

指定された2点間の経路の見やすさを考慮した略地図生成

竹内 真樹[†]森口 昌樹[‡]今井 桂子[§]中央大学大学院 理工学研究科 情報工学専攻[†]中央大学 理工学部 情報工学科^{‡§}

1 序論

必要な情報を強調するために道路や建物情報の変形や取捨選択などを行った地図を略地図という。現在、略地図は案内用地図や鉄道路線図などで利用されている。コンピュータを用いて地図を生成する際、道路網をノードとエッジの道路ネットワークとして考える事が多い。本研究でも、道路ネットワークとして考えて略地図を生成する。

略地図生成の既存手法の1つとして[1]がある。[1]の手法の特長として道路同士が近い場合、道路間の距離を離し見やすい略地図を生成する事ができる。[1]では地図の見やすさを定義し、その定義にしたがったコスト関数を作成し、コスト関数の値が良くなる様に道路網を変形させ、道路網が密集して見にくい箇所を広げる手法を提案している。また[1]は目的地点周辺の道路を詳細に抽出し、それ以外の箇所は主要道のみを抽出する事で、目的地点への案内に必要な情報を減らしている。

しかし、地図を実際に利用する場合、目的地点だけではなく、駅やバス停の様な出発地点、出発地点から目的地点までの経路の情報が重要だと考えられる。したがって、出発地点、目的地点、経路周辺を詳細に抽出する必要がある。その様な地図に対して[1]を適用させた場合、詳細に抽出した部分が複数あるため、目的地点(出発地点)周辺のみ広げ、出発地点(目的地点)周辺を広げられるとは限らない。

そこで本研究では道路ネットワークを出発地点を含む領域、目的地点を含む領域と分割し、各々に対して[1]を適用させ、出発地点から目的地点までの経路が見やすい略地図を生成する手法を提案する。

2 既存手法

ここでは既存手法[1]について説明する。[1]では以下のa, b, cの順に操作し、見やすい略地図を生成する。

- (a) 道路ネットワークの抽出
- (b) 道路ネットワークの簡略化
- (c) 道路ネットワークの変形

(a)の道路ネットワークの抽出では目的地点周辺はユーザが利用する可能性が高いため詳細に抽出し、それ以外の箇所はユーザが利用する可能性が低いと考え、主要道のみ抽出する。

(b)の道路ネットワークの簡略化では、次数2のノードの削除を繰り返す操作を行う。削除する順序はノードを削除時に生じる幾何誤差が小さいものから順に削除していく。削除するときに交差が生じる場合、およびノードが鋭利な角度を構成している場合は削除は行わない。終了条件は削除できるノードがなくなるまで行う。

最後に(c)の道路ネットワークの変形を行う。最初に見やすい略地図とはどのようなものかを考え、その様な略地図が

得られると思われる目的関数を定義をし、その定義に合う関数を作成する。見やすい略地図とは、以下のI~VIとする。 $e_{int}, e_{len}, e_{rlen}, e_{ang}, e_{rang}$ は定義に沿って作成した各関数で値が小さい程良いとし、 e_{int} はI, IIを、 e_{len} はIIIを、 e_{rlen} はIVを、 e_{ang} はVを、 e_{rang} はVIを考慮して作った関数である。 $\lambda = (\lambda_{int}, \lambda_{len}, \lambda_{rlen}, \lambda_{ang}, \lambda_{rang})$ は各関数 e に対応した重みである。各関数 e に対応する重み λ を掛け、足したものをコスト関数 E とする。 d_{int}, d_{len} は設定した十分な長さとする。

- I エッジが交差してはいけない。
- II 全てのエッジに対して端点を共有しない他のエッジとの最小距離が十分な長さ(d_{int})を持つと良い。
- III 全てのエッジの長さが十分な長さ(d_{len})を持つと良い。
- IV 2本のエッジの比が保たれる程良い。
- V エッジの角度のずれが小さい程良い。
- VI 2本のエッジの関係性が保たれる程良い。

