Vol.2012-MBL-64 No.24
Vol.2012-ITS-51 No.24
2012/11/16

道路幅を考慮した拡大周回道路生成手法とモバイルマップへの応用

伊藤 広記[1,a)]　山本 大介[1,b)]　高橋 直久[1,c)]

概要：本稿では，任意の地点が与えられると，その地点の周りを囲む道路を返す拡大周回道路導出システムを提案し，その実現法について述べる． 提案システムでは，道路幅を考慮して階層的に構造化した拡大周回道路を生成する．また，応答性を重視して，すべての周回道路を予め計算してデータベース化し，周回道路と拡大周回道路の導出要求に対してデータベースの探索だけで応えられるようにしている．本稿では，また，提案システムを用いたモバイルマップについて述べる．さらに，周回道路の導出時間を評価して，インタラクティブなモバイルマップシステムへ適用可能な応答性を有していることを示す．

キーワード：Emma，周回道路， focus+glue+context, ウェブマップサービス，ディストリクト

A Road-width-based Expanded Loop Road method for Mobile Maps

Abstract: This paper proposes a system that returns a road-width-based Expanded Loop Road, which is the area surrounding a given point, and its implementation. An Expanded Loop Road is a hierarchical structure made from small Loop Roads. To minimize the response time, all possible Loop Roads are computed beforehand and stored in a database. As a result, a road-width-based Extended Loop Road is returned on demand by just searching in the database. Moreover, this paper describes an implementation of the proposed system for Mobile Maps. We developed a prototype of our proposed system and evaluated the response time. The results of the evaluation demonstrated the feasibility of the proposed system for interactive Mobile Maps.

Keywords: Emma,LoopRoad,focus+glue+context, Web map service,district

1. はじめに

Focus+Glue+Context 型マップシステム Emma では，頭の中の地理的イメージ（認知地図）に基づいて地図を部分的に拡大，縮小，抽象化して表現することにより，従来のデジタルマップに比べて一覧性と視認性の高い地図の実現を目指している [1], [2]．Emma は，リンチの「都市のイメージ」[3] における５つの要素（パス，ノード，エッジ，ランドマーク，ディストリクト）の視認性を考慮して地図を描画する．利用者が注目する要素を指定すると，その要素を含むディストリクトを均一で大きな縮尺の領域（Focus），周辺を均一で小さな縮尺の領域（Context），縮尺の差により生じる歪みを吸収するように両者を結ぶ道路を抽象化して描画する領域（Glue）からなるマップを提供する．ここで，ディストリクトとは，商店街，オフィス街，公園，神社の境内，学校の敷地など，内部に同じ特徴が見られる都市の部分である．Emma では，ランドマーク（有名で目印になるような建物など）やノード（交差点や駅など）を取り囲む比較的狭い区域もディストリクトとみなして拡大対象とする．

一方，Google Map など多くの WEB マップサービスでは，神社や公園などの施設と位置情報を結びつけるジオコーディングと逆ジオコーディングの機能を提供している．ジオコーディングは施設の名称から，その代表地点の地番や緯度・経度を返し，逆ジオコーディングは，緯度・経度から代表地点がその場所に一番近い施設を距離順にしたリストを返す．しかし，これらの Web マップサービスや市販の地図データでは，ディストリクトの境界データは

[1] 名古屋工業大学大学院工学研究科
Graduate School of Engineering,Nagoya Institute of Technology
[a)] hito@moss.elcom.nitech.ac.jp
[b)] yamamoto.daisuke@nitech.ac.jp
[c)] naohisa@nitech.ac.jp

Vol.2012-MBL-64 No.24
Vol.2012-ITS-51 No.24
2012/11/16

提供されていない場合が多い．

このため，我々は，ディストリクトの境界を求めるために，道路に着目した．道路は，都市部では圧倒的に多く存在し，エッジの役割を果たすことも多い．このことから，我々は，これまでに，道路を境界とする領域をディストリクトの候補とみなして，周回道路を高速に求める手法を提案した．[4], [5] また，任意の地点が指定されると，その地点を含む周回道路を提供するサーバを実現した．

しかし，道路はディスクリクトの境界とはならずに，ディスクリクトの内部を貫通するパスとなることもある．このため，自動的にディストリクトの境界を決定することは難しい．また，幅が広い道路は，狭い道路に比べ，歩行者の横断を妨げるので，ディスクリクトの境界になりやすい傾向がある．これらの点を考慮して，本稿では，下記要件を満たす拡大周回道路導出システムを提案する．

