

(19) 日本国特許庁(JP)

(12) 公開特許公報(A)

(11) 特許出願公開番号

特開2010-96603

(P2010-96603A)

(43) 公開日 平成22年4月30日(2010.4.30)

(51) Int.Cl.	F I	テーマコード (参考)
G 0 1 C 21/00 (2006.01)	G 0 1 C 21/00 G	2 C 0 3 2
G 0 8 G 1/0969 (2006.01)	G 0 8 G 1/0969	2 F 1 2 9
G 0 9 B 29/10 (2006.01)	G 0 9 B 29/10 A	5 H 1 8 0
		5 H 1 8 1

審査請求 未請求 請求項の数 8 O L (全 15 頁)

(21) 出願番号	特願2008-267041 (P2008-267041)	(71) 出願人	000100768
(22) 出願日	平成20年10月16日 (2008.10.16)		アイシン・エイ・ダブリュ株式会社
			愛知県安城市藤井町高根10番地
		(74) 代理人	100092495
			弁理士 蛭川 昌信
		(74) 代理人	100088041
			弁理士 阿部 龍吉
		(74) 代理人	100139103
			弁理士 小山 卓志
		(74) 代理人	100139114
			弁理士 田中 貞嗣
		(72) 発明者	細井 保輝
			愛知県岡崎市岡町原山6番地18 アイシン・エイ・ダブリュ株式会社内

最終頁に続く

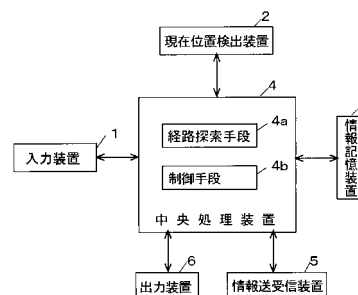
(54) 【発明の名称】 ナビゲーション装置およびプログラム

(57) 【要約】

【課題】経路探索で得られる経路の品質を維持しつつ、経路探索の所要時間を短縮化を図る。

【解決手段】出発地から目的地まで経路探索するナビゲーション装置において、少なくとも1つの階層の地図データを除いた階層構造を有する地図データを格納した情報記憶手段(3)と、前記情報記憶手段から階層構造の地図データを読み出し、出発地側、目的地側それぞれ所定領域内を経路探索し、探索領域の重なった範囲で最短コスト経路を取得する経路探索手段(4a)と、経路探索手段の探索を制御する制御手段(4b)とを備え、前記経路探索手段は、直近の上位階層の地図データが存在しない階層の探索において前記所定領域を上位階層の探索領域に拡張して探索し、前記制御手段は、取得した最短コスト経路のコストより上位階層まで探索して取得した最短コスト経路のコストが小さいとき、最短コスト経路のコストを更新する。

【選択図】 図1



【特許請求の範囲】

【請求項 1】

出発地から目的地まで経路探索するナビゲーション装置において、
少なくとも 1 つの階層の地図データを除いた階層構造を有する地図データを格納した情報
記憶手段と、

前記情報記憶手段から階層構造の地図データを読み出し、出発地側、目的地側それぞれ所
定領域内を経路探索し、探索領域の重なった範囲で最短コスト経路を取得する経路探索手
段と、

経路探索手段の探索を制御する制御手段とを備え、

前記経路探索手段は、直近の上位階層の地図データが存在しない階層の探索において前記
所定領域を上位階層の探索領域に拡張して探索し、

前記制御手段は、取得した最短コスト経路のコストより上位階層まで探索して取得した最
短コスト経路のコストが小さいとき、最短コスト経路のコストを更新することを特徴とす
るナビゲーション装置。

【請求項 2】

前記経路探索手段は、前記所定領域を上位階層の探索領域に拡張する際に、前記所定領域
内の境界までのコストが最小になる方向に優先的に拡張することを特徴とする請求項 1 記
載のナビゲーション装置。

【請求項 3】

前記制御手段は、出発地側、または目的地側の所定領域内で探索途中の到達コストが既に
取得した最短コスト経路のコストを超えたとき、それ以降のノードに対するコスト付けを
しないように経路探索手段を制御することを特徴とする請求項 1 記載のナビゲーション装
置。

【請求項 4】

前記制御手段は、出発地側、または目的地側の所定領域の境界ノードに到達コストが設定
されていないとき、経路探索手段の経路探索を終了させることを特徴とする請求項 1 記載
のナビゲーション装置。

【請求項 5】

前記制御手段は、出発地側、目的地側の所定領域の境界ノード最小コストの合計が既に取
得した最短コスト経路のコストより大きいとき、経路探索手段の経路探索を終了させるこ
とを特徴とする請求項 1 記載のナビゲーション装置。

【請求項 6】

下位階層の境界ノードに対応する上位階層ノードの隣接ノード到達コストが既に取得した
最短コスト経路のコストを超えたとき、それ以降のノードに対するコスト付けをしないよ
うに経路探索手段を制御することを特徴とする請求項 1 記載のナビゲーション装置。

【請求項 7】

道路に関する情報を記憶した情報記憶手段と、

前記情報記憶手段に記憶された道路情報に基づいて、目的地までの経路を探索する経路探
索手段と、

前記経路探索手段により探索された経路の品質を判定し経路探索処理を制御する制御手段
とを有し、

前記道路情報は、少なくとも 1 つの階層を除いた詳細度の異なる複数階層のデータレベル
で構成され、

前記経路探索手段は、出発地側・目的地側の所定領域の範囲を詳細なデータで探索し、直
近の上位階層のデータが存在しない階層の探索において前記所定領域を前記上位階層の探
索領域に拡張して探索した後、上位の階層のデータレベルへ移行して経路探索を継続し、

前記制御手段は、出発地と目的地の間で経路が繋がったとき、上位の階層のデータレベ
ルで探索された経路と下位階層のデータレベルで探索された経路とのコストを比較し、下
位階層のデータレベルで探索された経路のコストが、上位階層のデータレベルで探索され
た経路のコストより低いと判断したとき、経路探索処理を終了することを特徴とするナビ

10

20

30

40

50

ゲーショ ン装置。

【請求項 8】

出発地から目的地まで経路探索するナビゲーショ ン装置を制御す プログラムにおいて、情報記憶手段から少なくとも 1 つの階層の地図データを除いた階層構造を有する地図データを読み出すステップ、

出発地側、目的地側それぞれ所定領域内を経路探索し、直近の上位階層の地図データが存在しない階層の探索において前記所定領域を前記上位階層の探索領域に拡張して探索し、探索領域の重なった範囲で最短コスト経路を取得するステップ、

