Vol.2012-MPS-90 No.12
2012/9/19

文字列の動的配置手法

宮本　健[†1]　虻川　雅浩[†1]

カーナビゲーションシステムやスマートフォンなどに搭載している地図表示アプリケーションには，特定の位置から道路に沿って道路名称を表示する機能がある．特定の位置が固定位置の場合，道路が表示画面内であっても，固定位置が表示画面外であれば，名称が非表示になる．可能な限り多くの道路名称を表示するため，表示画面に応じて動的に道路名称の位置を変更する方法がある．本研究では，道路名称の可読性向上を目的とし，従来の方法と比べて，道路名称の位置変更回数，道路名称の重なり数を削減した文字列の動的配置手法を提案する．

A Dynamic Label Layout Method

KEN MIYAMOTO[†1]　MASAHIRO ABUKAWA[†1]

Map applications on car navigation system or smart phone have a function to display labels of road name. Labels are displayed from positions along roads. For displaying a lot of labels, a method to change label positions by a map region on a display was proposed. This method can't change optimal positions because the evaluation of readability is not performed. For making recognition of labels easier, we propose a method to change label positions found by minimizing a evaluation function. The function is composed of 2 parameters which are the distance from the center on a display and collisions among labels.

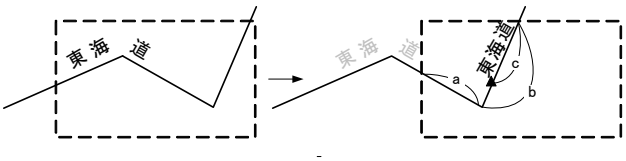
1. はじめに

カーナビゲーションシステム(以下，カーナビ)には，固定の位置から道路に沿って道路名称を表示する機能(図 1)がある．固定の位置がカーナビ表示画面外の場合，道路名称に対応する道路が画面内でも名称が非表示になる．表示画面内に存在する道路の名称を可能な限り表示するため，画面に表示する地図範囲(以下，表示範囲)に応じ，動的に道路名称の位置を変更する方法(以下，動的配置)を検討する．

従来法として，図 2 に示すように，表示画面で表示する範囲外になった道路名称を，表示範囲内に存在する道路の中心に移動する方法[1]が提案されている．[1]は変更後の位置の相応しさを評価していないため，道路名称同士の重なり (図 3)や道路名称の変更過多(図 4)が起こる．文字列の重なりを削減する方法としては[2][3]が提案されているが，組み合わせ最適化手法を用いているために計算速度が遅く，リアルタイムに配置位置の変更をする動的配置へのインプリメントには向かない．

よって，本研究では，[1]の方法と比べて道路名称同士の重なりと道路名称の位置変更回数を低減する他，リアルタイム処理に耐えうるアルゴリズムの検討をする．

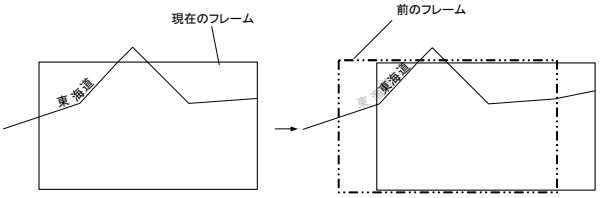
図 1 道路に沿って道路名称を表示する機能
(四角：文字列の配置を開始する固定位置，線：道路)



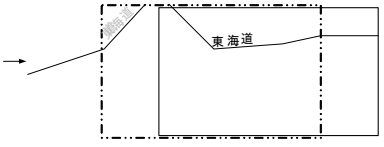
(a) 時刻 t の表示範囲と文字列の関係　　(b) 時刻 t+ α の表示範囲と文字列の関係

図 2 従来の動的配置手法
(三角：a+b=c×2 を満たす点，点線四角：表示画面)

図 3 道路名称同士の重なり



(a) 時刻 t の表示範囲と文字列の関係　　(b) 時刻 t+ α の表示範囲と文字列の関係



(c) 時刻 t+ β の表示範囲と文字列の関係(α < β)