要件 1　ディストリクトの境界となりうるすべての周回道路を網羅的に求めること．

要件 2　道路を境界とするディストリクトだけでなく，道路を内部に含むディストリクトも提示し，利用者が選択できること．

要件 3　幅の広い道路を境界とし，幅の狭い道路を内部に含むディスクリクトを提示できること．

要件 4　周回道路，拡大周回道路をインタラクティブに求める場合に，ユーザがストレスを感じない程度の時間内に高速に提示できること．

2. 道路幅を考慮した拡大周回道路導出システム

2.1 提案システムの概要

周回道路とは，**図 1** のように，施設を取り囲む道路のことである．1 章で述べた要件を満たすため，道路の幅に応じて分類された道路集合のそれぞれに対して独立に周回道路を作成する．その際，高速化のために，道路幅に応じた周回道路データベースを予め作成する． また，内部に道路を含むディストリクトを提示するには，図 1 のように，周回道路をさらに取り囲む周回道路（以下，拡大周回道路）を求める．しかし，このように道路の幅を考慮せず拡大するだけでは，要件 2 を満たさないので，**図 2** のように，道路幅により拡大する方向に制限を加えた拡大周回道路を求める．提案システムの特徴をまとめると以下のようになる．

特徴 1　重複探索防止機能及び二次メッシュ境界を考慮した周回道路探索機能を実現する（要件 1 に対応）

特徴 2　周回道路導出を，周回道路データベース構築と周回道路検索の 2 つの機能に分割して実現する．前者の機能では，予め，すべての道路幅に対して，それぞれ総ての周回道路をデータベース化しておく．これにより，地点と道路幅が指定されたときに，その道路幅の周回道路を高速に提示できるようにする．（要件 3 に対応）．

図 1　椙山女学園に対しての周回道路及び拡大周回道路

Fig. 1　Example of LoopRoad and Expanded LoopRoad.

図 2　椙山女学園に対しての道路幅を考慮した拡大周回道路

Fig. 2　Example of Expand LoopRoad considering the road width.

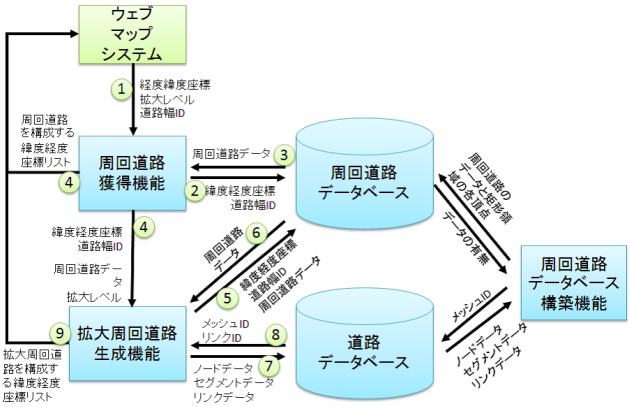


図 3　提案システムの構成図

Fig. 3　Diagram of the proposed system.

特徴 3　指定された道路幅の道路だけからなる周回道路を組み合わせて拡大周回道路を構築する機能を実現する（要件 1 に対応）．

特徴 4　道路幅を考慮して拡大する方向を制御することにより，道路幅を考慮して階層的に構造化した拡大周回道路を生成する機能を実現する．これにより，道路幅の広い道を境界として重視した拡大周回道路を提示可能になる（要件 2 に対応）．提案システムは，**図 3** に示すように， 周回道路データベース構築機能，周回道路獲得機能，拡大周回道路生成機能，周回道路データベース，道路データベースからなる．各構成要素について以下で述べる．

2.2 道路データベース

道路データは**図 4** のようにリンク，セグメント，ノードからなる．それぞれ，**表 1**，**表 3**，**表 4** のような形式で道路データベースに格納されている．

リンクは，隣接する二つの交差点の Head ノードと Tail

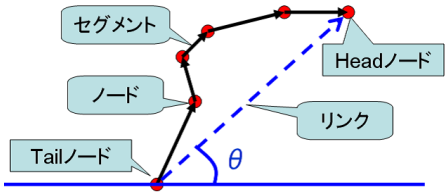


図 4 道路データの構造
Fig. 4 Structure of road data.

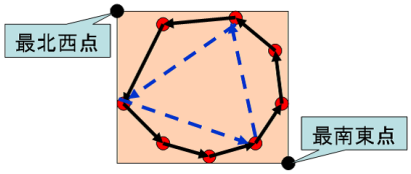


図 5 周回道路を内包する最小の矩形領域
Fig. 5 The smallest rectangle enclosing the LoopRoad.