次にコスト関数 E の値が良くなる様に道路ネットワークを変形させる。基本的には以下の様にノードの移動を行う。

1. 領域内にあるノード n をランダムに選択する。
2. 選択したノード n を通る直線 l をランダムに引く。
3. 直線 l によって分けられた2つの領域の片方をランダムに選択する。選択された領域を S とする。
4. 領域 S に含まれるノードに対して直角が放射線状にノードを移動させる。直角が放射線状かはランダムに選ばれる。
 - (a) 直角: 直線 l に対して垂直方向に移動させる。移動させる位置は直線 l から領域 S 内のノードまでの最小距離の $[0.95, 1.05]$ 倍の位置である。
 - (b) 放射線状: ノード n に対して放射線状に移動させる。移動させる位置はノード n から領域 S 内のノードの距離の $[0.95, 1.05]$ 倍の位置である。
5. 道路ネットワーク全体が領域内にちょうど収まる様にスケールを整える。コスト関数を用いて値の比較を行い、解が良くなったら更新する。

1~5の操作を500回行う。500回前のコスト関数の値と比較し、0.5%以上良い値になれば、さらに1~5の動作を500回繰り返す。0.5%未満の場合は終了する。

以上のアルゴリズムを重み λ を変えて、2回行う。1回目(Stage 1)では角度のずれを大きく生じさせてでもエッジを伸ばす様に重みを設定する。2回目(Stage 2)では角度を補正する様に重みを設定する。

3 提案手法

既存手法のノードの移動を行うと、エッジ間の最小距離やエッジ自体の距離を伸ばし、道路ネットワークを領域外に出す事がある。そのとき、領域外に出た道路ネットワークを領域内に収めるために道路ネットワーク全体を縮小する。この操作を出発地点から目的地点までの案内用地図に適用させると、出発地点(目的地点)周辺の経路上のエッジが伸ばされ、目的地点(出発地点)周辺の経路上のエッジを縮められる。すると目的地点(出発地点)周辺の経路上のエッジが見にくくなる可能性が高い。

そこで提案手法では出発地点(目的地点)周辺のエッジを伸

Schematic road maps generation considering the readability of the path between two specified points

[†] Masaki TAKEUCHI, Information and System Engineering Course, Chuo University

[‡] Masaki MORIGUCHI, Department of Information and System Engineering, Chuo University

[§] Keiko IMAI, Department of Information and System Engineering, Chuo University

ばす操作や、エッジ間の最小距離を離す操作を行ったとしても、目的地点（出発地点）周辺のエッジの長さの短縮や、エッジ間の最小距離を近づけない様な手法を提案する。既存手法との変更点を以下にまとめる。

- 出発地点、目的地点、経路周辺の道路ネットワークを詳細に抽出し、それ以外の所は主要道のみとする。
- 簡略化まで行なった道路ネットワークを2分割し、各々の道路ネットワークに対して既存手法のノード移動を適用させ、その後、2つの道路ネットワークを結合させる。
- 経路の見やすさを考慮したコスト関数に変更する。

提案手法では最初に出発地点、目的地点、経路周辺の道路ネットワークを詳細に抽出し、それ以外の所は主要道のみ抽出する。その様にして出発地点から目的地点までの案内に不要な情報を減らす。その後、既存手法と同じ簡略化を行う。

次に道路ネットワークの分割を行う。まず出発地点のノード n_s 、目的地点のノード n_t の垂直二等分線をもとめる。この垂直二等分線を分割直線と呼ぶ。分割直線によって領域が2つに分割される。分割された2つの領域を領域 R_s , R_t とし、 R_s に含まれる道路ネットワークを G_s , R_t に含まれる道路ネットワークを G_t とする。その様にして道路ネットワークの分割を行う。

最後に分割した道路ネットワークに対して既存手法の道路ネットワークの変形を適用させる。その後、分割した道路ネットワークを結合させる。まず分割した道路ネットワーク G_s , G_t 各々に対して、既存手法の道路ネットワークの変形を適用させる。なお、コスト関数の計算には分割直線と交差しているエッジに関する値も計算に含める。また、本研究ではコスト関数を経路の見やすさを考慮したコスト関数 E' に変更する。経路以外エッジ同士の最小距離や経路以外エッジはユーザが通る可能性が低いと考えられる。よって、既存手法の e_{int} , e_{len} を経路上のエッジに限定し、ユーザが通る経路上のエッジ自体の距離や経路上のエッジと他のエッジの最小距離をより伸ばせる様にする。新しく作成した関数を e'_{int} , e'_{len} とする。 e'_{int} , e'_{len} , e_{rlen} , e_{ang} , e_{rang} を用いて作成されたコスト関数を E' とする。その後、 G_s , G_t を結合させる。