取得した最短コスト経路のコストより上位階層まで探索して取得した最短コスト経路のコストが小さいとき、最短コスト経路のコストを更新するステップ、

をコンピュータに実行させることを特徴とするプログラム。

10

【発明の詳細な説明】

【技術分野】

【0001】

本発明は階層構造を有する地図データの容量を削減し、経路の品質を維持しつつ経路探索時間の短縮を図るようにしたナビゲーショ ン装置及びプログラムに関する。

【背景技術】

【0002】

ナビゲーショ ン装置では、経路の探索を行う際に、高速道路、有料道路、国道、主要地方道、県道、細街路等の道路種別や、右左折禁止、一方通行等の交通規制の有無、リンク長、道路の幅員、車線数等によって、それぞれ、リンク又はリンク間（ノード）に各種のコストが設定されている。そして、自車位置から目的地までの最適経路を探索する際、地図データに記憶されたリンクに沿って出発地側及び目的地側から経路を探索し、出発地側からの探索と目的地側からの探索との重なり部分において、出発地側から累積されたコストと目的地側から累積されたコストとを加算し、コスト加算値が最小になる経路を最適な誘導経路として設定している。

20

【0003】

このような経路探索の所要時間を短縮するために、階層構造を持つ地図データを使用した階層別探索手法が知られている（特許文献 1）。階層構造を持つ地図データは、下位階層は全ての道路データを有する詳細地図データ、上位の階層ほど主要な道路データのみ有する広域の地図データから構成される。階層別探索手法は、出発地と目的地の周辺は詳細な地図データを用いてコストを計算し、経路の中間地域では、より粗い地図データで概略的に計算し、両方の結果を統合して案内経路として出力する方法である。このような階層別探索手法によれば、出発地と目的地との間が遠距離の場合、中間経路を粗い地図データを使用して経路探索できるため、全経路を詳細な地図データで探索する場合に比べて、全体として経路探索に要する時間を短縮することができる。

30

【0004】

また、経路探索を行った下位階層の地図データにおいて最短コスト経路が確定した場合には、その時点で経路探索を終了し、経路探索の対象となる上位階層の地図データに対して経路探索を行うことなく、より短時間で最適経路を得る手法もある（特許文献 2）。例えば、図 13（a）において、詳細な地図データを有する階層 1 において出発地側、目的地側の所定領域内をそれぞれ探索し、出発地側の領域においては探索コスト最小の境界ノード S N 1 1、目的地側の領域においては探索コスト最小の境界ノード D N 1 1 が探索されると、これら境界ノードに該当する階層 2 のノード S N 2 1、D N 2 1 に移行し、階層 2 で経路探索を行う。階層 2 の重なり部分において出発地側からの経路と目的地側からの経路がつながると、この経路 P 1 を最適経路とする。

40

【特許文献 1】特開平 2－5 6 5 9 1 号公報

【特許文献 1】特開平 1 1－2 5 7 9 8 7 号公報

【発明の開示】

【発明が解決しようとする課題】

50

【0005】

上記従来の経路探索手法では、階層化された地図データと経路探索アルゴリズムを組み合わせる場合、経路探索の所要時間を短縮化するアルゴリズムが、経路探索で得られた経路P1の品質についても最適経路であることを保障するものではない。例えば、図13(b)において、階層1では所定領域の境界到達コストが最小でない出発地側境界ノードSN12、目的地側境界ノードDN12が存在し、これら境界ノードに対応する階層2におけるノードSN22、DN22を起点とする探索で階層2の出発地側、目的地側の所定領域における境界到達コストが最小のノードSN23、DN23が探索され、ノードSN23、DN23に対応する階層3のノードSN31、DN31を起点とする探索で経路P2が探索され、経路P2の探索コストが経路P1の探索コストより小さい可能性がある。経路P2の探索コストの方が小さければ、経路P1より品質について最適経路であることになる。すなわち、階層構造を有する地図データを使用して出発地と目的地とをつなぐ経路をできるだけ迅速に見つけるだけでは、例えば、曲がりくねった山道等を含む経路が探索される可能性があり、最小の探索コストの経路、即ち最短時間で品質が最適な経路を探索できるとは限らない。

【0006】

しかしながら、階層別探索手法で高品質な経路探索を可能にするためには地図データの階層数を増やす必要がある。例えば、図14に示すように、全ての道路データを有する詳細地図データからなる階層1、一般道路以上の道路データからなる階層2、県道以上の道路データからなる階層3、高速道路データのみ有する階層4のように、地図データの階層数を増やして出発地側、目的地側から経路探索することで非常に高品質な経路の探索が可能となるが、地図データの容量が大きくなり、SDカードのような小型の記憶媒体には、地図データの全てが収まりきらなくなってしまうという問題がある。

【課題を解決するための手段】

【0007】

本発明は上記課題を解決しようとするもので、階層別探索手法において地図データの容量を少なくするとともに、経路探索で得られる経路の品質を維持しつつ経路探索の所要時間を短縮化することを目的とする。

本発明は、出発地から目的地まで経路探索するナビゲーション装置において、少なくとも1つの階層の地図データを除いた階層構造を有する地図データを格納した情報記憶手段と、前記情報記憶手段から階層構造の地図データを読み出し、出発地側、目的地側それぞれ所定領域内を経路探索し、探索領域の重なった範囲で最短コスト経路を取得する経路探索手段と、経路探索手段の探索を制御する制御手段とを備え、前記経路探索手段は、直近の上位階層の地図データが存在しない階層の探索において前記所定領域を上位階層の探索領域に拡張して探索し、前記制御手段は、取得した最短コスト経路のコストより上位階層まで探索して取得した最短コスト経路のコストが小さいとき、最短コスト経路のコストを更新することの特徴とする。

また、本発明は、道路に関する情報を記憶した情報記憶手段と、前記情報記憶手段に記憶された道路情報に基づいて、目的地までの経路を探索する経路探索手段と、前記経路探索手段により探索された経路の品質を判定し経路探索処理を制御する制御手段とを有し、前記道路情報は、少なくとも1つの階層を除いた詳細度の異なる複数階層のデータレベルで構成され、前記経路探索手段は、出発地側・目的地側の所定領域の範囲を詳細なデータで探索し、直近の上位階層のデータが存在しない階層の探索において前記所定領域を前記上位階層の探索領域に拡張して探索した後、上位の階層のデータレベルへ移行して経路探索を継続し、前記制御手段は、出発地と目的地の間で経路がつながったとき、上位の階層のデータレベルで探索された経路と下位階層のデータレベルで探索された経路とのコストを比較し、下位階層のデータレベルで探索された経路のコストが、上位階層のデータレベルで探索された経路のコストより低いと判断したとき、経路探索処理を終了することの特徴とする。

