

(19) 世界知的所有権機関
国際事務局(43) 国際公開日
2005 年 9 月 22 日 (22.09.2005)

PCT

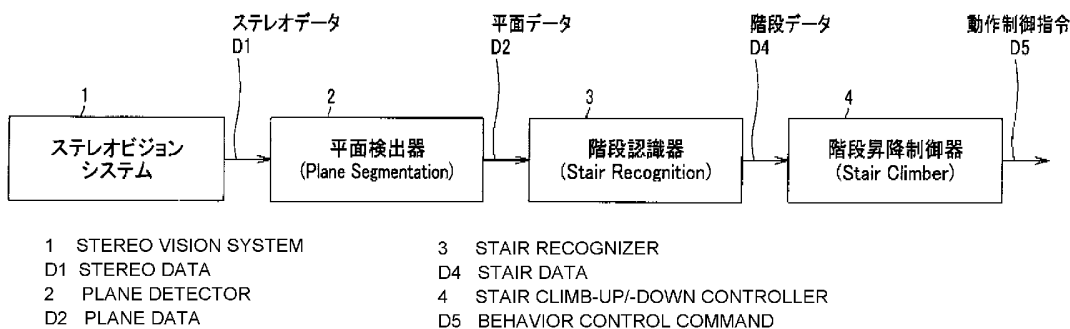
(10) 国際公開番号
WO 2005/087452 A1

- (51) 国際特許分類: B25J 13/00, 5/00 (74) 代理人: 小池 晃, 外(KOIKE, Akira et al.); 〒1000011 東京都千代田区内幸町一丁目 1 番 7 号 大和生命ビル 11 階 Tokyo (JP).
- (21) 国際出願番号: PCT/JP2005/004838
- (22) 国際出願日: 2005 年 3 月 17 日 (17.03.2005)
- (25) 国際出願の言語: 日本語
- (26) 国際公開の言語: 日本語
- (30) 優先権データ:
特願2004-077214 2004 年 3 月 17 日 (17.03.2004) JP
- (71) 出願人 (米国を除く全ての指定国について): ソニー株式会社 (SONY CORPORATION) [JP/JP]; 〒1410001 東京都品川区北品川 6 丁目 7 番 3 5 号 Tokyo (JP).
- (72) 発明者; および
- (75) 発明者/出願人 (米国についてのみ): グットマン ステフェン (GUTMANN, Steffen) [DE/JP]; 〒1410001 東京都品川区北品川 6 丁目 7 番 3 5 号 ソニー株式会社内 Tokyo (JP). 福地 正樹 (FUKUCHI, Masaki) [JP/JP]; 〒1410001 東京都品川区北品川 6 丁目 7 番 3 5 号 ソニー株式会社内 Tokyo (JP).
- (81) 指定国 (表示のない限り、全ての種類の国内保護が可能): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NA, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK, SL, SM, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, YU, ZA, ZM, ZW.
- (84) 指定国 (表示のない限り、全ての種類の広域保護が可能): ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), ユーラシア (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), ヨーロッパ (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IS, IT, LT, LU, MC, NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

/ 続葉有 /

(54) Title: ROBOT DEVICE, BEHAVIOR CONTROL METHOD FOR THE ROBOT DEVICE, AND MOVING DEVICE

(54) 発明の名称: ロボット装置、及びその動作制御方法、並びに移動装置



(57) Abstract: A robot device observes the external world by a stereo vision system (1) and outputs stereo data (D1) as a distance image, the stereo data (D1) being three-dimensional distance information calculated by a parallax of both eyes. A plane detector (2) detects planes from the distance image to recognize the planes present in the environment. From plane data (D2), a stair recognizer (3) extracts a plane which the robot can climb up and down, recognizes a stair from that plane, and outputs stair data (D4). A stair climb-up/-down controller (4) outputs a behavior control command (D5) for realizing the movement for climbing up and down the stair by using the stair data (D4). This enables a mobile body itself to obtain information on a stair to autonomously perform the movement of stair climb up and down.

