2007－AL－112 (9)
2007／5／11

地図上の経路探索におけるラベルの更新問題

山本優二† 今井桂子††

概要

近年，カーナビゲーションシステム等のように，現在地から目的地までの経路探索を動的に行なうような電子地図が存在する．しかし，現在のカーナビゲーションシステムでは経路を探索すると，経路と文字情報であるラベルが重なることがある．このような場合，ラベルが読めなくなってしまうため，ラベルを適切な位置に移動する必要がある．そこで，本研究では，経路と重なったラベルを移動するという問題を考える．この問題に対し，ラベルを対話的な時間で更新するアルゴリズムを提案する．また，そのアルゴリズムを実装し，計算機実験の結果を示す．

Label updating problems for routes in digital maps

Yuji Yamamoto† and Keiko Imai††

Abstract

Recently, there are many kinds of digital maps. We can use some digital maps to search a route from present place to a destination such as car navigation systems. In such a case, labels (or names) of sites might overlaps the route in the map and it hard to get the information about the route or sites. For solving this problem, each label overlapping the route should be moved to an appropriate place and/or scaled down. We formulate this problem as a label updating problem for a route. We present an algorithm for the problem and experimental results are also shown.

1. 序　　論

ラベル配置問題とは，地図上のそれぞれのノードやエッジに対して，文字情報 (ラベル) を適切な位置に読みやすい大きさで配置する問題である．近年，地理情報システム（GIS）における重要な課題の１つとして，デジタル化された地図へ自動的にラベルを配置する研究が行なわれている ([4] に主な論文リストがある)．GIS の急速な発達と普及に伴い，地理情報の伝達方法は多様化し，インターネットや携帯情報端末を利用した道案内のための情報提供も行なわれるようになってきた．また，このような道案内のための情報端末として，カーナビゲーションシステムがある．

カーナビゲーションシステムは，ラベル配置問題によってラベルを配置した地図を用いて，現在地から目的地までの経路を動的に探索するのに用いられる．しかし，現在のカーナビゲーションシステムでは，探索した経路や自分の現在地点をあらわす車体とラベルが重なることがある．ラベルが経路や車体と重なった場合，ラベルが読めなくなってしまうため，ラベルを適切な位置に更新し，見やすくする必要性がある．ラベル配置問題に対する既存のアルゴリズムでは地図上のすべての点に対して，ラベル配置を行なってしまうため，地図上の点数が多い場合多くの時間を必要とする．カーナビゲーションシステムでは，経路を探索するたびにラベルを更新する必要があるため，短時間でラベルを配置しなおす必要がある．そこで，ラベル全体を配置しなおすのではなく，一部のラベルのみ配置しなおすラベルの更新問題というものが研究されてきた．

その研究として，Rostamabadi と Ghodsi が文献 [1], [2], [3] で，地図上に未知のパス上を動く質問点があった場合，その質問点とラベルが重ならないようにラベルを更新するアルゴリズムを提案した．

彼らの研究は質問点を避けるラベルの更新問題だったため，経路とラベルが重なっている場合はラベルを更新することができなかった．そこで，本研究では，ラベルと経路が重なった場合，経路を避けるようにラベルを対話的な時間で更新するアルゴリズムを提案する．

† 中央大学大学院 理工学研究科 情報工学専攻
Information and System Engineering Course, Graduate School of Science and Engineering, CHUO University
†† 中央大学 理工学部 情報工学科
Department of Information and System Engineering, Faculty of Science and Engineering, CHUO University

本研究では, x 軸, y 軸に平行で, 高さが一定の長方形ラベルを用いる. また, 更新の命令には移動と縮小を用いる. 本研究の目的は, 経路を避けるように, ラベルを可能な限り大きなラベルサイズで配置することである.

2. 前処理

点 $\mathcal{P} = \{p_1, p_2, \ldots, p_n\}$ と, 地図上の x 軸, y 軸に平行な高さの一定なラベル $\mathcal{L} = \{l_1, l_2, \ldots, l_n\}$ が配置されたラベルの集合 L が与えられる. また, L の各ラベルは, ラベルのいずれかの頂点と点が一致する 4-Position Model を用いて配置されたラベルとする.