また、本発明は、出発地から目的地まで経路探索するナビゲーション装置を制御するプログ

ラムにおいて、情報記憶手段から少なくとも1つの階層の地図データを除いた階層構造を有する地図データを読み出すステップ、出発地側、目的地側それぞれ所定領域内を経路探索し、直近の上位階層の地図データが存在しない階層の探索において前記所定領域を前記上位階層の探索領域に拡張して探索し、探索領域の重なった範囲で最短コスト経路を取得するステップ、取得した最短コスト経路のコストより上位階層まで探索して取得した最短コスト経路のコストが小さいとき、最短コスト経路のコストを更新するステップをコンピュータに実行させることを特徴とする。

【発明の効果】

【0008】

本発明は、階層化された地図データのうちの ある階層を除くことで地図データの容量を削減し、かつ経路探索で得られる経路の品質を維持しつつ経路探索の所要時間を短縮化することが可能となる。

【発明を実施するための最良の形態】

【0009】

以下、本発明の実施の形態について説明する。

図1は本実施形態に係るナビゲーション装置の例を示すブロック図である。出発地や目的地の情報を入力するキーボード、マウス、タッチパネル、操作キー等からなる入力装置1、現在位置に関する情報を検出する現在位置検出装置2、階層構造を有する地図データであって、少なくとも一つの階層の地図データを除いてデータ容量を削減した地図データ、交差点データ、経路の探索に必要なナビゲーション用データ、経路案内に必要な表示／音声の案内データ、さらに地図の表示、経路探索、音声案内等の案内を行うためのプログラム（アプリケーション及び／又はOS）等を記憶した情報記憶装置3、階層構造を有する地図データを読み出し、出発地側、目的地側それぞれ所定領域内を経路探索し、探索領域の重なった範囲で最短コスト経路を取得する経路探索手段4a、経路探索手段により探索された経路の品質をコストの比較により判定し、経路探索手段による探索処理を制御する制御手段4bを備え、ナビゲータ処理手段として地図の表示処理、経路案内に必要な表示／音声案内処理、さらにシステム全体の制御を行う中央処理装置4、車両の走行に関する情報である、例えば道路情報、交通情報を送受信したり、車両の現在位置に関する情報を検出したり、さらに現在位置に関する情報を送受信したりする情報送受信装置5、経路案内に関する情報を出力するディスプレイやスピーカその他の出力装置6から構成されている。

【0010】

図2は本実施形態の階層構造の地図データの例を説明する図である。

本実施形態の階層構造を持つ地図データは、少なくとも一つの階層の地図データを除いてデータ容量を削減したデータ構造をしており、図示の例は4階層構造で、階層2のデータが省略されている。階層1は、全ての道路データを有する詳細地図データを有しており、一般道路以上の道路データからなる階層2は除かれている。階層3は県道以上の道路データ、階層4は高速道路データのみ有している。階層2については、ブロック毎の探索用領域については定義されているものの、道路データは存在せず、そのことはデータフォーマット上で定義されていて、中央処理装置4では、経路探索に際して、探索領域の取得については階層2のデータを利用するが、道路データがないことは分かっているので中身のデータの取得処理は行わない。なお、地図データ容量を削減するためには、詳細な道路データを有しているデータ容量の大きい下位階層の地図データを除くのが効果的であるが、一方で全道路データは持っている必要がある。かかる観点から本実施形態では最下位階層より一段上の階層である階層2の地図データを省略している。階層1が全道路データを有しているので、他の階層すべてを省略して階層1のデータのみを用いて経路探索することも可能であるが、それでは探索時間や探索経路の品質に影響が生ずる。従って、データ容量を削減するためには階層構造にもよるが、最下位階層より上であってできるだけ下位に属する階層の地図データを除くのが望ましい。

【0011】

10

20

30

40

50

図 3 は本実施形態の経路探索、コスト付け処理についての概略を説明する図である。

ここでは、階層 2 の地図データを省略した 4 階層構造の地図データを用いる場合の処理手順を示している。階層構造を持つ地図データは、各階層とも所定領域（データ量の多い地域ほど狭い）の複数ブロックから構成され、下位階層は全ての道路データを有する詳細地図データ、上位の階層ほど主要な道路データのみ有する広域の地図データから構成されている。まず、階層 1 の詳細な地図データを用いて出発地側、目的地側の所定領域についてそれぞれ経路探索するが、階層 2 のデータが除かれているため階層 2 の範囲を階層 1 でカバーして探索する。本来、階層 1 で探索する領域は狭いが、階層 2 で探索する範囲まで領域を拡張して探索し（ステップ S 1、S 2）、各領域の境界ノードのコストを求めて保持し、これら両領域で探索した経路に重なりがあれば最短コスト経路としてそのコストを求める（ステップ S 3）（図の星印はコストを保持していることを示す）。なお、階層 2 には地図データはないが、領域については定義されているので階層 1 の探索結果を階層 2 の領域に適用する。次いで、階層 1 の探索で到達した各領域の境界ノードに対応する階層 3 のノードを取得し（ステップ S 4、S 5）、これら各ノードを起点として所定領域の経路探索を行って各ノードの到達コストを保持し（ステップ S 6、S 7）、階層 3 の出発地側、目的地側の領域で探索した経路に重なりがあり、つながった経路のコストが階層 1 で取得した最短コスト経路のコストより小さければ、最短コスト経路のコストを更新する（ステップ S 8）。このように、各階層で出発地側、目的地側の領域で探索し、両領域で探索した経路の重なりがあり、つながった経路のコストが下位階層で取得した最短コスト経路のコストより小さければ、最短コスト経路のコストを更新し、最上位階層 4 へ移行する（ステップ S 9、S 10）。最上位階層 4 では必ず経路はつながる筈なので、目的地側から探索を行い（ステップ S 11）、最上位階層 4 で得られた経路のコストがそれより下位階層での最短コスト経路のコストより小さければ最短コスト経路のコストを更新する（ステップ S 12）。このように最短コストの経路が求まるまで各階層での探索を継続することにより取得した経路の品質（コスト）を保持する。また、各階層においてノードの到達コストが最短コスト経路のコストより大きい場合には、以降のノードに対してはコスト付けを行わないとともに、上位階層へ移行する場合には、出発地側と目的地側の境界ノードの中で最小到達コストの和が最短コスト経路のコストより大きい場合には経路探索を終了する（詳細は後述）ことで探索時間が長くなるのを防止する。こうして、地図データの容量を削減しつつ、探索経路の品質の保持と探索時間の短縮を図ることができる。