(57) 要約: ロボット装置は、ステレオビジョンシステム (1) によって外界を観測し、両眼の視差によって算出される 3 次元距離情報であるステレオデータ (D1) を距離画像として出力する。平面検出器 (2) は、この距離画像から平面を検出することで、環境内に存在する複数の平面を認識する。階段認識器 (3) は、平面データ (D2) からロボット装置が昇降可能な平面を抽出し、その平面から階段を認識し階段データ (D4) を出力する。階段昇降制御器 (4) は、この階段データ (D4) を用いて階段の昇降動作を実現する動作制御指令 (D5) を出力する。これによって、移動体自身が階段に係る情報を取得して自律的に階段昇降動作を可能とする。



添付公開書類:

— 国際調査報告書

— 請求の範囲の補正の期限前の公開であり、補正書受領の際には再公開される。

2文字コード及び他の略語については、定期発行される各PCTガゼットの巻頭に掲載されている「コードと略語のガイダンスノート」を参照。

明 細 書

ロボット装置、及びその動作制御方法、並びに移動装置

技術分野

- [0001] 本発明は、例えば脚部などの移動手段を有し、複数段からなる階段の昇降動作等を可能とするロボット装置、移動装置、及びその階段昇降方法に関する。

本出願は、日本国において2004年3月17日に出願された日本特許出願番号2004-077214を基礎として優先権を主張するものであり、これらの出願は参照することにより、本出願に援用される。

背景技術

- [0002] 従来、車輪等の移動手段を有する移動装置を含むロボット装置に対し、段差のある環境や、階段の昇降動作をさせることを可能とする技術が複数開示されている。

例えば、対象物体の形状特徴点を認識するため対象物体のノードデータを有した、図1に示す既知の階段情報を用いて昇降動作を行う方法(特許第3176701号公報)や、図2に示すように、多数の接触センサ602が保護膜603で被覆されつつマトリクス状に足平601の裏面の全面的に貼り付けられた足平を利用して、階段昇降動作を行う方法がある(特許第3278467号公報)。更に、図3に示すように、足底の脇に赤外センサを装備し、階段にランドマークテープを付与することで、階段昇降動作を行う方法がある(特許第3330710号公報)。これは、図3Aに示すように、2足歩行可能なロボット装置の足平622R/Lの左右の脇部分に複数の光センサ検出部682を設け、黒色ペイントなどの光をよく吸収する塗料で描いた直線からなる所定幅の面領域であるランドマーカ680を利用することで、対になっているセンサ出力を比較することでランドマーカ680に対する相対的な方角を検出するものである。このような足平622R/Lとランドマーカ680とを使用することで、図3Bに示すように、階段690の位置を認識することが可能となる。

しかしながら、特許第3176701号公報に記載の技術においては、既知の階段情報に基づいているため、未知の環境においては適応することができなく、例えば自律型のロボット装置などに適用することが困難である。また、特許第3278467号公報及

び特許第3330710号公報に記載の技術においては、足裏もしくは足底の脇に設けられた複数のセンサを利用して階段昇降動作を行うため、着床するまで階段の情報が取得できない。したがって、遠くからの観測が不可能である。このため、ある程度目の前に階段があることが予測されるときにしか利用することができない。

発明の開示

発明が解決しようとする課題

[0003] 本発明は、このような従来の実情に鑑みて提案されたものであり、移動体自身が階段に係る情報を取得して自律的に階段昇降動作を可能とするロボット装置及び移動装置、並びにロボット装置の動作制御方法を提供することを目的とする。

上述した目的を達成するために、本発明に係るロボット装置は、移動手段により移動可能なロボット装置において、3次元の距離データから環境内に含まれる1又は複数の平面を検出し、平面情報として出力する平面検出手段と、上記平面情報から移動可能な平面を有する階段を認識し、該階段の踏面に関する踏面情報及び蹴り上げ情報を有する階段情報を出力する階段認識手段と、上記階段情報に基づき、階段昇降可能か否かを判断し、昇降動作が可能であると判断した場合には、その踏面に対して自律的に位置決めして階段昇降動作を制御する階段昇降制御手段を有することを特徴とする。

本発明においては、移動手段として例えば脚部などを備えて移動可能なロボット装置において、階段の踏面の例えば大きさや位置などに関する踏面情報から、その足底がその踏面に載せることができる大きさか否かを判断したり、階段の段差を示す蹴り上げの情報からその高さの踏面への移動が可能か否かを判断し、移動可能であると判断した場合には自律的に位置決めすることで階段を登ったり降ったりすることが可能となる。