各点 p_i は 4-Position Model でラベルを配置しているため, 図 1 のように 4 つのラベル配置候補に分けることができる. このラベル配置候補を, それぞれ $n_{i1}, n_{i2}, n_{i3}, n_{i4}$ と記す. また, ここで後の処理を簡単にするため, それぞれのラベル配置候補に仮頂点 $v_{i1}, v_{i2}, v_{i3}, v_{i4}$ を挿入する (図 2). ここで, ラベル l_i が配置されているラベル配置候補の領域を $o(n_{ik})$, それ以外のラベル配置候補の領域を $f(n_{ik})$ と記す.

また, ラベルを更新する命令として, 本研究では移動と縮小を考えている. そこで, 縮小したラベルの表記を $r(n_{ik}, \gamma_{ik})$ とする. ここで, γ_{ik} は, 元のラベルサイズを 1 としたときの縮小したラベルサイズの比率とする.

2.1 コンフリクトグラフ

コンフリクトグラフ $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ は, ラベルの位置とその重なりを頂点集合 $\mathcal{V}$ とエッジ集合 $\mathcal{E}$ であらわした重みつきグラフである. コンフリクトグラフのエッジの集合 $\mathcal{E}$ は, DominoEdge と BanishEdge の 2 種類のエッジで構成される.

ここで, $o(n_{ik})$ と $f(n_{jl})$ が交差しているものとする. $o(n_{ik})$, $f(n_{jl})$ に対応する挿入した仮頂点をそれぞれ v_{ik}, v_{jl} とする. ここで $i = j$ の場合, 有向エッジ (v_{ik}, v_{il}) は DominoEdge に属する. $i \neq j$ の場合, 有向エッジ (v_{jl}, v_{ik}) が DominoEdge に属する. また, $f(n_{ik})$ と $f(n_{jl})$ が交差していて, それぞれに対応する仮頂点をそれぞれ v_{ik}, v_{jl} とすると, 無向エッジ $\{v_{ik}, v_{jl}\}$ は BanishEdge に属する. また, ここで DominoEdge に属するエッジを有向エッジとし, BanishEdge に属するエッジを無向エッジとする. エッジの集合は次のように書くことができる.

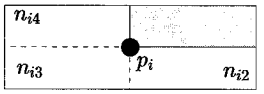


図 1 ラベル配置候補

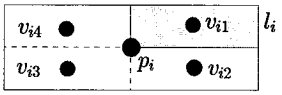


図 2 仮頂点の挿入

$\mathcal{E}$ = DominoEdge ∪ BanishEdge,
DominoEdge = $\{(v_{ik}, v_{jl}) | o(n_{ik}) \cap f(n_{jl}) \neq \phi, i = j\} \cup \{(v_{jl}, v_{ik}) | o(n_{ik}) \cap f(n_{jl}) \neq \phi, i \neq j\}$,
BanishEdge = $\{\{v_{ik}, v_{jl}\} | f(n_{ik}) \cap f(n_{jl}) \neq \phi)\}$.

初期ラベルとそれに対応するコンフリクトグラフをそれぞれ図 3, 図 4 (a), (b) に示す. 図 4 の頂点は挿入した仮頂点, 灰色のラベル配置候補は $o(n_{ik})$ を示している. DominoEdge は実線で, BanishEdge は破線であらわされている.

2.2 重みの計算

本研究ではラベルの更新の命令として, 縮小を許している. そのため, それぞれのラベル配置候補がどれだけのラベルサイズでラベルを配置できるか計算し, 対応する仮頂点に記憶する.

まず, $\mathcal{G}$ におけるラベル配置候補の重みを定義する. ラベル配置候補 n_{ik}, n_{jl} が重なっているものとする. それぞれのラベル配置候補に対して, 縮小させたラベル $r(n_{ik}, \gamma_{ik}), r(n_{jl}, \gamma_{jl})$ が重ならないような値 γ_{jk}, γ_{jl} を決める. 本研究の目的として, 可能な限りラベルサイズを大きくしたいため, $\min\{\gamma_{ik}, \gamma_{jl}\}$ が最大となるように, 値 γ_{jk}, γ_{jl} を決定する. この値を $g(n_{ik}, n_{jl})$ として定義する. この関数 g を用いて, ラベル配置候補に対応する仮頂点の重み $w(v_{ik})$ を次のように定義する.

$$w(v_{ik}) = \min\{\{g(n_{ik}, n_{jl})\} \cup \{g(n_{jl}, n_{ik})\} \cup \{1.0\} \mid (v_{ik}, v_{jl}), (v_{jl}, v_{ik}) \in \text{DominoEdge}\}$$

2.3 リスクの計算

次にラベル配置候補に対応する仮頂点のリスクについて定義する. これは, それぞれのラベル配置候補にラベルを配置しようとした際, どれだけ縮小する可能性があるかを示すものである.