10

20

30

【0012】

図 4 は階層 1 の探索領域を階層 2 の探索領域まで拡張する場合の説明図である。

上記したように、階層 2 の地図データは除かれているが、探索領域については定義されているので、これを利用して階層 1 の探索において階層 2 で探索する範囲まで領域を広げて探索する。図示の例で、領域 A が階層 1 で定義されている探索領域（探索コア領域）とすると、まず、探索コア領域の探索を行ってその境界ノード N d 1、N d 2、N d 3、N d 4 のコスト（境界までのコスト）を取得し、境界までのコストが最小となる方向に優先的に探索領域を拡張していく。例えば、N d 1 のコストが最小であるとする、N d 1 を起点として隣接する領域、この例では領域 D へ優先的に探索領域を拡張していく。次いで、A、D を合わせた領域を 1 つの領域とみて、この領域の境界までのコストが最小となる方向、例えば、図示の例で、N d 3 のコストが最小であれば領域 B へ優先的に探索領域を拡張し、以後、同様に順次探索領域を拡張していく。こうして、階層 2 で探索すべき範囲まで探索領域を広げてその領域の境界ノードとコストを求める。

40

【0013】

図 5 は図 2 に示したような階層 2 の地図データを省略した階層構造の地図データを用いた場合の経路探索処理の全体フローを説明する図である。

まず、出発地側の探索開始地点と目的地側の探索開始地点を設定する（ステップ S 101）。これらの地点は、例えば、図 6 に示すように、出発地 S に直近の探索対象道路上のノード S N 1（図 6（a））、目的地 D に直近の探索対象道路上のノード D N 1（図 6（b））として設定される。次いで、出発地側／目的地側の探索において探索階層を最下位層

50

とし（ステップS102）、出発地側、目的地側の一定領域内をダイクストラ法を用いて探索する（ステップS103、104）。図4で説明したように、階層1の出発地側と目的地側の一定領域（探索コア領域）内をダイクストラ法を用いて探索する。道路には、高速道路、有料道路、国道、主要地方道、県道、細街路等の道路種別や、右左折禁止、一方通行等の交通規制の有無、リンク長、道路の幅員、車線数等によって、それぞれ、リンク又はリンク間（ノード）に各種のコストが設定されており、ダイクストラ法により出発地から目的地までの経路を探索してこれらコストを加算していき最短コストの経路を求める（詳細は後述）。ステップS103、104の探索で出発地側探索領域と目的地側探索領域が重なるか否か判断し（ステップS105）、重なる場合には、探索領域の重なった範囲で最短コスト経路を算出する（ステップS106）。

10

【0014】

次いで、最短コスト経路が既に求められていて、そのコストが保存されているか否か判断し（ステップS107）、保存されている場合は、新たに算出された最短コスト経路のコストが、保存されている最短コスト経路のコストより低いかなどを判断する（ステップS108）。ステップS107で最短コスト経路が保存されていない場合、ステップS108で新たに算出された最短コスト経路のコストの方が小さい場合は、最短コスト経路のコストを更新する（ステップS109）。ステップS105で出発地側探索領域と目的地側探索領域が重ならない場合、ステップS108で新たに算出された最短コスト経路のコストの方が保持されている最短コストより大きい場合は、経路探索終了判定（詳細は後述）を行う（ステップS110）。この判定で経路探索終了と判定されたか否か判断し（ステップS111）、終了と判定されなかった場合は、直近の上位階層が存在するか否か判断する（ステップS112）。階層1では直近の上位階層（階層2）が存在しないので、現階層（階層1）の上位階層（階層2）に相当する領域に拡張して探索する（ステップS113）（詳細は後述）。次いで、ステップS113における探索の中で経路探索終了と判定されたか否か判断し（ステップS114）、経路探索終了と判定されていない場合、上位階層へのコスト付け（詳細は後述）を行い（ステップS115）、出発地側／目的地側の探索の探索階層を上位階層とする（ステップS116）。次いで、この上位階層が最上位階層か否か判断し（ステップS117）、最上位階層でない場合は、ステップS103に戻って出発地側から同様に探索処理を繰り返し、最上位階層である場合は、ステップS104に戻って目的地側から同様に探索処理を繰り返す。ステップS111、ステップS114で経路探索終了と判定された場合、経路が求められたか否か判断し（ステップS118）、経路が求められていない場合は経路探索失敗とし（ステップS120）、経路が求められている場合は経路探索結果から経路を構成（ステップS117）して経路探索処理を終了する。

20

30

【0015】

図7はダイクストラ法を用いた探索処理フローを示す図である。
探索対象の階層の出発地側／目的地側の一定領域内でコスト付け対象ノードを検索し（ステップS201）、コスト付け対象の全てのノードにコスト付けが行われたか否か判断し（ステップS202）、対象ノード全てにコスト付けが行われていない場合、コスト付け対象ノードの到達コストを取得する（ステップS203）。例えば、図8（黒丸は探索対象領域におけるコスト付け対象ノードを示す）において、コスト付け対象ノードN3が、ノードN1、N2を経由する経路上のノードでその到達コストとして300を取得する。次いで、コスト付け対象ノードに接続する全てのリンクを取得し（ステップS204）、取得した全てのリンクで探索したか否か判断する（ステップS205）。全てのリンクで探索していない場合、コスト付け対象ノードの隣接ノードの到達コストを取得する（ステップS207）。図8の例で、コスト付け対象ノードN3の隣接ノードをN4とし、他のノードNmを経由してきた経路で隣接ノードN4に到達コスト500が設定されていたとして、この値を取得する。次いで、リンクコスト／リンク間（ノード）コストをコスト付け対象ノードの到達コストに加算し（ステップS208）、加算したコストが最短コスト経路のコストを超えているか否か判断し（ステップS209）、超えていない場合はステ

40

50

ップS 2 0 8で求めたコストがステップS 2 0 7で取得した隣接ノードの到達コストを超えているか否か判断する（ステップS 2 1 0）。図8の例では、コスト付け対象ノードN 3に加算するリンクのコストLCは1 5 0であり、これを加算すると隣接ノードN 4の到達コストは4 5 0となる。一方、既に取得されている最短コスト経路のコストが7 0 0であると仮定すると、コスト付け対象ノードN 3にリンクコストLCを加算した隣接ノードN 4のコストは4 5 0で最短コスト経路のコスト7 0 0より小さく、さらに、隣接ノードN 4に既に設定されている到達コスト5 0 0より小さいので、隣接ノードN 4の到達コストを4 5 0に登録更新して、ステップS 2 0 5に戻り同様の処理を行う。