ノードを Tail ノードから Head ノードに向かう有向枝であり，セグメントは道路形状に沿った有向枝である．これらは，**表 2** の形式のデータによって対応付けられ同様に道路データベースに格納されている．
また，道路データは全国のデータをすべて 1 つのテーブルに格納するには膨大すぎる．このため，第 2 次地域区画（二次メッシュ）[6] と呼ばれるメッシュ構造に分割した領域ごとにテーブルを構成し，二次メッシュにそれぞれ割り当てられた番号（以下，メッシュ ID）をテーブル名とした．

2.3 周回道路データベース

周回道路と拡大周回道路は複数のリンクからなる．周回道路データベースは，**表 5**，**表 6** に示す項目からなる周回道路テーブル，隣接周回道路テーブルからなる．
周回道路テーブルのスキーマは表 5 に示すように，周回道路 ID，矩形領域の各頂点，リンク ID リスト，緯度経度座標リストからなる．ただし，周回道路 ID は主キーとしユニークである．リンク ID リスト及び緯度経度座標リストは，周回道路の探索時に順番に出現したリンクとその端の点の緯度経度座標を格納したリストである．これらは，データベースから獲得する際に再構成しやすくしている．矩形領域とは，**図 5** のように周回道路を内包する最小の矩形であり，矩形の緯度経度の最大値と最小値からなる計 4 点を周回道路テーブルに格納する．これにより，周回道路をデータベースから効率的に検索しやすくした．

隣接周回道路テーブルのスキーマは表 6 に示すように，リンク ID，左隣接周回道路 ID，右隣接周回道路 ID) の項目から成る．ただし，リンク ID は，主キーとし，ユニークである．左隣接周回道路 ID は，数値が入っていた場合，そのリンク ID の左周りに探索した周回道路の ID であり，もし数値が入っておらず，null となっていた場合は，左回りには周回道路が探索されていないことを表す．右隣接周回道路 ID も，数値が入っていた場合と入っていなかった場合は同様の状態を表す．これにより，周回道路の重複探索を防止することができる．また，周回道路の隣接関係をも示すため，拡大周回道路を効率的に導出しやすくした．

また，周回道路テーブル，隣接周回道路テーブルは道路幅ごとに作成し，テーブル名の末尾に道路幅 ID を付与した．周回道路テーブル，隣接周回道路テーブルともに，初期状態は null である．

表 1 リンクテーブル
Table 1 Links

項目名	型	説明
リンク ID	int	リンクを一意に識別する ID
角度	double (-180〜180)	西から東を 0 度とした Tail ノードから Head ノードを結ぶ角度
Tail ノード ID	int	リンクの始点となるノードを一意に識別する ID
Head ノード ID	int	リンクの終点となるノードを一意に識別する ID
道路幅 ID	int	1〜4 からなる道路幅の広さを識別する ID. 数字が小さいものほど道路幅が大きい

表 2 リンクとセグメントを一致させるテーブル
Table 2 match the Links and Segments

項目名	型	説明
リンク ID	int	リンクを一意に識別する ID
セグメント ID	int	セグメントを一意に識別する ID

表 3 セグメントテーブル
Table 3 Segments

項目名	型	説明
セグメント ID	int	セグメントを一意に識別する ID
角度	double (-180〜180)	西から東を 0 度とした Tail ノードから Head ノードを結ぶ角度
Tail ノード ID	int	セグメントの始点となるノードを一意に識別する ID
Head ノード ID	int	セグメントの終点となるノードを一意に識別する ID

表 4 ノードテーブル
Table 4 Nodes

項目名	型	説明
ノード ID	int	ノードを一意に識別する ID
緯度	double	ノードの緯度
経度	double	ノードの経度

表 5 周回道路テーブルのデータ構造
Table 5 Table of LoopRoad

項目名	型	説明
周回道路 ID	int	周回道路を一意に識別する ID
最東点	double	周回道路を内包する矩形領域の緯度の最高値
最西点	double	周回道路を内包する矩形領域の緯度の最低値
最北点	double	周回道路を内包する矩形領域の経度の最高値
最南点	double	周回道路を内包する矩形領域の経度の最低値
リンク ID リスト	text	周回道路を構成するリンクをカンマにて区切って格納
緯度経度座標リスト	text	周回道路を構成する多角形の緯度経度座標の組毎にカンマにて区切って格納

表 6 隣接周回道路テーブルのデータ構造
Table 6 Table of Adjoinig LoopRoad

項目名	型	説明
メッシュ ID	int	リンクを一意に識別する二次メッシュ番号
リンク ID	int	リンクを一意に識別する ID
左隣接周回道路 ID	int	リンク方向に対して左に隣接する周回道路 ID
右隣接周回道路 ID	int	リンク方向に対して右に隣接する周回道路 ID