あるラベル配置候補 n_{ik} について考える. n_{ik} が他のラベル配置候補と重ならない場合, そのラベル配置

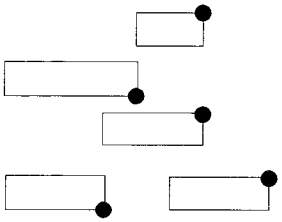
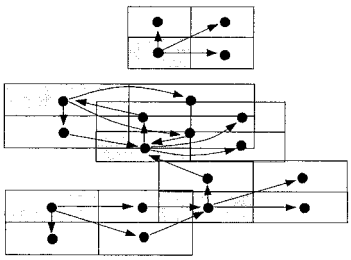
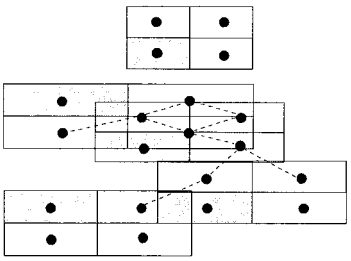


図 3 初期ラベル



(a) DominoEdge



(b) BanishEdge

図 4 コンフリクトグラフ

候補にラベルを配置しても他のラベルに影響を与えない．そのため，n_{ik} にラベルを配置すれば，ラベルを縮小することなくラベルを配置することができる．

次に，n_{ik} に対して，他のラベル配置候補と重なる場合，その候補にラベルを配置すると重なっているラベルに影響し，どちらかのラベルが縮小される可能性がある．

最後に，n_{ik} の内部に他のノード p_j が含まれる場合，n_{ik} にラベルを元のラベルサイズで配置すると内部に含まれるノード p_j と重なってしまうため，ラベルを縮小して配置する必要性がある．以上のことから，それぞれのラベル候補位置に対応する仮頂点のリスク r の値を次のように定義する．

$$r(v_{ik}) = \begin{cases} 0 & n_{ik} \text{ がどのラベル配置候補と} \\ & \quad \text{重ならない場合} \\ 1 & n_{ik} \text{ が他のラベル配置候補と} \\ & \quad \text{重なる場合} \\ 2 & n_{ik} \text{ のラベル配置候補内に他の} \\ & \quad \text{ノード } p_j \text{ が含まれている場合} \end{cases}$$

前処理ではコンフリクトグラフを作成するのに平面走査法を用いる．よって，コンフリクトグラフの作成は $O(n \log n)$ 時間で行なうことができる．また，重みやリスクの計算は交差しているラベル配置候補がわかっていれば $O(1)$ 時間で行なうことができる．よって，前処理では $O(n \log n)$ 時間で処理することができる．

3. 更新処理

現在地から目的地までの経路を探索した結果，経路と最初の入力で与えたラベル $\mathcal{L}$ が重なることがある (図 5 (a))．このように経路とラベルが重なると，ラベルが見づらくなってしまうためラベルを移動や縮小する必要がある．

3.1 重みとリスクの値の変更

まず，経路と重なっているラベル配置候補に対応する仮頂点の重みとリスクの値を変更する必要がある．経路と重なっているラベル配置候補を探し出し，重なっているすべてのラベル配置候補に対して，経路に重ならず，ラベルサイズが最大となるような重みを計算する．また，経路と重なっているラベル配置候補は，ラベルを配置する場合，縮小する必要があるため，経路が重なったラベル配置候補に対応する仮頂点のリスクの値を 2 にする．

ここで，重みとリスクの値を変更する必要があるラベルに対応する仮頂点を要更新ノードとし，要更新ノードキューに記憶する．要更新ノードとなった仮頂点は，経路か他のラベルが重なっていることになるため，ラベルを移動，縮小する必要がある．よって，経路と重なっている $o(n_{ik})$ は，対応する仮頂点 v_{ik} を要更新ノードキューに記憶する．

3.2 update(v_α)

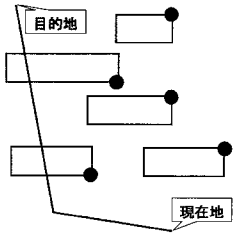
要更新ノードキューにノードが入っている場合，ラベルを更新する処理 update(v_α) を行なう．ここで，要更新ノードキューの先頭に入っているノードを v_α とする．まず，ノード v_α からリスクの値が最も小さい仮頂点 (リスクの値が同じ場合は重みの大きな仮頂点) を終点に持つ DominoEdge をたどっていく．Domi-

noEdge が接続していない, または, 1 度通った仮頂点にたどりつく, または, リスクの値が 2 となる仮頂点にたどりついたら, それまでにたどってきた仮頂点の中で一番大きな重みの値を $W_{\max}$ とする (図 6 (a)). また, そのパスを P とする.

