research, *Technical Report TR2000-381, Dartmouth College* (Nov. 2000).

[6] H. Chen, T. Finin, and A. Joshi: An Ontology for Context-Aware Pervasive Computing Environments, *Knowledge Engineering Review*, Vol. 18, No. 3, pp. 197–207 (Sept. 2003).

[7] 長 健太, 岡本雄三, 山崎智弘, 岡本昌之, 服部正典, 大須賀昭彦: エージェントフレームワークを用いたコンテクストアウェアなテレマティクスサービスの構築, 電子情報通信学会論文誌 D-I, Vol. J88-D-I, No. 9, pp. 1448–1458 (Sept. 2005).

[8] A. Deshpande and S. Madden: MauveDB: Supporting Model-Based User Views in Database Systems, in *Proceedings of the 2006 ACM SIGMOD International Conference on Management of Data (SIGMOD 2006)*, pp. 73–84 (June 2006).

[9] A. K. Dey, T. Sohn, S. Streng, and J. Kodama: iCAP: Interactive Prototyping of Context-Aware Applications, in *Proceedings of the 4th International Conference on Pervasive Computing (Pervasive 2006)*, pp. 254–271 (May 2006).

[10] P. T. Eugster, P. A. Felber, R. Guerraoui, and A.-M. Kermarrec: The Many Faces of Publish/Subscribe, *ACM Computing Surveys*, Vol. 35, No. 2, pp. 114–131 (June 2003).

[11] C. L. Forgy: Rete: A Fast Algorithm for the Many Patterns/Many Objects Match, *Artificial Intelligence*, Vol. 19, No. 1, pp. 17–37 (Sept. 1982).

[12] 藤波香織, 中島達夫: コンテクストアウェアなアプリケーション構築のためのフレームワーク, 情報処理学会論文誌: コンピューティングシステム, Vol. 44, No. SIG 10(ACS 2), pp. 107–118 (July 2003).

[13] J. Gao and P. Steenkiste: An Adaptive Protocol for Efficient Support of Range Queries in DHT-Based Systems, in *Proceedings of the 12th IEEE International Conference on Network Protocols (ICNP 2004)*, pp. 239–250 (Oct. 2004).

[14] B. Guo, S. Satake, and M. Imai: Sixth-Sense: Context Reasoning for Potential Objects Detection in Smart Sensor Rich Environment, in *Proceedings of the IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT 2006)*, pp. 191–194 (Dec. 2006).

[15] 春本 要, 藤原謙太郎, 寺西裕一, 秋山豊和, 竹内 亨: 存在確率分布の伝播を用いた自己位置推定手法, 情報処理学会論文誌, Vol. 52, No. 5, pp. 1862–1870 (May 2011).

[16] 服部正典, 長 健太, 大須賀昭彦, 本位田真一, 深澤良彰: エージェントフレームワークを用いた車載端末向け情報提供システムの構築と評価, 情報処理学会論文誌, Vol. 44, No. 12, pp. 3024–3037 (Dec. 2003).

[17] 服部正典, 長 健太, 大須賀昭彦, 一色正男, 本位田真一: ユビキタス環境における Context-Aware なパーソナルエージェントの構築とその実証実験, 電子情報通信学会論文誌 D-I, Vol. J86-D-I, No. 8, pp. 543–552 (Aug. 2003).

[18] 広田 裕, 川島英之, 佐竹 聡, 梅澤 猛, 今井倫太: セマンティックセンサネットワークの実現に向けた実世界指向メタデータ管理システム MeT の設計と実装, 電子情報通信学会論文誌 A, Vol. J89-A, No. 12, pp. 1090–1103 (Dec. 2006).

[19] 久永 聡, 菅沼優子, 前原秀明, 脇本浩司, 田中 聡: 歩行者 ITS のためのパノラマ画像を用いた 3 次元地図表示方式, 情報処理学会論文誌, Vol. 47, No. 12, pp. 3090–3100 (Dec. 2006).

[20] S.-Y. Hu, J.-F. Chen, and T.-H. Chen: VON: A Scalable Peer-to-Peer Network for Virtual Environments, *IEEE Network*, Vol. 20, No. 4, pp. 22–31 (July 2006).

[21] M. Imai, Y. Hirota, S. Satake, and H. Kawashima: Semantic Connection between Everyday Objects and a Sensor Network, in *Proceedings of the Semantic Sensor Networks (SSN 2006) Workshop*, pp. 1–13 (Nov. 2006).