2.4 周回道路データベース構築機能

周回道路データベース構築機能は道路データベースから道路幅ごとに道路リンクを取得し，周回道路を探索し，道路幅毎に定められたテーブルに格納する．これにより，周回道路データベースを構築する．しかし，単純にリンクを取得し探索しようとすると，次のような問題が発生する．

例えば，リンクを選択して探索する場合，**図 6** で示すように，周回道路を探索する方向が片方向のみであると，探索されない周回道路が出来る可能性がある．しかし単純に両方向に周回道路を探索すると，既に導出された周回道路を再び探索する場合が有り効率が良くない．さらに，複数の道路リンクから成る周回道路の場合，その道路リンク数に応じて何度も周回道路を探索する場合がある．上記問題を解決するために，隣接周回道路テーブルを用いたタビュレーション（索表計算）により，複数回同じ周回道路を探索しないように工夫した．

また，すべての 2 次メッシュの道路リンクに対して周回道路データベースを構築しようとした場合，すべての二次メッシュの道路リンク集合を取得し，周回道路を探索すれば解決できるかもしれない．しかし，それには大量のメモリを必要とする．そこで，n × m のメッシュの領域の周回道路データベースを作成するに当たり，まず**図 7** のように 2 × 2 の 4 つのメッシュに含まれる道路リンクを取得する．そして，二次メッシュの大きさは緯度経度座標により一定であるので，メッシュが 4 分の 1 ずつ含む中央部に道路リンクが含まれる場合，周回道路の探索を行う．このような 4 メッシュを n × m 個の二次メッシュすべてに繰り返し適応することにより，少ないメモリ量で，すべての周回道路をデータベースに格納することが可能になる．

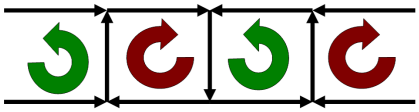


図 6 片方向探索のみでは探索漏れが起きる道路データの例
Fig. 6 Pattern of missing data.

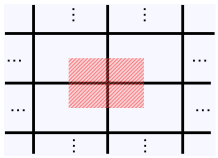


図 7 4 つのメッシュと探索範囲
Fig. 7 4 meshes and search range.

さらに，周回道路を探索するメッシュを選択し，探索範囲に何も制限を設けず周回道路を探索した場合，取得したメッシュ範囲の一番外側をなぞるような周回道路を作成してしまう．この際に作成される周回道路は意図する周回道路ではないため，誤りである．そのため，4 つのメッシュの境界線に探索しているリンクがたどりついた時点で探索を中止することにより，メッシュ範囲をなぞるような周回道路の作成を防止することが可能となる．

2.5 周回道路獲得機能

周回道路獲得機能では，図 3 のように道路幅 ID と緯度経度座標から，周回道路データベースに対応する道路幅 ID と経度緯度座標を入力することで，対応した道路幅の周回道路を高速に獲得することができる．

2.6 拡大周回道路生成機能

拡大周回道路生成機能では，道路幅を考慮て周回道路を拡大する．具体的には，周回道路が与えられると，その周回道路を含む，より幅の広い周回道路を取得しその領域と幅の狭い道路の拡大周回道路で囲まれた領域の論理積を取ってできる領域を囲む道路を拡大周回道路とする．これにより，道路幅の広い道を境界として重視した拡大周回道路を生成することが可能となる．また，上記拡大周回道路道路が幅の広い道路とだけ一致した場合には，道路幅の広い周回道路を用いて拡大する．

次章より，機能毎の実装手順について述べる．

3. 提案システムの実現法

3.1 周回道路データベース構築手順

以下の手順を道路幅に応じて繰り返すことにより，道路幅に応じた周回道路を周回道路データベースに格納する．

手順 1 全ての二次メッシュからなる集合 M を取得する．

手順 2 M のすべての要素 m に対して，手順 3-1, 3-2, 3-3 を行う．

手順 3-1 m を基準に周回道路探索を行うための範囲とその中心領域を決定する．

手順 3-2 範囲内の全ての道路リンクの集合 S を取得する．

手順 3-3 S のすべての要素に対し，手順 4-1, 4-2 を行う．

手順 4-1 S から任意の道路リンク L を選択し，そのリンクが範囲の中心領域内であるか判定する．

手順 4-2 中心領域内の場合，L をたどり，両方向に周回道路を探索し，それぞれ周回道路データベースに格納する．そうでない場合，周回道路探索を行わない．

手順 3-1, 3-2, 3-3 は 3.1.1 節にて説明する周回道路探索範囲選択機能により実現される．また，手順 4-2 は 3.1.2 節にて説明する周回道路探索機能により実現される．