【0 0 1 6】

ステップS 2 0 9においてステップS 2 0 8で求めたコストが最短コスト経路のコストを超えている場合、ステップS 2 1 0において、ステップS 2 0 8で取得したコストが隣接ノードのコストを超えている場合は、既に設定されている隣接ノードの到達コストは維持されてステップS 2 0 5に戻り同様の処理を行う。例えば、図8の例において、コスト付け対象ノードN 3と隣接ノードN 5間のリンクコストLCが5 0 0とすると、隣接ノードN 5の到達コストは8 0 0となり、既に取得している最短コスト経路のコスト7 0 0を超えてしまうため、それ以後のコスト付けは行わず、このノードにはコスト ∞ を設定してステップS 2 0 5に戻る。他の隣接ノードN 6についてみると、コスト付け対象ノードN 3との間のリンクコストLCが1 0 0で、これを加算すると到達コストは4 0 0となり、このコストを設定する。このような処理を順次行い、ステップS 2 0 5でコスト付け対象ノードに接続する全てのリンクで探索され、ステップS 2 0 2で領域内対象ノード全てにコスト付けが行われると領域内探索処理は終了する。

【0 0 1 7】

図9は現階層を上位階層に相当する領域に拡張して探索処理するフローを説明する図である。

現階層の地図データからその上位階層に相当する範囲の複数探索領域を出発地側、目的地側について取得する（ステップS 5 0 1）。これは、図3、図4の例で説明したように除かれた上位階層の探索領域をカバーするためである。図4の例では、図5の処理フローにおけるステップS 1 0 3、1 0 4で探索コア領域（A領域）は既に取り込んでいるのでステップS 5 0 1では残りの3つの領域（B、C、D領域）を取得する。次いで、目的地側における探索対象の領域を選定する（ステップS 5 0 2）。この処理は、図4の例の場合には、B、C、D領域の選択順序を選定する処理で、前述したように、探索コア領域の境界ノードのコストの小さい隣接領域の順に選定する。次いで、選定した領域についてダイクストラ法を用いた一定領域内探索を行い（ステップS 5 0 3）、出発地側探索コア領域と目的地側探索領域が重なるか否か判断し（ステップS 5 0 4）、重なる場合には探索領域の重なった範囲で最短コストの経路を算出する（ステップS 5 0 5）。次いで、最短コスト経路のコストは既に保存されているか否か判断し（ステップS 5 0 6）、保存されている場合には、新たに算出された最短コスト経路のコストは保存している最短コスト経路のコストより低い場合、ステップS 5 0 6において最短コスト経路コストが保存されていない場合には、最短コスト経路のコストを更新する（ステップS 5 0 8）。ステップS 5 0 4で出発地側探索コア領域と目的地側探索領域が重ならない場合、ステップS 5 0 7において新たに算出された最短コスト経路のコストの方が保存されている最短コスト経路のコストより低くない場合には、経路探索終了判定を行う（ステップS 5 0 9）（詳細は後述）。なお、最下位の階層であっても、出発地と目的地が同じ探索コア領域にある場合は経路探索終了経路探索は終了する。経路探索終了と判定されない場合には（ステップS 5 1 0、N O）、目的地側複数探索領域全てにコスト付けしたか否か判断し（ステップS 5 1 1）、複数探索領域全てにコスト付けしてない場合にはステップS 5 0 2に戻って同様の処理を繰り返す。

【0 0 1 8】

ステップS 5 1 1において複数探索領域全てにコスト付けした場合は、出発地側の探索

10

20

30

40

50

対象の領域を選定する（ステップ S 5 1 2）。この処理は、ステップ S 5 0 2 と同様に、図 4 の例の場合には、B、C、D 領域の選択順序を選定する。次いで、選定した領域についてダイクストラ法を用いた一定領域内探索を行い（ステップ S 5 1 3）、出発地側探索領域と目的地側探索領域が重なるか否か判断し（ステップ S 5 1 4）、重なる場合には探索領域の重なった範囲で最短コストの経路を算出する（ステップ S 5 1 5）。次いで、最短コスト経路のコストは既に保存されているか否か判断し（ステップ S 5 1 6）、保存されている場合には、新たに算出された最短コスト経路のコストは保存している最短コスト経路のコストより低いかなど判断し（ステップ S 5 1 7）、新たに算出された最短コスト経路のコストの方が低い場合、ステップ S 5 1 6 において最短コスト経路のコストが保存されていない場合には、最短コスト経路のコストを更新する（ステップ S 5 1 8）。ステップ S 5 1 4 で出発地側探索領域と目的地側探索領域が重ならない場合、ステップ S 5 1 7 において新たに算出された最短コスト経路のコストの方が保存されている最短コスト経路のコストより低くない場合には、経路探索終了の判定を行い（ステップ S 5 1 9）、経路探索終了と判定されない場合（ステップ S 5 2 0、NO）は、出発地側複数探索領域全てにコスト付けしたか否か判断し（ステップ S 5 2 1）、複数探索領域全てにコスト付けていない場合にはステップ S 5 1 2 に戻って同様の処理を繰り返す。ステップ S 5 1 0、ステップ S 5 2 0 において経路探索終了と判定された場合には、現階層を上位階層に相当する領域に拡張しての探索を終了し、ステップ S 5 2 1 において出発地側複数探索領域全てにコスト付けした場合には、経路探索を続行（図 5 のステップ S 1 1 4、ステップ S 1 1 5 へと移行）する。

10

20

【0019】

図 10 は経路探索終了判定を説明する図である。

探索対象の階層が最上位階層か否か判断し（ステップ S 3 0 1）、最上位階層でない場合は、出発地側上位移行ノード（領域境界ノード）の中で最小到達コストノードを検索し（ステップ S 3 0 2）、出発地側上位移行ノードの 1 つでも到達コストが設定されているか否か判断する（ステップ S 3 0 3）。次いで、目的地側上位移行ノード（領域境界ノード）の中で最小到達コストノードを検索し（ステップ S 3 0 4）、目的地側上位移行ノードの 1 つでも到達コストが設定されているか否か判断する（ステップ S 3 0 5）。出発地側、目的地側とも上位移行ノードが設定されている場合、出発地側／目的地側それぞれの上位移行ノードの中での最小コストの合計が最短コスト経路のコストより小さいか否か判断し（ステップ S 3 0 6）、各上位移行ノード最小コストの合計が最短コスト経路のコストより小さいときには、経路探索を続行する。ステップ S 3 0 3、ステップ S 3 0 5 で到達コストが設定されていない場合は上位階層へ移行しての探索は行わず、ステップ S 3 0 6 で出発地側／目的地側それぞれの上位移行ノード最小コストの合計が最短コスト経路のコストより大きい場合には、上位階層で探索しても最短コスト経路は存在しないので経路探索を終了する（ステップ S 3 0 8）。