[22] Y. Ishi, T. Yoshihisa, T. Kawakami, and Y. Teranishi: A Distributed Sensor Data Stream Delivery System with Communication Loads Balancing for Heterogeneous Collection Cycle Requests, in *Proceedings of the 18th IEEE International Conference on Parallel and Distributed Systems (ICPADS 2012) Posters and Demos*, pp. 728–729 (Dec. 2012).

[23] 石井那由他, 白石 陽, 石塚宏紀, 戸辺義人: 空間補間による異種解像度センサデータの統合手法, 情報処理学会論文誌: データベース, Vol. 49, No. SIG 7(TOD 37), pp. 44–58 (Mar. 2008).

[24] 和泉 諭, 加藤 靖, 高橋 薫, 菅沼拓夫, 白鳥則郎: オントロジを利用した健康支援システムの提案とその評価, 情報処理学会論文誌, Vol. 49, No. 2, pp. 822–837 (Feb. 2008).

[25] 金子 雄, 春本 要, 福村真哉, 下條真司, 西尾章治郎: ユビキタス環境における端末の位置情報に基づく P2P ネットワーク, 情報処理学会論文誌: データベース, Vol. 46, No. SIG 18(TOD 28), pp. 1–15 (Dec. 2005).

[26] 片山拓也, 村尾和哉, 田中宏平, 寺田 努, 西尾章治郎: 装着型センサを用いた経路推薦機構を持つナビゲーションシステムの構築, 情報処理学会論文誌, Vol. 50, No. 9, pp. 2350–2359 (Sept. 2009).

[27] 川上朋也, 石 芳正, 義久智樹, 寺西裕一: 収集周期からの負荷推定に基づくセンサデータストリーム配信手法, 電子情報通信学会技術研究報告, Vol. 112, No. 350, pp. 7–12 (Dec. 2012).

[28] T. Kawakami, Y. Ishi, T. Yoshihisa, and Y. Teranishi: A Delivery Method considering Communication Loads for Sensor Data Stream with Different Collection Cycles, in *Proceedings of the 28th ACM Symposium on Applied Computing (SAC 2013)* (Mar. 2013, to appear).

[29] T. Kawakami, B. L. N. Ly, S. Takeuchi, Y. Teranishi, K. Harumoto, and S. Nishio: Distributed Sensor Information Management Architecture Based on Semantic Analysis of Sensing Data, in *Proceedings of the 2008 International Symposium on Applications and the Internet (SAINT 2008) Workshops*, pp. 353–356 (July 2008).

[30] 川上朋也, Ly Bich Lam Ngoc, 竹内 亨, 寺西裕一, 春本 要, 西尾章治郎: センシングデータの意味的解釈に基づく分散センサ情報管理アーキテクチャ, マルチメディア，分散，協調とモバイル (DICOMO 2008) シンポジウム論文集, pp. 1746–1753 (July 2008).

[31] 川上朋也, 三原慶彦, 寺西裕一, 春本 要, 下條真司: ユビキタス環境における位置依存情報の分散管理機構の設計と実装, マルチメディア，分散，協調とモバイル (DICOMO 2006) シンポジウム論文集, pp. 325–328 (July 2006).

[32] 川上朋也, 竹内 亨, 寺西裕一, 春本 要, 西尾章治郎: 地理的オーバレイネットワークの使い分けが可能な位置依存コンテンツ分散管理手法の提案, 第 18 回情報処理学会マルチメディア通信と分散処理ワークショップ (DPSWS 2010) 論文集, pp. 81–87 (Oct. 2010).

[33] 川上朋也, 竹内 亨, 寺西裕一, 春本 要, 西尾章治郎, 下條真司: 地理的オーバレイネットワークを用いた位置依存コンテンツ分散管理手法, 情報処理学会論文誌, Vol. 52, No. 12, pp. 3289–3299 (Dec. 2011).

[34] T. Kawakami, S. Takeuchi, Y. Teranishi, K. Harumoto, and S. Shimojo: A P2P-based Mechanism for Managing Location-dependent Contents in Ubiquitous Environments, in *Proceedings of the 2007 International Symposium on Applications and the Internet (SAINT 2007) Workshops*, pp. 54–57 (Jan. 2007).