また、上記3次元の距離データを取得する距離計測手段を有することができ、階段を検出した際には自律的に昇降動作が可能となる。

また、上記階段認識手段は、与えられた平面情報から移動可能な平面を有する階段を検出して統合前階段情報を出力する階段検出手段と、上記階段検出手段から出力される時間的に異なる複数の統合前階段情報を統計的に処理することにより統

合した統合済階段情報を上記階段情報として出力する階段統合手段とを有することができ、例えば視野が狭いロボット装置であったり、1度の検出処理ではうまく階段認識できないような場合であっても、時間的に統計的に統合した統合済階段情報とすることで、正確かつ高域な認識結果を得ることができる。

更に、上記階段検出手段は、上記平面情報に基づき踏面の大きさ及び空間的な位置を認識し、この認識結果である踏面情報を上記統合前階段情報として出力し、上記階段統合手段は、時間的に前後する踏面情報から、所定の閾値より大きい重複領域を有しかつ相対的な高さの違いが所定の閾値以下である2以上の踏面からなる踏面群を検出した場合、当該踏面群を何れをも含む一の踏面となるよう統合することができ、統合の際には統合すべきとして選択された踏面を全て含むように統合することで、広い範囲にわたって認識結果を得ることができる。

更にまた、上記階段認識手段は、上記平面情報に基づき踏面の大きさ及び空間的な位置を認識し上記踏面情報とすることができ、踏面情報は、少なくとも移動方向に対して該踏面の手前側の境界を示すフロントエッジ及び奥側の境界を示すバックエッジの情報を含むものとすることができ、踏面のフロントエッジ及びバックエッジを認識するため、例えばスパイラル形状の階段などであっても正確に踏面を認識して階段昇降動作を可能にする。

また、上記フロントエッジ及びバックエッジに挟まれた領域である安全領域の左右両側に隣接した領域であって移動可能である確率が高いと推定されるマージン領域を示す右側マージン情報及び左側マージン情報、上記平面情報に基づき踏面と推定された領域の重心を示す参照点情報、踏面となる平面を構成する点群の3次元座標情報などを有することができる。これらの踏面情報を使用することで、階段昇降動作をより正確に制御することができる。

更に、上記階段認識手段は、上記平面情報に基づき平面の境界を抽出して多角形を算出し、該多角形に基づき上記踏面情報を算出することができ、例えば、視野が狭い場合、3次元距離データの信頼性が高いような場合には、上記多角形は、上記平面情報に基づき抽出された平面の境界に外接する凸多角形領域とすることができ、実際に検出されている平面を含んだ領域とすることができ。一方、ノイズが多い

距離データなどの場合には、上記多角形は、上記平面情報に基づき抽出された平面の境界に内接する凸多角形領域とすることができ、実際に検出されている平面に内包される領域とすることで、ノイズ部分をカットして正確に踏面を検出することができる。

更にまた、現在移動中の移動面におけるバックエッジに対峙した所定位置に移動した後、昇降動作を実行するよう制御することができ、例えば蹴り上げが小さい階段など、フロントエッジとバックエッジが重なるような場合には、バックエッジを目標に移動して昇降動作することができる。同様に、フロントエッジを目標に移動して昇降動作を行ってもよい。

また、上記階段昇降制御手段は、現在移動中の移動面におけるバックエッジが確認できない場合は、次に昇降動作の対象となる次段の踏面におけるフロントエッジに対峙した所定位置に移動した後、昇降動作を実行するよう制御することができ、例えば床面を移動していて階段を検出した場合、床面のバックエッジが階段初段のフロントエッジに重ならない場合があり、そのような場合は階段の初段、すなわち昇降動作の対象となる次段の踏面のフロントエッジを目標に移動して昇降動作をすることができる。

また、上記階段昇降制御手段は、次に移動対象となる踏面を検出し、当該移動対象となる踏面に対峙した所定位置に移動する一連の動作を行って昇降動作を実行するよう制御することができ、新たな踏面に移動する毎に、踏面に対してサーチ・アライン・アプローチ動作を実行することで昇降動作を可能にする。