図 4 道路名称の配置変更過多
(表示画面の境界付近に再配置された(b)ため，わずかな表示画面の移動に伴い再々配置される(c))

†1 三菱電機(株)　情報技術総合研究所
Mitsubishi Electric Corporation Information Technology R&D Center

2. 文字列の動的配置方法

[1]のように配置位置の相応しさを評価しない場合，他の道路名称との重なりや，表示画面の境界付近への配置が起こる．道路名称を適切な位置に配置するため，配置の相応しさを評価する関数を定義し，その関数が最小となる位置に配置する．

関数を最小化(もしくは最大化)する方法として，大きく分けて2つの方法がある．1つは，[2][3]で用いている Genetic Algorithm や Simulated Annealing のように，複数の組み合わせの中から，最も関数の値が低い組み合わせを選択する方法である．道路名称の数を x 個，道路名称各々が配置可能な位置を y 個とすると，x^y 個の配置組み合わせの中から最適な 1 つを探索する方法である．

2 つ目は，動的計画法 [4]である．この方法は，解決すべき問題を複数の部分問題に分割し，これらを逐次的に解く方法である．道路名称 1 つにつき，y 個の位置を評価するため，計 x×y 個の位置を評価する。

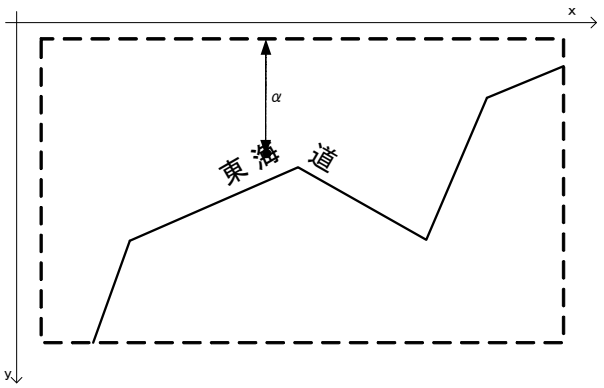
2 つの方法を比べると，多項式時間で計算が完了する後者の方が，リアルタイム処理へのインプリメントに向いている．そこで，本研究では，動的計画法を用いて道路名称の位置を決める．疑似コードを以下に示す.

○　文字列の動的配置
・表示範囲外の文字列を抽出
For n=1:表示範囲外の文字列数
　・道路に沿って配置候補を複数作成
　・最小値=MAX
　For m=1:候補数
　　・候補 m から道路に沿って文字列 n を配置
　　・文字列評価関数の値を算出(3 章に記載)
　　If 最小値 > 文字列評価関数の値
　　・最小値=文字列評価関数の値
　　・文字列の配置を保存
　　End
　End
　・保存した文字列の表示
End

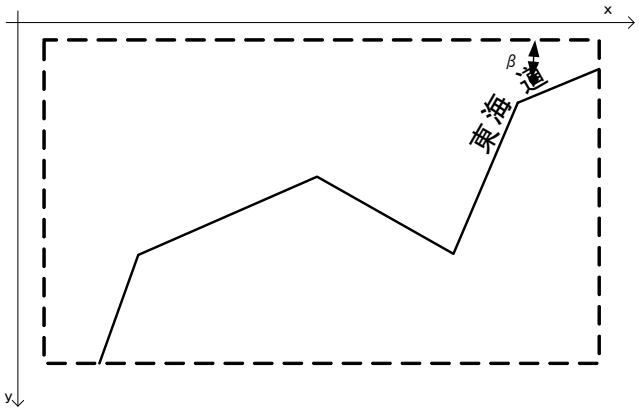
3. 文字列評価関数

道路名称同士の重なりと位置変更回数低減を目的とした評価関数を本節で述べる．

まず，道路名称の位置変更回数低減について議論する．道路名称の位置変更回数低減のため，表示画面中心への文字列再配置をする．何故なら，表示画面の中心に再配置すれば，数式 1 に示す MinDist が増加し，再び文字列が表示画面外になるまでにスクロールする距離が増加するからである．なお，数式 1 中の $dist_{Left}$，$dist_{Top}$，$dist_{Right}$，$dist_{Bottom}$ は，それぞれ，表示範囲の左辺，上辺，右辺，下辺から文字列までの距離を示す.