[35] 川上朋也, 寺西裕一, 春本 要, 下條真司: 大規模コンテキストアウェアサービスのための Rete アルゴリズムを用いた地理的コンテンツ配信手法について, 情報処理学会研究報告, Vol. 2012-DPS-153, No. 4, pp. 1–8 (Nov. 2012).

[36] 川上朋也, 寺西裕一, 春本 要, 下條真司: センサデータの意味的解釈に基づく分散センサ情報管理システム, 情報処理学会論文誌, Vol. 54, No. 2 (Feb. 2013, 採録決定).

[37] 川上朋也, 義久智樹, 石 芳正, 寺西裕一: 収集周期の異なるセンサデータストリームのための通信負荷を考慮した配信手法, 第 20 回情報処理学会マルチメディア通信と分散処理ワークショップ (DPSWS 2012) 論文集, pp. 136–143 (Oct. 2012).

[38] 木村春彦, 住吉一之, 小林真也, 武部 幹: プロダクションシステムの直接条件照合アルゴリズム, 電子情報通信学会論文誌 D-II, Vol. J77-D-2, No. 2, pp. 370–379 (Feb. 1994).

[39] 小林重信: プロダクションシステム, 情報処理, Vol. 26, No. 12, pp. 1487–1496 (Dec. 1985).

[40] 小西佑治, 寺西裕一, 竹内 亨, 春本 要, 下條真司, 西尾章治郎: センサ観測値分布の概要把握を可能とする階層化ドロネーオーバレイネットワーク構築手法, 情報処理学会論文誌, Vol. 51, No. 2, pp. 624–634 (Feb. 2010).

[41] 米田信之, 阿部昭博, 狩野 徹, 加藤 誠, 大信田康統: 携帯電話とアクティブ RFID による UD 観光情報システムの開発と社会実験, 情報処理学会論文誌, Vol. 49, No. 1, pp. 45–57 (Jan. 2008).

[42] S. Matsuura, K. Fujikawa, and H. Sunahara: Mill: A Geographical Location Oriented Overlay Network Managing Data of Ubiquitous Sensors, *IEICE Transactions on Communications*, Vol. E90-B, No. 10, pp. 2720–2728 (Oct. 2007).

[43] D. P. Miranker: TREAT: A Better Match Algorithm for AI Production Systems, in *Proceedings of the 6th National Conference on Artificial Intelligence (AAAI'87)*, pp. 42–47 (July 1987).

[44] 宮前雅一, 寺田 努, 岸野泰恵, 塚本昌彦, 西尾章治郎: ウェアラブル環境のためのイベント駆動型ナビゲーションプラットフォーム, 情報処理学会論文誌, Vol. 46, No. 3, pp. 694–703 (Mar. 2005).

[45] 南保英孝, 木村春彦, 広瀬貞樹: プロダクションシステムにおけるジョイン演算の順序に関する一考察, 電子情報通信学会論文誌 D-II, Vol. J80-D-2, No. 10, pp. 2790–2799 (Sept. 1997).

[46] R. F. B. Neto, A. A. Macedo, and J. Camacho-Guerrero: Configurable Semantic Services Leveraging Applications Context-Aware, in *Proceedings of the 11th Brazilian Symposium on Multimedia and the Web (WebMedia 2005)*, pp. 1–9 (Nov. 2005).

[47] L. M. Ni, Y. Zhu, J. Ma, M. Li, Q. Luo, Y. Liu, S. C. Cheung, and Q. Yang: Semantic Sensor Net: An Extensible Framework, in *Proceedings of the 3rd International Conference on Computer Networks and Mobile Computing (ICCNMC 2005)*, pp. 1144–1153 (Aug. 2005).

[48] シルビアニッテル, グァンジン, 白石 陽: センサネットワークにおけるネットワーク内空間推定, 電子情報通信学会論文誌 A, Vol. J88-A, pp. 1413–1421 (Dec. 2005).

[49] 大西真晶, 源元佑太, 江口隆之, 加藤宏章, 西出 亮, 上島紳一: ノード位置を用いた P2P モデルのためのドロネー図の自律分散生成アルゴリズム, 情報処理学会論文誌: データベース, Vol. 47, No. SIG 4(TOD 29), pp. 51–64 (Mar. 2006).