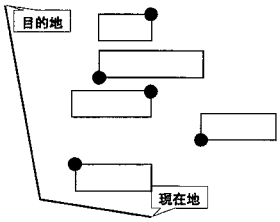
次に, v_α から重み $W_{\max}$ を持つ仮頂点にたどり着くまで, ラベルを移動, 縮小の命令のどちらか一方, もしくは両方を行ない (図 6 (b)), それぞれのラベルに対応する点 $p_i \in \mathcal{P}$ のフラグの値を 1 にする. ラベルを移動, 縮小の命令のどちらか一方, もしくは両方を行なったら, その命令を行なったラベルと重なるラベル配置候補に対応する仮頂点の重みとリスクの値を変更する. ここで, $o(n_{ik})$ が移動, 縮小の命令のどちらか一方, もしくは両方を行なったラベルと重なり, n_{ik} に対応する点 p_i のフラグの値が 0 の場合は仮頂点 v_{ik} を要更新ノードキューに記憶する. この動作を要更新ノードキュー内のノードがなくなるまで繰り返す.

図 5 (a) の更新後の結果は図 5 (b) のようになる.

すべてのラベルを更新し終わるのにかかる手間は, k をラベルの更新命令を行なった数とすると, 最大のラベルサイズを探索するのに $O(k)$ 時間, ラベルを更新するのに $O(k)$ 時間を必要とする. また, 1 度ラベルを更新した点 $p_i \in \mathcal{P}$ は, 2 度更新することはないため, ラベルを更新し終わるのにかかる時間は, 全体で $O(k)$ 時間必要とする. ここで, $k < n$ である.

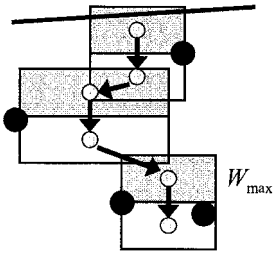


(a) ラベルと経路が重なる例

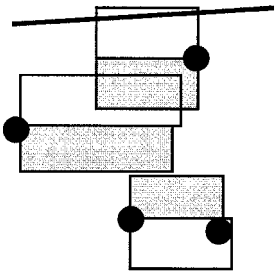


(b) ラベルの更新後

図 5 ラベルの更新例



(a) 更新前



(b) 更新後

図 6 $W_{\max}$ までラベルを移動・縮小

4. アルゴリズム

以上のことから, アルゴリズムは, 前処理と更新処理の 2 つに分けることができる. 前処理は, データが入力されたときに行なわれ, また, 更新処理は, 経路が見つかった後に行なわれる. それぞれの処理におけるアルゴリズムをまとめたものを図 7 と 図 8 に示す.

1. 入力されたすべての点 $\mathcal{P}$ に対し, ラベル配置候補位置 n_{ik} に対応する仮頂点 v_{ik} を挿入する.
2. コンフリクトグラフ $\mathcal{G}$ を構築する.
3. それぞれのラベル配置候補位置 n_{ik} に対応する仮頂点 v_{ik} に対して, 重みとリスクの値を計算し, 記憶する.

図 7 前処理におけるアルゴリズム

1. 経路に重なっているすべてのラベル配置候補位置に対して, 重みとリスクの値を更新する.
2. 経路と重なっている $o(n_{ik})$ に対応する仮頂点 v_{ik} をすべて要更新ノードキューに挿入する.
3. もし, 要更新ノードキューにノードがない場合, 終了する.
4. 要更新ノードキューからノードがなくなるまで, ノードを取り出し, **update**(v_α) を行なう.

図 8 対話処理におけるアルゴリズム

5. 計算機実験

前節の手法を実装し, ランダムに生成したデータに対して計算機実験を行なった. 実験を行った計算機はOSが Windows XP Media Center Edition, CPUが Intel Core 2 Duo プロセッサー E6300 1.86GHz, メモリが 2.00GB である. 入力データは点数 333 点, 文字数を 2 文字から 5 文字の中でランダムに生成し, 任意の場所にラベルを配置したものを与えた.