(a) y 軸正方向に距離 α 分スクロールすると，
再配置の対象になる



(b) y 軸正方向に距離 β 分スクロールすると，
再配置の対象になる

図 5 表示範囲と文字列との距離

数式 1 表示画面と文字列との距離

$$MinDist = \arg\min\{dist_{Left}, dist_{Top}, dist_{Right}, dist_{Bottom}\}$$

上述のように，表示画面の中心を最適な配置位置としたが，スクロール方向によっては，必ずしも表示画面の中心が最適とは限らない．スクロール方向を予測できれば，道路名称の位置変更回数を，さらに削減できると考える．本研究では，動きの推定が困難な車の移動により，表示範囲が変化するアプリでの使用を想定している．そのため，車の移動方向により最適位置を決めず，表示画面の中心を最適位置とした.

図 5 に配置位置の比較を示す. 図 5(a)では y 軸正方向に距離 α (MinDist= α), (b)では距離 β (MinDist= β)表示範囲が動けば，道路名称が再配置の対象となる．このように，最悪のケースを想定すると，道路名称の表示画面中心付近への配置が相応しい.

次に，道路名称同士の重なりについて議論する．[2][3]の評価指標である道路名称同士の重なり数を使用すると，道路名称の重なりは防げるが，密集を避けることができない．そこで，本研究では，道路名称同士の密集度を評価指標として用いることで，道路名称同士の重なりの他，密集回避も試みる．

このことから，以下 2 つの評価指標を反映した文字列評価関数を検討する．

① 表示範囲と文字列の位置関係
② 文字列の密集度

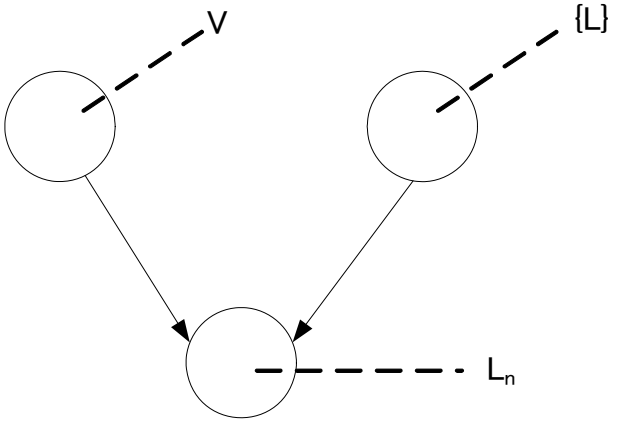


図 6 評価指標①②により，n 番目の文字列の配置位置が決まる事を示すモデル

評価指標①②を noisy-or model[5]でモデル化すると，図 6 となる．図 6 の V は表示範囲を示し，{L}は既に配置が決まった文字列，L_n は配置検討対象である n 番目の文字列を示す． 2 章の疑似コードに示すように，n は表示画面外の文字列各々に振るインデックスを示す． n 番目の文字列の配置位置が，既に配置した文字列と表示範囲に依存するモデルが，図 6 である．仮定したモデルより，文字列の評価関数は数式 2 で示すことができる．数式 2 の $P(L_n^0|V, \{L\})$ は，V と{L}が与えられたとき，n 番目の文字列 L_n の配置位置が不適切な確率を示す．また，$P(L_n^0|V)$，$P(L_n^0|\{L\})$は，それぞれ，V が与えられたときに L_n の配置位置が不適切な確率，{L}が与えられたときに L_n の配置位置が不適切な確率を示す．noisy-or model を仮定しているため，V と{L}が与えられたとき，n 番目の文字列 L_n の配置位置が適切な確率 $P(L_n^1|V, \{L\})$は，$P(L_n^1|V, \{L\})=1- P(L_n^0|V, \{L\})$と示すことができる．本研究では，配置位置が不適切な確率 $P(L_n^0|V, \{L\})$を，動的計画法で最小化することで，文字列の配置最適化を図る．