3.1.1 周回道路探索範囲選択機能手順

以下のステップを繰り返すことにより，n × m メッシュの周回道路を探索する道路リンクを選択する．

STEP1 探索するメッシュの最南西メッシュを探索基準メッシュ Md とする。

STEP2 Md を最南西端とする 2 × 2 のメッシュ Me の全ての道路リンク集合 S を取得する．

STEP3 Me の中心緯度経度座標を算出する．そして，この座標から東西に 1/16 ずつ，南北に 1/24 ずつ広げた領域 MA を導出し，S のすべての要素に対し STEP4 を繰り返す．

STEP4 S からリンクを 1 つ選択し L とする。L の Tail もしくは Head ノードのどちらかが領域 MA 内に含まれていたとき，周回道路探索を行う．両方含まれていないときは探索を行わない．

STEP5 Md が最東端メッシュを含む場合，STEP6 を実行する．そうでない場合，Md を 1 メッシュ東にずらし，新たな Md とする。また、Md を基準とした 2 × 2 メッシュを新たに Me とし、STEP2 に戻る．

STEP6 Md が最北端メッシュを含む場合，終了する．そうでない場合，Md を 1 メッシュ北にずらし新たな Md とし，Md を基準とした 2 × 2 メッシュを新たに Me とし STEP2 に戻る．

3.1.2 周回道路探索機能手順

ある範囲に対し探索を行う初期リンク $L0$ に対し以下のステップを実行することにより，$L0$ からなる周回道路を左回り，右回りに対し探索し，周回道路データベースに格納することができる．

入力 探索を行うリンク集合 S，探索を行うリンク $L0$，隣接周回道路テーブル

出力 $L0$ を起点とする左右の両方向の周回道路を格納した隣接周回道路テーブル，矩形領域の各頂点と構成するリンク ID のリストを格納した周回道路テーブル

STEP1 探索を行うリンクを一つ選択し $L0$ とする．隣接周回道路テーブルから，$L0$ のリンク IDk の行を獲得する．その行の左隣接周回道路 ID が null の場合は，STEP2 を行い，周回道路探索を実行する．ID が存在した場合は，STEP5 を実行し，リンク IDk に対しての左回りの周回道路探索を実行しない．

STEP2-1 $L0$ に対し，左回りの周回道路探索を実行する場合，$L0$ の Head ノードと一致するノードを持つ道路データ集合 $L0s$ を取得する．右回りの探索を行う場合，$L0$ の Tail ノードと一致するノードを持つ道路データ集合 $L0s$ を取得する．

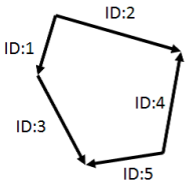
STEP2-2 **図 10** のように $L0$ と $L0s$ の角度を比較し，左回りの場合，$L0s$ の中の Tail から Head への方向に対し最左枝を $L0s1$ とし，右回りの場合，$L0s$ の中の Head から Tail への方向に対し最左枝を $L0s1$ とし，周回道路候補集合 $Lstack$ にリンク ID とリンクの向きを追加する．

STEP2-3 $L0$ の Head ノードと $L0s1$ の Head ノードが一致していた場合は $L0s1$ の Tail ノード，$L0$ の Head ノードと $L0s1$ の Tail ノードが一致していた場合は $L0s1$ の Head ノードと一致するノードを持つ道路データ集合 L1s を取得する．

STEP2-4 $L0s1$ と $L1s$ の角度を比較し，$L1s$ の中の最左枝を $L1s1$ とし，$Lstack$ に $L1s1$ のリンク ID とリンクの向きを追加する．

STEP2-5 STEP2-3,STEP2-4 を繰り返し，**図 9** のように $L0$ の ID と向きが $Lstack$ に含まれる ID と向きと一致した場合，一致したものを除いた $Lstack$ を周回道路として出力，該当周りの探索を終了する。また、道路データのノードがメッシュの外側にたどり着いたときに失敗として該当回りの探索を終了する．

STEP3 周回道路探索を実行し，探索失敗の場合は，STEP5 へ，探索できた場合，今回探索した周回道路 ID を決め（今回は A とする），探索できた周回道路を構成する全てのリンクのリスト，D に対して，リンクの向きを調べ，隣接周回道路テーブルに格納する．リンクの向きが初期リンク L と同じならば，リンク ID の行の左隣接周回道路 ID に今回探索した周回道路 IDA を書き込む．逆方向であれば，リンク ID の行の右隣接周回道路 ID に今回探索した周回道路 IDA を書き込む．周回道路を探索し，A を 1，すべてのリンクのメッシュ ID が 523656 であった場合に格納した例を **図 8** に示す．これを，今回探索した周回道路を構成す



メッシュID	リンクID	左隣接周回道路ID	右隣接周回道路ID
523656	1	1	Null
523656	2	Null	1
523656	3	1	Null
523656	4	1	Null
523656	5	Null	1

図 8 隣接周回道路テーブルの格納の例
Fig. 8 Example of storing adjoining table.