[50] 奥 智照, 西出 亮, 上島紳一: P2P 型ジオキャストのための階層ネットワークの提案と評価, 日本データベース学会論文誌, Vol. 7, pp. 97–102 (June 2008).

[51] J. A. Orenstein and T. H. Merrett: A Class of Data Structures for Associative Searching, in *Proceedings of the 3rd ACM SIGACT-SIGMOD Symposium on Principles of Database Systems (PODS'84)*, pp. 181–190 (Apr. 1984).

[52] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker: A Scalable Content-Addressable Network, in *Proceedings of the ACM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM 2001)*, pp. 161–172 (Aug. 2001).

[53] A. I. T. Rowstron and P. Druschel: Pastry: Scalable, Decentralized Object Location, and Routing for Large-Scale Peer-to-Peer Systems, in *Proceedings of the IFIP/ACM International Conference on Distributed Systems Platforms (Middleware 2001)*, pp. 329–350 (Nov. 2001).

[54] 相良 毅, 有川正俊, 坂内正夫: ジオリファレンス情報を用いた空間情報抽出システム, 情報処理学会論文誌: データベース, Vol. 41, No. SIG 6(TOD 7), pp. 69–80 (Sept. 2000).

[55] M. Satyanarayanan: Pervasive Computing: Vision and Challenges, *IEEE Personal Communications*, Vol. 8, No. 4, pp. 10–17 (Aug. 2001).

[56] C. Schmidt and M. Parashar: Squid: Enabling Search in DHT-Based Systems, *Journal of Parallel and Distributed Computing*, Vol. 68, No. 7, pp. 962–975 (July 2008).

[57] Y. Shu, B. C. Ooi, K.-L. Tan, and A. Zhou: Supporting Multi-Dimensional Range Queries in Peer-to-Peer Systems, in *Proceedings of the 5th IEEE International Conference on Peer-to-Peer Computing (P2P 2005)*, pp. 173–180 (Aug. 2005).

[58] Y. Shvartzshnaider, M. Ott, and D. Levy: Publish/Subscribe on Top of DHT Using RETE Algorithm, in *Proceedings of the 3rd Future Internet Symposium (FIS 2010)*, pp. 20–29 (Sept. 2010).

[59] I. Stoica, R. Morris, D. Liben-Nowell, D. R. Karger, M. F. Kaashoek, F. Dabek, and H. Balakrishnan: Chord: A Scalable Peer-to-Peer Lookup Protocol for Internet Applications, *IEEE/ACM Transactions on Networking*, Vol. 11, No. 1, pp. 17–32 (Feb. 2003).

[60] 高田大輔, 小川剛史, 清川 清, 竹村治雄: 身体動作に基づき提示情報を切り替えるコンテキストアウェアなウェアラブル AR システム, ヒューマンインタフェース学会論文誌, Vol. 12, No. 1, pp. 47–56 (Feb. 2010).

[61] 内山寛之, 出口大輔, 高橋友和, 井手一郎, 村瀬 洋: 拡張 DP マッチングを用いた視野角の異なるカメラ映像間の時空間対応付けによる自車位置推定, 電子情報通信学会論文誌 D, Vol. J93-D, No. 12, pp. 2654–2665 (Dec. 2010).

[62] M. Varvello, E. Biersack, and C. Diot: Dynamic Clustering in Delaunay-Based P2P Networked Virtual Environments, in *Proceedings of the 6th ACM SIGCOMM Workshop on Network and Systems Support for Games (NetGames 2007)*, pp. 105–110 (Sept. 2007).

[63] 王 佳な, 野田雅文, 高橋友和, 出口大輔, 井手一郎, 村瀬 洋: Web 上の大量の写真に対する画像分類による観光マップの作成, 情報処理学会論文誌, Vol. 52, No. 12, pp. 3588–3592 (Dec. 2011).

[64] 山崎智弘, 長 健太, 岡本昌之, 岡本雄三, 服部正典: 状況依存エージェントフレームワークの実装と評価, 電子情報通信学会論文誌 D-I, Vol. J88-D-I, No. 9, pp. 1331–1343 (Sept. 2005).

[65] 吉田 幹, 奥田 剛, 寺西裕一, 春本 要, 下條真司: マルチオーバレイと分散エージェントの機構を統合した P2P プラットフォーム PIAX, 情報処理学会論文誌, Vol. 49, No. 1, pp. 402–413 (Jan. 2008).