数式 2 文字列の評価関数

$$P\left(L_n^0 \mid V,\{L\}\right)= P(L_n^0 \mid V)P(L_n^0 \mid \{L\})$$

3.1 表示画面と文字列の位置関係

数式 2 中の表示画面と文字列の位置関係を示す項 $P(L_n^0|V)$を，数式 3 で定義する．観測データ$(x_1, x_2, \cdots, x_m)$各々が独立的に得られたと仮定し，結合確率を計算する式が，数式 3 である．

数式 3 中の C_n は n 番目の文字列の文字数，$dist_{jm}$ は文字 m の位置 x_m と表示画面の一辺との距離を示す．また，α_{jm} は表示画面の一辺と文字 m の位置関係によって決まるパラメータであり，数式 4 で定義する．Z は正規化のための定数である．数式 3 中，4 つのシグモイド関数の和は，文字 m が表示画面の中心に近い程，低い値を示す．

数式 3 表示画面と文字列の関係

$$P(L_n^0 \mid V) = \prod_{m=1}^{C_n} \frac{\sum_{j=1}^{4} Sigmoid(dist_{jm}, \alpha_{jm})}{Z}$$

$$Sigmoid(a,b) = \frac{1}{1+\exp(-ab)}$$

数式 4 数式 3 で用いるパラメータ α_{jm} の定義

$$\begin{cases} \alpha_{1m} = 1(\text{左辺の左側に文字}m\text{が存在}) \\ \alpha_{1m} = -1(\text{左辺の右側に文字}m\text{が存在}) \end{cases}$$

$$\begin{cases} \alpha_{2m} = 1(\text{上辺の上側に文字}m\text{が存在}) \\ \alpha_{2m} = -1(\text{上辺の下側に文字}m\text{が存在}) \end{cases}$$

$$\begin{cases} \alpha_{3m} = 1(\text{右辺の右側に文字}m\text{が存在}) \\ \alpha_{3m} = -1(\text{右辺の左側に文字}m\text{が存在}) \end{cases}$$

$$\begin{cases} \alpha_{4m} = 1(\text{下辺の下側に文字}m\text{が存在}) \\ \alpha_{4m} = -1(\text{下辺の上側に文字}m\text{が存在}) \end{cases}$$

3.2 文字列の密集度

数式 2 中の文字列同士の密集を示す項 $P(L_n^0|\{L\})$を，数式 5 で定義する．3.1 と同様に，観測データ$(x_1, x_2, \cdots, x_m)$が独立的に得られたと仮定して結合確率を計算する．

数式 5 中の x_j, x_m は文字 j, m の中心座標を示す．また，*C'*は既に配置が決まった文字列に含まれる文字数を示し，Σは共分散行列，右肩文字の T は転置を示す．*Z'*は正規化のための定数である．数式 5 中，ガウス関数の和は，文字列が密集する領域で高い値を示す．

数式 5 文字列の密集度

$$P(L_n^0 \mid \{L\}) = \prod_{m=1}^{C_n} \frac{\sum_{j=1}^{C'} Gauss(x_j - x_m)}{Z'}$$

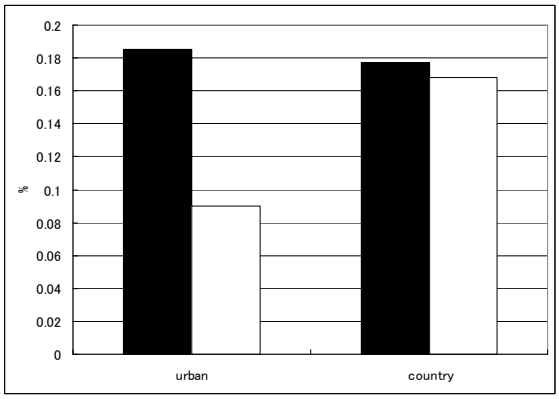
$$Gauss(a) = \exp(-\frac{a^T \Sigma^{-1} a}{2})$$