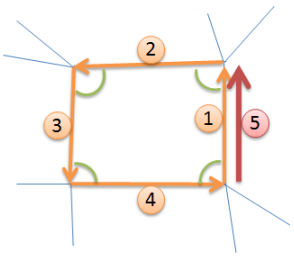


図 9 あるリンクから探索される周回道路
Fig. 9 LoopRoad of some Link to be searched.

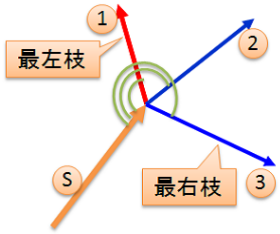


図 10 周回道路探索時の優先順序
Fig. 10 Order of priority in searching LoopRoad.

る全てのリンクに対して実行する．

STEP4 隣接周回道路テーブルに書き込んだ後，探索した周回道路を構成するリンクを構成するセグメントを取得する．全てのセグメントのノードの点，経度緯度のそれぞれの最大値，最低値を計算して，矩形領域の四隅の点の座標と，周回道路を構成するリンク ID のリスト及び周回道路を構成する緯度経度座標のリストを，今回探索した周回道路 IDA とともに，周回道路テーブルに格納する．

STEP5 右回りに対して，STEP1 から STEP4 までほぼ同様の手順にて実行する．

3.2 周回道路獲得手順

周回道路獲得機能の入出力は以下の通りである．

入力 経度緯度座標 P，道路幅 IDRW

出力 周回道路を一意に識別する ID，周回道路を構成するリンク ID のリスト，周回道路を構成する緯度経度座標のリスト

また，以下のステップを実行することにより，道路幅 IDRW に応じた入力座標 P を内包するの周回道路を獲得することができる．

STEP1 座標 P を用いて，**図 11** で示す SQL 文を用いて，道路幅 IDRW に応じた周回道路テーブルにアクセスし，座標 P が矩形の範囲内に存在する周回道路候補集合 As を獲得する．As は，それぞれ構成するリンク ID のリスト LL と経度緯度座標のリスト PL を持つ．LL は周回道路の探索順になっており，LL のリンクを順に構成していくと，リンクの多角形を構成することができる．

```
select リンク ID from 周回道路テーブル
where 最西点 ≤ 経度 AND 最東点 ≥ 経度
AND 最南点 ≤ 緯度 AND 最北点 ≥ 緯度 ;
```

図 11 周回道路を取得する SQL 文
Fig. 11 SQL statement to get a LoopRoad

STEP2 経度緯度座標リスト PL に対して，多角形を構成し，多角形内に座標 P が含まれるかどうか判定する手法を用いる．

STEP3 全ての候補集合 As に対し，STEP2 を実行し，多角形内に P を含む周回道路 PN が，求めるべき周回道路である．

3.3 拡大周回道路導出手順

拡大周回道路導出の入出力は以下の通りである．

入力 緯度経度座標 P，道路幅 IDRW，拡大レベル $Level$

出力 道路幅 ID を用いた拡大レベルの拡大周回道路を構成する経度緯度座標のリスト

また，以下のステップを**図 12** のように実行することにより道路幅 IDRW に応じた入力 P を内包する周回道路を $Level$ 回拡大した，道路幅を考慮した拡大周回道路を導出することができる．

STEP1 周回道路導出手法により対応する道路幅 IDRW の周回道路の周回道路 IDAid と周回道路を構成する LinkID 群 AL を取得する．

STEP2 取得した道路幅 IDRW が 1 の場合，STEP4 を実行する．道路幅 IDRW が 1 でない場合，対応する周回道路 ID-1 の周回道路 U_Aid とその周回道路を構成する LinkID 群 U_AL を取得する．この時点で取得した AL と U_AL の例を**図 12(a)** に示す．橙の境界が AL であったとき，赤色の境界が U_AL となる．