30

【0020】

次に、図 11、図 12 により上位階層へのコスト付け処理を説明する。

図 11 は上位階層へのコスト付け処理フローを示す図、図 12 は上位階層へのコスト付けの例を示す図である。

40

図 11 において、探索した階層の一定領域内に存在する全ノードを取得する（ステップ S 4 0 1）。探索した階層はダイクストラ法で一定領域内の全ノードに対してのコスト付けは完了している。これら全ノードに対して上位階層へのコスト付け判定が終了したか否か判断し（ステップ S 4 0 2）、終了していない場合、着目している該当ノードは探索した階層の一定領域内の境界に存在するノードか否か判断し（ステップ S 4 0 3）、境界ノードでない場合はステップ S 4 0 2 に戻り処理を継続する。この処理は上位階層に上げるノードは境界ノードだけとする処理である。ついで、境界ノードである場合は、該当ノードが上位階層の一定領域内（境界を含む）に存在するか否か判断し（ステップ S 4 0 4）、上位階層の一定領域内（境界を含む）に存在しない場合は探索の継続ができないのでステップ S 4 0 2 に戻って、処理を継続する。該当ノードが境界ノードであり、かつ該当ノード

50

が上位階層に存在する場合は、該当ノードに接続する全てのリンクを取得し（ステップS 405）、該当ノードに接続する全てのリンクで探索したか否か判断する（ステップS 406）。境界ノードに接続する全リンクでの探索は、探索階層での最短コスト経路の探索処理であり、上位階層へコスト付けする処理の段階において、探索階層での最短コスト経路の取得処理を並行して行っている。例えば、図12において、探索階層での探索処理により、出発地側の5つの境界ノードのコストが、それぞれ200、400、800、1100、1400であり、目的地側の5つの境界ノードのコストが300、500、700、1100、1200であるものとする、これらのノードを上位階層にそのまま上げるとともに、探索階層での5つの境界ノード間を経路探索して上位階層へのコスト付けと並行して最短コスト経路を探索する。

10

【0021】

ステップS 406で全てのリンクでの探索が終了していない場合、該当ノードの到達コストを取得し（ステップS 407）、該当ノードの上位階層ノードに接続する全てのリンクを取得する（ステップS 408）。次いで、該当ノードの上位階層ノードに接続する全てのリンクで探索したか否か判断し（ステップS 409）、全てのリンクで探索していない場合、該当ノードの上位階層ノードの隣接ノードを取得し（ステップS 410）、その到達コストを取得する（ステップS 411）。次いで、該当ノードと隣接ノード間のリンクコスト／リンク間（ノード）コストをステップS 407で取得した該当ノードの到達コストに加算し、加算したコストがこの段階で取得されている最短コスト経路のコストを超えるか否か判断する（ステップS 413）。超えていない場合は加算したコストが該当ノードの上位階層ノードの隣接ノードの到達コストを超えるか否か判断し（ステップS 414）、いずれも超えていない場合は該当ノードの上位階層ノードの隣接ノードの到達コストを登録更新して、ステップS 409に戻って同様の処理を継続する。例えば、図12の出発地側の領域において、コスト400のノードM1に着目し、ノードM1の隣接ノードM2との間のリンクコストLCが100とすると、ノードM1のコスト400にリンクコスト100を加算して隣接ノードM2の到達コスト500を取得する。この段階で取得されている最短コスト経路のコストが1000であるとする、隣接ノードM2の到達コストは最短コスト経路のコストより小さい。また、他経路からの探索で隣接ノードM2に到達コスト600が設定されていたと仮定すると、ノードM1からの探索で取得した隣接ノードM2の到達コストはそれより小さいので、隣接ノードM2の到達コストを600から500に登録更新する。こうして順次コスト付けを行って、上位階層の領域の境界ノードM3のコスト800を取得する。

20

30

【0022】

ステップS 413において、ステップS 412で算出したコスト（該当ノードの到達コストにリンクコスト／リンク間（ノード）コストを加算したコスト）が最短コスト経路のコストを超える場合は、それ以上のコスト付けは行わず、ステップS 409に戻って同様の処理を継続する。例えば、図12の出発地側領域において、到達コスト800のノードM4に着目し、その隣接ノードM5との間のリンクLCのコストが250であるとする、隣接ノードM5の到達コストは1050となって最短コスト経路のコスト1000を超えてしまうので、これ以降のノード付けは行わず、ノードM5に対してはコスト ∞ を設定する。また、ノードM6、ノードM7は上位階層への移行時は最短コスト経路のコスト1000が分かっていたため、それぞれコスト1100、1400に設定されているが、これらのノードに接続するどのノードも到達コストが最短コスト経路コストを超えるので同様にこれ以降のノード付けは行わず、それらのノードに対してはコスト ∞ を設定する。また、ステップS 414において、ステップS 412で算出したコストが隣接ノードの到達コストを超える場合もコストの登録はせずにステップS 409に戻って同様の処理を継続する。こうして、ステップS 409で上位階層ノードに接続する全てのリンクでの探索が終了し、ステップS 402で探索階層での全ノードに対して上位階層へのコスト付け判定で終了と判定されると、上位階層へのコスト付けが終了する。