初期入力のラベルを図 9 に, 更新後のラベルを図 10 に示す. 図 9 では, 四角の点が入力時に与えた点の集合, 破線が入力する経路, 実線の枠線がラベル, 破線の枠線が経路と重なっているラベルである. また, 図 10 では, 実線の線分が入力した経路, 破線の枠線が更新により位置が変わったラベル, 塗りつぶしたラベルが更新により縮小されたラベルである. 図 10 から, 経路を避けるようにして, ラベルが配置できていることが見てとれる.

また, 図 10 の結果を表 1 の 1 行目に, そして, 他の経路に変更した結果を表 1 の 2, 3 行目に示す. ここで, 表 1 の最小比率とは, 元のラベルサイズを 100 としたときにおける, 一番小さく縮小したラベルサイズの比率である. また, 前処理時間とは, データが入力されてからコンフリクトグラフを作成し, 経路を入力するまでの時間であり, 更新時間とは, 経路が決定されてから, ラベルを更新し終わるまでにかかった時間である (描画の時間は含まれていない). 表 1 の更新時間から, 対話的な時間でラベルを更新していることが見てとれる.

表 1 前処理にかかった時間と更新にかかった時間

初期点数	最小比率	前処理時間	更新時間
333	71.4 (%)	20.52 (ms)	3.7409 (ms)
333	69.7 (%)	20.07 (ms)	3.7126 (ms)
333	65.0 (%)	20.33 (ms)	3.5433 (ms)

更新時間は経路がラベルと重なる数に関わらず 3.6 ミリ秒前後で更新することができた. ここで, 経路と重なるラベル配置候補を探索する時間を見ると, 今回の実験では線形探索を行なっているので約 2.9 ミリ秒であった. このことから, 経路と重なるラベルを探索する時間を短縮することで, より対話的な時間でラベルを更新できると考えられる.

6. 結　　論

本研究では, 経路と重なるラベルを対話的に更新する手法として,

- 前処理で, コンフリクトグラフを作成し, 重みやリスクの値を計算する.
- 更新処理で, 要更新ノードキューからノードがなくなるまで, update(v_α) を行なう.

といった 2 つの処理を行ない, 前処理に $O(n \log n)$ 時間, $O(n)$ の領域を必要とし, また, ラベルを更新し終わるのに $O(k)$ 時間を必要とする手法を提案した. ここで, n は地図上の点の数, k は更新における移動と縮小の命令数である.

前処理で, コンフリクトグラフを構築し, どのラベル配置候補と重なるかを調べておくことで, 更新の時間を短縮することができた. また, 計算機実験により提案手法の動作を評価したところ, 実際に対話的な時間でラベルが更新できていることがわかった.

今回の実験では, 経路と重なるラベルを探索するため線形探索を用いている. しかし, これでは重なったラベルを見つけるのに $O(n)$ の時間を必要とする. 本研究の目的として, 対話的にラベルを更新する必要があるため, $O(n)$ より少ない時間で重なったラベルを探索することが今後の課題としてあげられる.

また, 今回提案したアルゴリズムでは, 更新したラベルのラベルサイズが必要以上に小さくなることがあるため, より元の大きさに近づけたラベルサイズにラベルを更新することを考える必要がある.

謝辞　本研究の一部は科学研究費補助金の援助を受けて行なったものである.

参 考 文 献

[1] Farshad Rostamabadi and Mohammad Ghodsi, "A fast algorithm for a labeling to avoid a moving point," In *Proceedings of the 16th Canadian Conference on Computational Geometry* (*CCCG'04*), pp. 204–208, 2004.

[2] Farshad Rostamabadi and Mohammad Ghodsi, "An efficient algorithm for label updating in 2PM model to avoid a moving object," In *Proceedings of the 21st European Workshop on Computational Geometry* (*EWCG'05*), pp. 131–134, 2005.

[3] Farshad Rostamabadi and Mohammad Ghodsi, "Label updating to avoid point-shaped obstacles in fixed model," *Theoretical Computer Science*, Elsevier, Volume 369, Issues 1-3, pp. 197–210, 2006.

