

## カーネル密度推定を用いた 文字列の最適配置手法

宮本 健<sup>†</sup> 田中 昭二<sup>†</sup>

カーナビゲーションシステムには、文字列の配置を開始する位置から、道路線に沿って、道路名称を表示する文字列配置機能がある。文字列配置機能で用いる文字列の配置を開始する位置は、文字列の可読性を保つため、①文字列同士の重なり数、②道路線の角度変化、③文字列の密集度をパラメータとした評価関数が最小となる位置に設定する必要がある。そこで、本稿では、まず、①②③を示す目的関数を提案する。その後、提案する目的関数を動的計画法に組み込み、有効性を検証する。検証の結果、従来の目的関数を使用した場合と比較して、文字列同士の重なり数、文字列の密集度が90%以上低減した。

### An Optimal Label Layout Method using Kernel Density Estimation

Ken Miyamoto<sup>†</sup> and Shoji Tanaka<sup>†</sup>

In order to make labels which show street names along the street lines on a Car Navigation Map be easily readable, a method for finding optimum label position is proposed. This method introduces a function which takes into account of the three parameters, 1: a number of collisions to other labels, 2: angle-variant of a polyline of a street, 3: density of labels. It then finds the optimum label position by minimizing the function. In our experiment, we reveal that a number of collisions and density of labels are reduced by more than 90% compared to minimizing a function of conventional method.

### 1. はじめに

カーナビゲーションシステムには、図1に示すように、文字列の配置を開始する位置(以下、文字列配置開始点)から、道路を示すリンク(以下、道路リンク)に沿って、道路名称を表示する文字列配置機能がある。文字列配置機能で用いる文字列配置開始点は、文字列の可読性を保つため、文字列を配置した時に、以下3つが小さくなる位置を設定する必要がある。

- ① 文字列同士の重なり数
- ② 道路リンクの角度変化
- ③ 文字列の密集

①と②を低減するように文字列配置開始点を最適化する方法として、[1]が提案されている。[1]は、①と②を示す目的関数を Simulated Annealing で最小化することで、文字列配置開始点を最適化する方法である。

[1]の目的関数は、①と②を示す目的関数であるため、最小化しても③を低減できない。そのため、①②③を示す目的関数を開発した。以下では、まず、2章で①②③を示す目的関数を提案する。その後、3章で提案する目的関数を動的計画法に組み込み、有効性を検証する。

### 2. カーネル密度推定を用いた目的関数の定義

1章で述べた3つのパラメータ①②③を示す目的関数  $J_n$  を(1)式で定義した。(1)式中の  $L_n$  は最適化対象の  $n$  番目の文字列(以下、文字列  $n$ )を示し、 $\{L_{n-1}\}$  は既に配置位置が決まった  $0$  番目から  $n-1$  番目の文字列の集合を示す。 $R_n$  は文字列  $n$  が沿う道路リンクを示す。なお、右辺の  $P(L_n | \{L_{n-1}\}, R_n)$  は、 $\{L_{n-1}\}, R_n$  が与えられた時の  $L_n$  の確率を示す。

$$J_n = P(L_n | \{L_{n-1}\}, R_n) \quad (1)$$

(1)式の右辺は、事後確率とみなすことができ、(2)式に示すように、ベイズの定理[2]を用いて尤度  $P(R_n | L_n)$  と事前確率  $P(L_n | \{L_{n-1}\})$  に分解できる。

$$P(L_n | \{L_{n-1}\}, R_n) \propto P(R_n | L_n) P(L_n | \{L_{n-1}\}) \quad (2)$$

以下では、まず、2.1で尤度の算出方法を述べた後、2.2で事前確率の算出方法を述べる。

<sup>†</sup> 三菱電機株式会社デザイン研究所  
Mitsubishi Electric Corporation Industrial Design Center