40

【図面の簡単な説明】

50

【 0 0 2 3 】

【図 1】 本実施形態に係るナビゲーション装置の例を示すブロック図である。

【図 2】 本実施形態の階層構造の地図データの例を説明する図である。

【図 3】 経路探索、コスト付け処理についての概略を説明する図である。

【図 4】 階層 1 の探索領域を階層 2 の探索領域まで拡張する場合の説明図である。

【図 5】 本実施形態の経路探索処理の全体フローを説明する図である。

【図 6】 出発地側、目的地側の探索開始地点を説明する図である。

【図 7】 ダイクストラ法を用いた探索処理フローを示す図である。

【図 8】 探索階層におけるノードのコスト付けを説明する図である。

【図 9】 現階層を上位階層に相当する領域に拡張して探索処理するフローを説明する図である。 10

【図 10】 経路探索終了判定を説明する図である。

【図 11】 上位階層へのコスト付け処理フローを示す図である。

【図 12】 上位階層へのコスト付けを説明する図である。

【図 13】 階層構造を有する地図データを用いた経路探索を説明する図である。

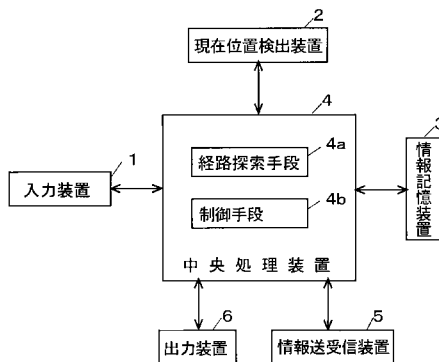
【図 14】 階層数を増やした地図データを説明する図である。

【符号の説明】

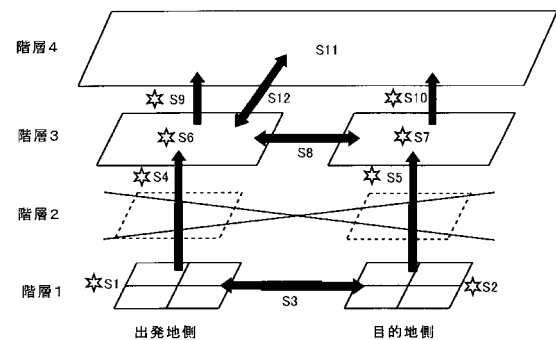
【 0 0 2 4 】

1 … 入力装置、2 … 現在位置検出装置、3 … 情報記憶装置、4 … 中央処理装置、4 a … 経路探索手段、4 b … 制御手段、5 … 情報送受信装置、6 … 出力装置。 20