4. 検証

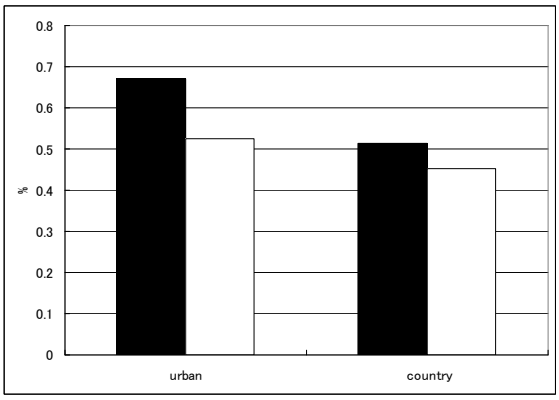
任意出発地から目的地迄スクロールしたときの道路名称の重なりと配置変更回数を検証する．1 フレーム毎に他と重なる道路名称と，配置を変更した道路名称の数をカウントすることで検証する．検証に用いたデータは，カーナビ用地図データに含まれるパリ市街地と郊外の道路名称である．このうち，パリ市街地にはのべ 1600 個，郊外にはのべ 1500 個の道路名称が含まれている．ここでの『のべ』とは，複数フレームに渡って同じ文字列が存在する場合，フレーム数分文字列数をカウントしている．また，検証で使用する重なり数は，複数フレームに渡って文字列の重なりがある場合，フレーム数分カウントしている．

また，本研究で用いたカーナビ表示画面のサイズは，800×480 であり，数式 5 中のΣは diag(400, 400)を用いた．diag(a, b)は 1 行 1 列目が a, 2 行 2 列目が b の対角行列を示す．

検証結果を図 7 に示す．図 7 に示すように，[1]の方法に比べて提案法は，道路名称の重なり率，位置変更割合共に削減している．中でも，市街地では，重なり率が 50%，位置変更割合が 22%減少した．これは，郊外に比べて，道路名称の位置が頻繁に移動することで，重なりを避ける位置に道路名称が移動したためと考えられる．一方，郊外における道路名称の重なり率，位置変更割合は，都市部に比べて削減していない．これは，道路名称の位置変更が都市部に比べて頻繁に発生しないため，重なりがある状態が複数フレームに渡って発生したためと考えられる．



(a) 道路名称の重なり率
(縦軸：道路名称重なり率)



(b) 道路名称の位置変更割合
(縦軸：位置変更が発生する道路名称の割合)

図 7 検証結果
(黒線：[1]，白線：提案法)

5. おわりに

本稿では，表示画面の範囲と文字列の位置関係と文字列の密集度を示す文字列を評価する関数を用いて，文字列の位置をカーナビ画面が表示する地図範囲の変化に応じて，最適な位置に文字列を変更する方法を提案した．結果，従来の方法[1]と比較して，文字列の重なりと文字列の更新回数を各々，25%，16%削減した．これは，[1]で実施していない再配置位置の可読性評価を，提案法でしているためと考える．

計算速度に関して本稿では述べていないが，道路名称の可読性評価をしているために[1]と比べて増加していると考えられる．

今後，計算速度に関する検討と，本研究では未評価である道路名称の密集度評価をする予定である．

参考文献

1 柳田, “特開 2005-077428”
2 Shawn Edmondson: A General Cartographic Labeling Algorithm, The International Journal for Geographic Information and Geovisualization, Volume 33, Number4/Winter 1996
3 Steven van Dijk: Towards an Evaluation of Quality for Name Placement Methods, International Journal of Geographical Information Science, Volume 16, Issue 7 November 2002, page 641 – 661
4 金谷健一: これなら分かる最適化数学，共立出版株式会社 (2006)
5 Daphne Koller, Nir Friedman: Probabilistic Graphical Models, The MIT Press(2009)