[4] Alexander Wolff, "The Map-Labeling Bibliography," 2007. http://i11www.iti.uni-karlsruhe.de/awolff/map-labeling/bibliography/

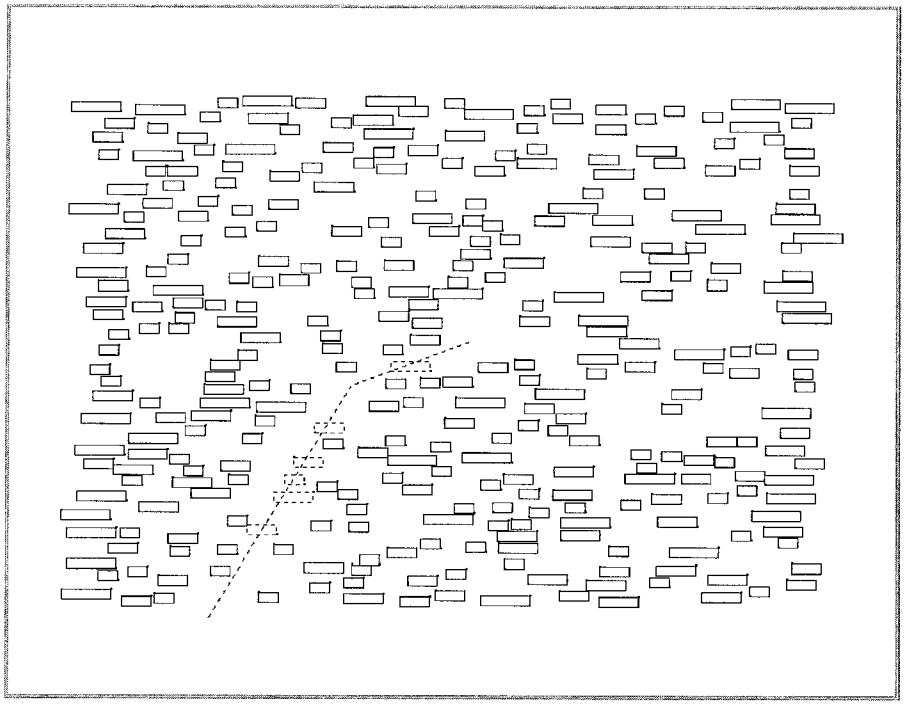


図 9　初期入力におけるラベル配置

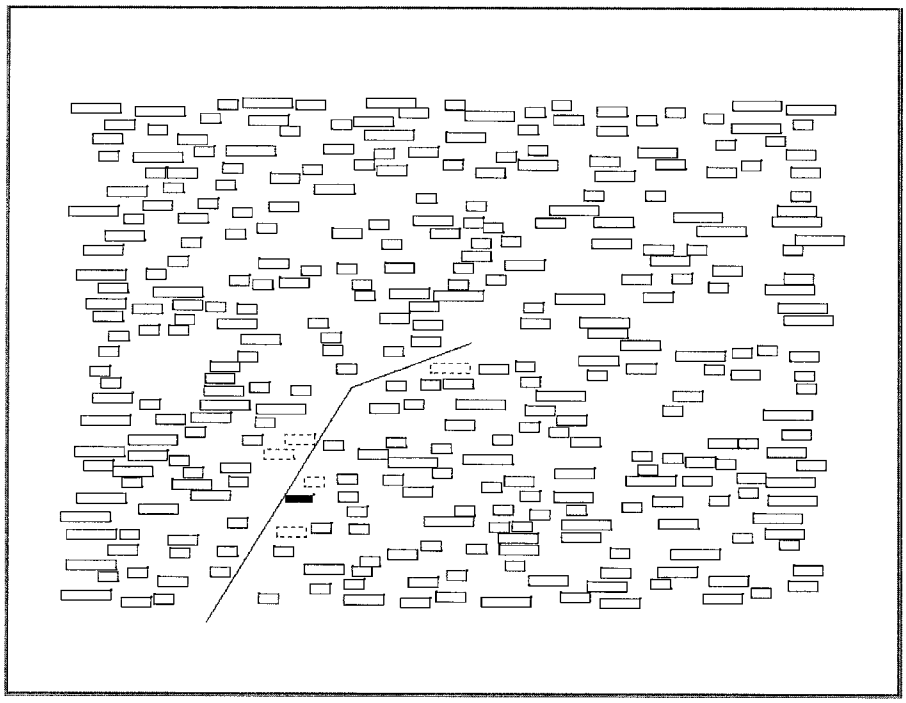


図 10　更新後におけるラベル配置