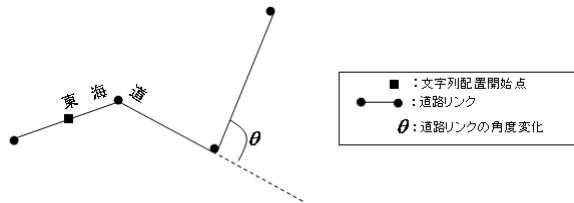


図 1 文字列配置機能

### 2.1 尤度

尤度  $P(R_n | L_n)$  は, (3)式で定義した. (3)式中の  $No_n$  は  $n$  番目の文字列が沿う道路リンク  $R_n$  のノード数,  $C_n$  は文字列  $n$  の文字数を示す.  $x_i, x_j$  は, それぞれ,  $i$  番目のノードの位置,  $j$  番目の文字の中心位置を示すベクトルである. また,  $O_i$  は道路リンクの角度変化が大きいく程, 大きな値を示すパラメータであり, カーネル関数  $K(u, A)$  は(4)式で定義する. なお,  $Z$  は正規化のための定数,  $\Sigma$  は共分散行列を示す.

$$P(R_n | L_n) = \frac{\sum_{j=0}^{C_n} \sum_{i=0}^{No_n} O_i K(x_i - x_j, \Sigma)}{Z} \quad (3)$$

$$K(u, A) = \exp\left(-\frac{u^T A^{-1} u}{2}\right) \quad (4)$$

### 2.2 事前確率

事前確率  $P(L_n | \{L_{n-1}\})$  は, (5)式で定義した. (5)式中の  $C_m$  は  $m$  番目の文字列の文字数,  $O'$  は尤度と事前確率のバランスを調整するための定数,  $Z'$  は正規化のための定数,  $\Sigma'$  は共分散行列を示す.

$$P(L_n | \{L_{n+1}\}) = \frac{\sum_{j=0}^{C_n} \sum_{m=0}^{n-1} \sum_{p=0}^{C_m} O' K(x_p - x_j, \Sigma')}{Z'} \quad (5)$$

## 3. 検証

### 3.1 検証方法

提案した目的関数の有効性を示すため, カーナビ用地図データの文字列 10 万個の文字列配置開始点を, 動的計画法で最適化した. 具体的な方法を, 以下の疑似コードに沿って説明する. まず, 道路リンク上に複数の候補点を作成する. 次に, 候補点か

ら道路リンクに沿って文字列を配置する. その後, 目的関数を計算する. 目的関数の値が最小の場合, 候補点を文字列配置開始点として登録する. 結果を比較するため, 動的計画法で使用する目的関数は,  $J_n$  と, [1]の目的関数の 2つを使用した.

また, 評価方法は, 1 章に示すパラメータ①②③の比較である. なお, 3 つのパラメータのうち, 文字列の密集は近接する文字列の数で評価した.

- ・ 文字列配置開始点の最適化 疑似コード

For  $n=0$  to  $n<$ 文字列数

- ・ 候補点の作成

For  $q=0$  to  $q<$ 候補点数

- ・ 候補点  $q$  から道路リンクに沿って文字列を配置

- ・ 目的関数の計算

If 目的関数の値が最小

- ・ 候補点  $q$  を文字列  $n$  の文字列配置開始点に設定

End

End

End

### 3.2 検証結果

検証の結果を, 表 1 に示す. 表 1 に示すように, 文字列の重なり数, 近接する文字列の数がそれぞれ, 92%, 99%低減した. 道路リンクの平均角度変化は, 大差ない結果となった.

表 1 目的関数の検証結果

	[1]の目的関数	$J_n$
①文字列同士の重なり数(個)	12	1
②道路リンクの平均角度変化(rad)	$4.5 \times 10^{-3}$	$4.7 \times 10^{-3}$
③文字列の密集(個) (近接する文字列の数)	346	4

## 4. おわりに

本稿では, 文字列同士の重なり, 文字列が沿う道路の角度変化と文字列の密集を示す目的関数を提案した. 検証の結果, 提案した目的関数を用いることで, 文字列同士の重なり数と, 文字列の密集を示す近接する文字列数が 90%以上低減した. 今後, グラフィカルモデルを用いて文字列同士の関係を表現し, 確率伝搬法[3]を用いることで, 文字列同士の重なり数や近接する文字列の数を, さらに, 削減する予定である.

**謝辞** アルゴリズムの開発にあたり，助言を頂いた皆様に，謹んで感謝の意を表す  
る．

### 参考文献

- [1] Shawn E, “A General Cartographic Labeling Algorithm”, International Journal for Geographic Information and Geovisualization, Vol133 No4
- [2] Christopher M.Bishop : Pattern Recognition And Machine Learning 2<sup>nd</sup> printing edition (2007)
- [3] Felzenszwalb.P. F, “Efficient Belief Propagation for Early Vision”, CVPR(2004) 261-268