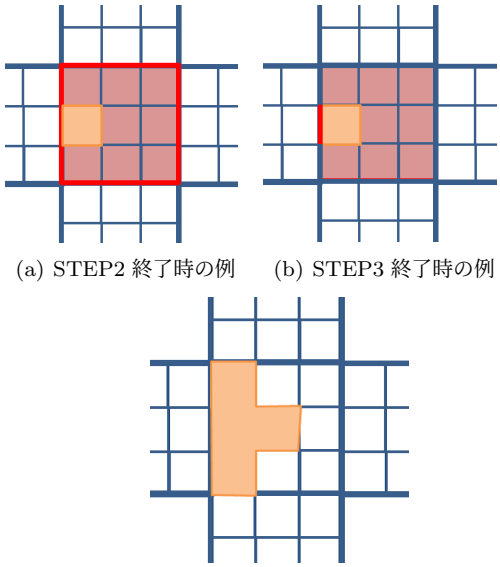
STEP3 取得した周回道路 ID-1 の周回道路を構成する LinkID 群 U_AL と周回道路を構成する LinkID 群 AL と重複している LinkID を重複 LinkID 集合 DL とする．**図 12(b)** に DL の例を示す．DL を赤いラインで示す．

重複 LinkID 集合 DL がすべての周回道路 ID-1 の周回道路を構成する LinkID 群 U_AL と一致した場合，STEP2 の RW に RW-1，Aid に周回道路 ID-1 の周回道路 IDU_Aid，AL に周回道路を構成する LinkID 群と周回道路 ID-1U_AL を入力として STEP2 を実行する．

STEP4 周回道路 ID を構成する LinkID 群 AL を Area_Links テーブルに対し検索し，LinkID 群 AL を含む周回道路 ID 集合 N_Aid を取得する．

STEP5 LinkID 群を含む周回道路 ID 集合 N_Aid から現在の周回道路 IDA_id を取り除き新たな N_Aid とす

Vol.2012-MBL-64 No.24
Vol.2012-ITS-51 No.24
2012/11/16



(a) STEP2 終了時の例　(b) STEP3 終了時の例

(c) STEP8 終了時の例

図 12　道路幅を考慮した拡大手法の実行例

Fig. 12　Sample of Expand LoopRoad method considering the road width.

る．そして周回道路 ID 集合 N_Aid を構成する LinkID 群 N_AL を取得する．

STEP6　周回道路 IDN_Aid を構成する LinkID 群 N_AL から現在の周回道路 ID を構成する LinkID 群 AL を取り除き新たな N_AL とする．

STEP7　STEP3 の入力 AL に周回道路 ID を構成する LinkID 群 N_AL に重複 LinkIDDL を追加したもの，STEP3 の入力 Aid に周回道路 ID 集合 N_Aid として拡大レベル L 回繰り返す．

STEP8　LinkID 群を含む周回道路 ID 集合 N_Aid を構成する LinkID 群 N_AL を用いて周回道路探索を行い，周回道路を構成する．これが $Level$ 回拡大した拡大周回道路である．**図 12(c)** に $Level$ が 1 の例を示す．橙に塗りつぶされたところが 1 レベルの拡大周回道路である．

4.　周回道路を用いたモバイルマップシステム

図 4 に道路幅を考慮した周回道路をモバイルマップに適応させた例を示す．音声や操作等により施設等の代表地点を取得し，図に存在する+と-ボタンを用いることにより，道路幅を用いた拡大周回道路を拡大，縮小することを可能とする．

また，Emma に適用させた例を**図 4** に示す．Emma に提案手法を適応させることにより，Focus 部にディストリクトを拡大表示させることができる．

音声や操作等により施設等の代表地点を取得し，F ボタンにより Focus 部を作成する．そして，図に存在する+と-ボタンを用いることにより，道路幅を用いた拡大周回道路を拡大，縮小することを可能とする．

図 13　椙山女学園に対し周回道路の範囲をマップに適応した例

Fig. 13　Example of applying a LoopRoad on a map.

図 14　椙山女学園に対し周回道路の範囲を Emma に適応した例

Fig. 14　Example of Emma that reflects the LoopRoad.

5.　評価実験

提案手法の有効性を示すために周回道路，道路幅を考慮した拡大周回道路の導出時間について評価を行った．評価を行うにあたり作成した周回道路データベースついての結果も載せる．

5.1　周回道路データベース作成結果

今回，周回道路を作成するに当たり，道路データは近畿，中部圏を中心とした 2 次メッシュ 385 個からなる住友電工株式会社の道路データを用いた．実行環境は以下の通りである．

- CPU：Core i7 870 (2.93GHz)
- メモリ：3.49GB
- OS：Windows7 Professional 32bit
- 使用言語:Java,MySQL