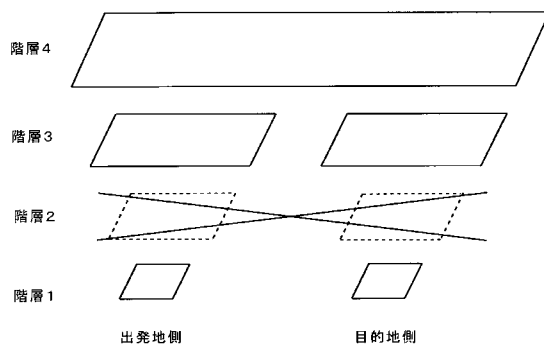
【 図 1 】



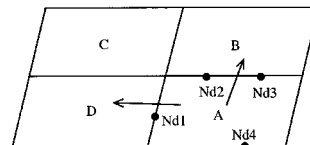
【 図 3 】



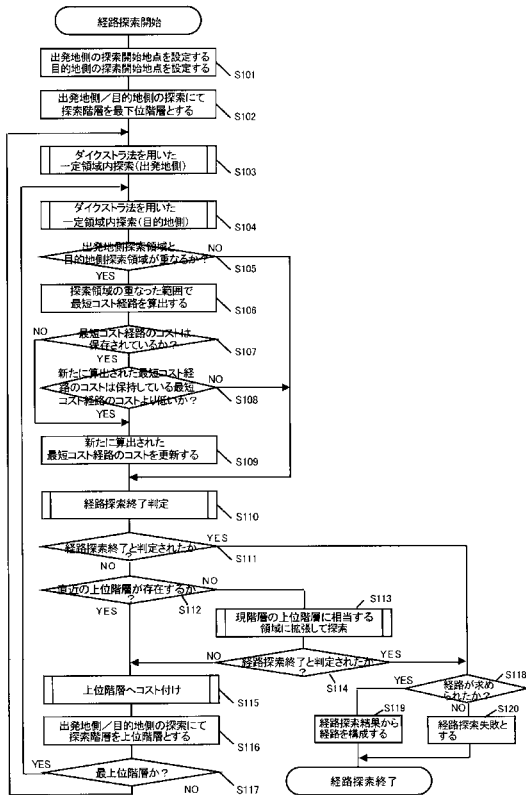
【 図 2 】



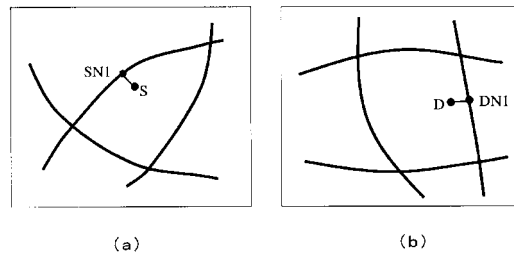
【 図 4 】



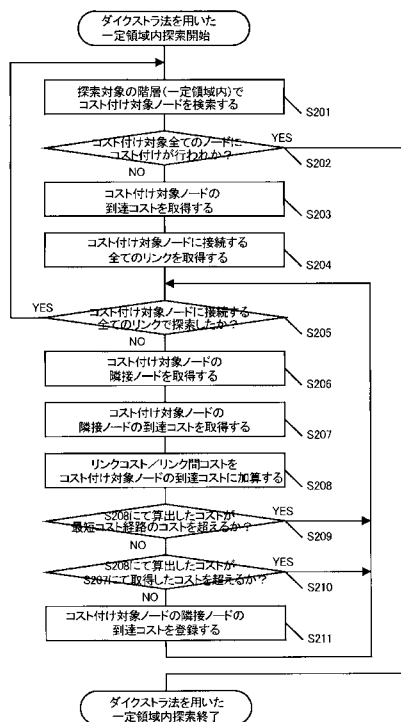
【図 5】



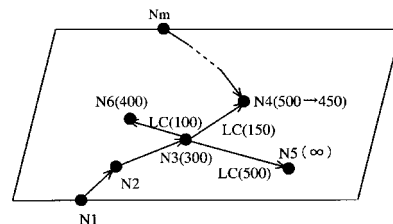
【図 6】



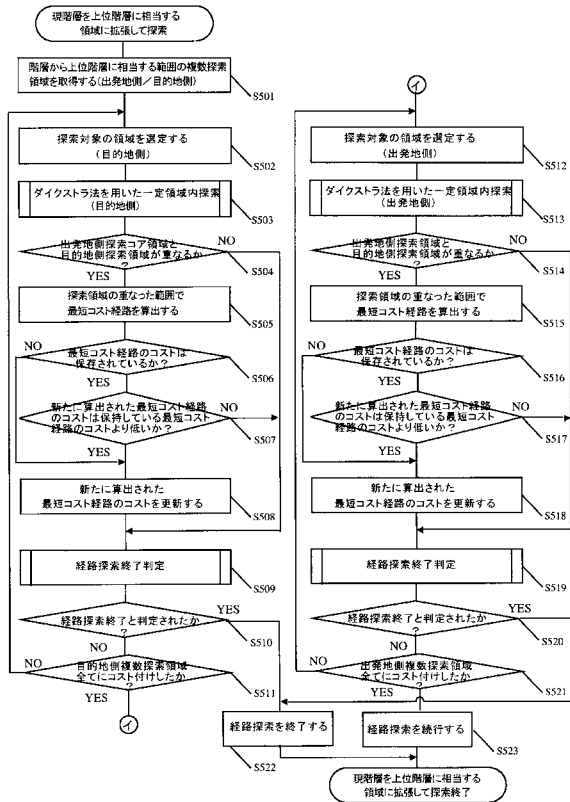
【図 7】



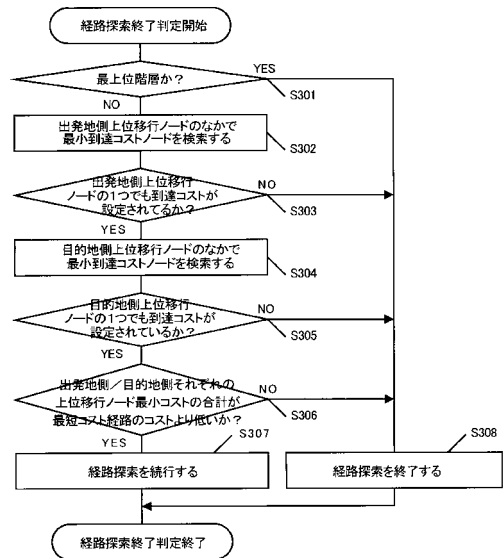
【図 8】



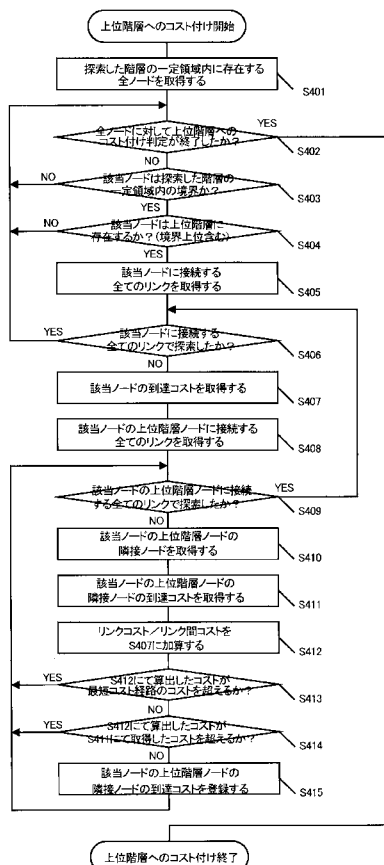
【図 9】



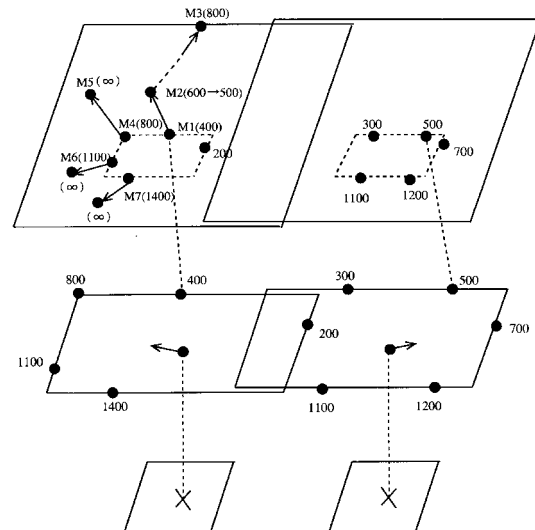
【図 10】



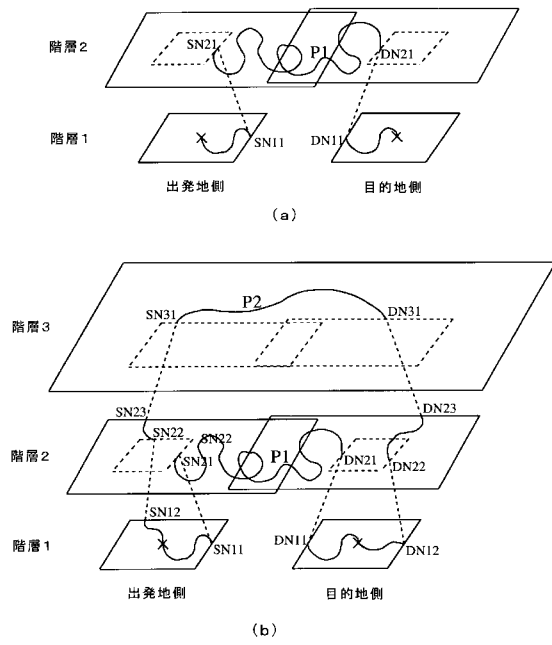
【図 11】



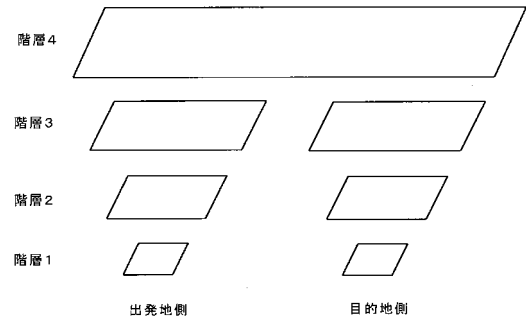
【図 12】



【図 13】



【図 14】



フロントページの続き

(72)発明者 松久 正吾

愛知県岡崎市岡町原山6番地18 アイシン・エィ・ダブリュ株式会社内

Fターム(参考) 2C032 HC14 HC31 HD07 HD16

2F129 AA03 BB03 CC16 DD03 DD05 DD21 DD30 DD32 DD62 EE02

EE43 EE52 EE57 EE62 FF04 FF07 FF41 FF52 HH02 HH04

HH12

5H180 AA01 BB13 FF04 FF05 FF11 FF12 FF14 FF22 FF25 FF27

FF33

5H181 AA01 BB13 FF04 FF05 FF11 FF12 FF14 FF22 FF25 FF27

FF33