4 計算機実験

既存手法と提案手法をC言語で実装し、実際の道路ネットワークを用い、計算機実験を行った。ただし、既存手法で抽出する道路ネットワークは提案手法と同じにし、コスト関数 E' を用いて、道路ネットワークの変形を行う。実験環境はCPUはIntel Core i7 (2.93GHz) でMemoryは8.00GBである。また、道路ネットワークのコスト関数 E' の重みの設定は表1に示す。また描画する領域は70mm×70mmとし、 d_{int} は2mmとし、 d_{len} は6.25mmとする。

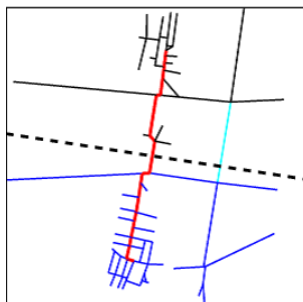


図1 抽出した道路ネットワークに対して簡略化を行った地図



図2 既存手法を適用させた地図 (コスト関数 E')

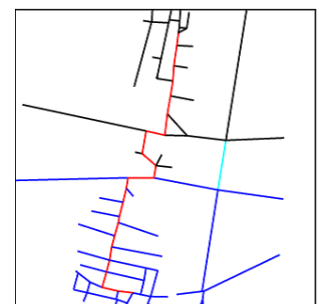


図3 提案手法を適用させた地図 (コスト関数 E')

表1 道路ネットワークの重み設定

	λ_{int}	λ_{len}	λ_{rlen}	λ_{ang}	λ_{rang}
Stage 1	1000	2000	0.1	1000	0
Stage 2	1000	2000	0.1	5000	500

図1は道路ネットワーク抽出、簡略化を行った地図、図2は既存手法を用いた地図、図3は提案手法を用いた地図である。

図1の点線は分割直線で、分割直線より上部の領域を R_s 、下部の領域を R_t とする。また、太いエッジは経路である。

既存手法、提案手法に対して経路上のエッジの不足している長さ（経路上のエッジ長さが d_{len} 以下で、 d_{len} と経路上のエッジ長さの差）の合計を表2に示す。また、既存手法、提案手法の角度のずれの平均と最大も表2に示す。表2の経路上のエッジの不足している長さの合計において、既存手法より提案手法の方が短いため、提案手法の方が経路上のエッジを伸ばし、見やすい略地図が生成できたといえる。また、既存手法より提案手法の方が角度のずれの平均、最大が小さいため、角度を保ち、大きなずれもなく、見やすい略地図を生成できたといえる。

また、今回、各領域に d_{len} 以上のエッジを含んだ地図を用いて実験した。その結果、提案手法では十分長い経路上のエッジを短くし、それ以外エッジを伸ばす事ができた。

表2 計算機実験の数値結果

	経路上のエッジの不足している長さの合計	角度のずれ	
		平均	最大
既存手法	72.44mm	3.61度	37.57度
提案手法	56.67mm	2.44度	16.52度

5 結論と今後の課題

本研究では[1]の手法を基に、経路の見やすさを考慮した略地図生成手法を提案した。その結果、分割した各領域に十分な長さ (d_{len}) 以上のエッジを含んだ場合、既存手法より提案手法の方が見やすい略地図を生成できた。

今後の課題としては、地図の特徴（領域内のエッジ数や道路ネットワークの構成）から、重みの決定方法を明らかにする事が考えられる。

参考文献

- [1] Johannes Kopf, Maneesh Agrawala, David Salesin, David Barger and Michael F. Cohen, "Automatic Generation of Destination Maps," In *Proceedings ACM SIGGRAPH Asia 2010*, pp. 1–12, 2010.