表 7 に結果を示す．358 個の 2 次メッシュに対し，すべての道路を用いたデータベースの作成時間は約 5 時間 15 分程度であり，4 種の道路幅の周回道路データベースはそれぞれ別道路テーブルに書き込むため，別プロセスで作成するように並列化可能である．そのため，今回のように逐次処理をしたにもかかわらず 1 日以内にて作成できたことは妥当な速度であるといえる．

また，中部・近畿圏を中心とした 385 メッシュに対して作成した周回道路データベースの容量は，**表 ??**の各道路幅に対する作成データ量を合計して求まる 577.2MB である．全国の陸地の二次メッシュの総数は高々 30 倍であるため，全国の道路に対して周回道路データベースを作成した場合でも，20GB 程度に収まると推定できる．したがって，一般的な PC を用いて周回道路データベースサーバを構築で

表 7 周回道路データベース構築結果
Table 7 Result of LoopRoad database construction

	道路幅 ID:1	道路幅 ID:2	道路幅 ID:3	道路幅 ID:4
データベース作成時間 (s)	503	11028	18195	18788
作成周回道路数	1473	137022	394316	394794
作成データ量 (MByte)	3.7	95.0	238.7	239.8

表 8 道路データ種別毎のリンク数
Table 8 Type of road data

二次メッシュ ID	道路幅 ID:1	道路幅 ID:2	道路幅 ID:3	道路幅 ID:4
523656	890	19271	12134	30
523657	1719	32528	7155	0

表 9 周回道路平均導出時間（ms）
Table 9 Average derivation time of the LoopRoad

道路幅 ID:1	道路幅 ID:2	道路幅 ID:3	道路幅 ID:4
14	255	362	353

表 10 道路幅を考慮した拡大周回道路平均導出時間（ms）
Table 10 Average derivation time of the Expand LoopRoad

拡大レベル	道路幅 ID:1	道路幅 ID:2	道路幅 ID:3	道路幅 ID:4
1	105	295	519	842
2	135	305	531	864
3	NaN	304	531	866
4	NaN	314	526	876
5	NaN	328	536	886

きるといえる．

5.2 周回道路導出に関しての評価

周回道路データベースを作成後，周回道路及び道路幅を用いた拡大周回道路を導出する API を作成し，5.1 節と同様の環境にて導出時間を取得した．緯度経度座標をランダムに 200 回入力して，それぞれの入力座標に対する周回道路と拡大周回道路を求めた．このときの平均導出時間を，**表 9**, **表 10** に示す．表で NaN は導出不能であることを示す．すべての場合において，平均導出時間が 1 秒以下であった．インタラクティブなシステムに適応できるといえる．道路幅 ID:1 の拡大レベル 3 以上の値が NaN であるのは，**表 8** のように道路幅 ID:1 の道路データが少ないために周回道路があまり作成されず，また隣接関係を持たないことが多く，拡大周回道路を導出することができなかったためである．

6. 終わりに

本稿では，道路幅を考慮した拡大周回道路導出手法について述べた．また，評価実験を行い，インタラクティブなシステムとして耐えられる時間にて提示ことができることを示した．本稿では，任意の地点に対し，その地点の周りを道路幅を考慮して囲む道路を返す拡大周回道路導出システムを提案し，その実現法について述べた． また，周回道路の導出時間を評価し，インタラクティブなモバイルマップシステムへ適用可能な応答性を有していることを示した．今後の課題として，より高速に道路幅を考慮した拡大周回道路を導出するべきだと考える．

参考文献

[1] Naohisa Takahashi: An Elastic Map System with Cognitive Map-based Operations, International Perspectives on Maps and the Internet, Springer-Verlag, pp.73-87, Feb. 12, 2008.
[2] Daisuke Yamamoto, Shotaro Ozeki, Naohisa Takahashi, Focus+Glue+Context: An Improved Fisheye Approach for Web Map Services, Proceedings of the ACM SIGSPATIAL GIS 2009, Seattle, Washington, pp.101-110, 2009.11
[3] Lynch,K. , The Image of the City, Cambridge,MIT Press,1960
[4] 伊藤広記，山本大介，高橋直久，周回道路データベースの構築とそれに基づく高速な周回道路導出手法，第 3 回データ工学と情報マネジメントに関するフォーラム (DEIM2011)，B4-3(2011)
[5] Daisuke Yamamoto, Hiroki Itoh, Naohisa Takahashi, "One Click Focusing: An SQL-based Fast Loop Road Extraction Method for Mobile Map Service", Proceedings of the 4th International Conference on Advanced Geographic Information Systems, Applications, and Services (GEOProcessing 2012), pp.7-16, Valencia, Spain, 2012.1.30.
[6] 日本統計協会：メッシュ統計 (online)，入手先 ⟨http://www.jstat.or.jp/mesh/index.html⟩