更に、上記階段昇降制御手段は、現在位置から次に移動対象となる次段又は次段以降の踏面が検出できない場合、過去に取得した階段情報から当該移動対象となる次段の踏面を検索することができ、予め数段上又は下の階段情報を取得しておくことで、ロボット装置が自身の直近の情報が得られないような構成で視野が狭い場合であっても昇降動作を可能とする。

また、上記階段昇降制御手段は、現在の移動面におけるバックエッジに対峙した所定位置に移動した後、次の移動対象となる踏面を検出し、当該踏面におけるフロントエッジに対峙した所定位置に移動し、当該踏面に移動する昇降動作を実行するよう

制御することができ、フロントエッジ及びバックエッジを使用することで、両エッジが平行していない螺旋状の階段であっても昇降動作を可能とする。

更に、上記昇降制御手段は、踏面に対する上記移動手段の位置を規定したパラメータを使用して昇降動作を制御することができ、このパラメータは、例えば上記脚部の足上げ高さ又は足下げ高さに基づき決定されることができる。そして、階段を登る動作と降りる動作とで上記パラメータの数値を変更するパラメータ切り替え手段を有することができ、階段を登る動作であっても、降りる動作であってもパラメータ変更するのみで同様に制御することができる。

また、上記平面検出手段は、3次元空間で同一平面上にあると推定される距離データ点群毎に線分を抽出する線分抽出手段と、上記線分抽出手段によって抽出された線分群から同一平面に属すると推定される複数の線分を抽出し該複数の線分から平面を算出する平面領域拡張手段とを有し、

上記線分抽出手段は、距離データ点の分布に応じて適応的に線分を抽出することができ、線分抽出手段は、3次元の距離データが同一平面上にある場合同一直線上に並ぶことを利用して線分を抽出するが、この際、ノイズなどの影響により距離データ点の分布に違いが生じるため、この距離データの分布に応じて適応的に線分を抽出する(Adaptive Line Fitting)ことにより、ノイズに対してロバストに、精確な線分抽出を可能とし、抽出された多数の線分から線分拡張法により平面を求めるため、ノイズの影響などにより、本来複数平面が存在するのに1つの平面としたり、1つの平面しか存在しないのに複数平面としたりすることなく精確に平面抽出することができる。

更に、上記線分抽出手段は、上記距離データ点間の距離に基づき同一平面上にあると推定される距離データ点群を抽出し、該距離データ点群における距離データ点の分布に基づき、当該距離データ点群が同一平面上にあるか否かを再度推定することができ、距離データ点の3次元空間における距離に基づき一旦距離データ点群を抽出しておき、データ点の分布に基づき再度同一平面上にあるか否かを推定することにより精確に線分抽出することができる。

更にまた、上記線分抽出手段は、上記同一平面上にあると推定される距離データ点群から第1の線分を抽出し、該距離データ点群のうち該第1の線分との距離が最も

大きい距離データ点を着目点とし、当該距離が所定の閾値以下である場合に該距離データ点群から第2の線分を抽出し、該第2の線分の一方側に距離データ点が所定の数以上連続して存在するか否かを判定し、所定の数以上連続して存在する場合に該距離データ点群を該着目点にて分割することができ、例えば抽出したデータ点群の端点を結ぶ線分を第1の線分とし、上記距離が大きい点が存在する場合には、例えば最小二乗法により第2の線分を生成し、この第2の線分において一方側に連続して複数のデータ点が存在する場合には、データ点群は例えば線分に対してジグザグな形などをとっていることが想定でき、したがって抽出したデータ点群には偏りがあると判断して、上記着目点などにてデータ点群を分割することができる。

また、上記平面領域拡張手段は、同一の平面に属すると推定される1以上の線分を選択して基準平面を算出し、該基準平面と同一平面に属すると推定される線分を該線分群から拡張用線分として検索し、該拡張用線分により該基準平面を更新すると共に該基準平面の領域を拡張する処理を繰り返し、更新が終了した平面を更新済平面として出力することができ、同一平面に属するとされる線分により平面領域拡張処理及び平面更新処理を行うことができる。

更に、上記更新済平面に属する距離データ点群において、当該更新済平面との距離が所定の閾値を超える距離データ点が存在する場合、これを除いた距離データ点群から再度平面を算出する平面再算出手段を更に有することができ、更新済平面はそれに属する全線分の平均した平面として得られているため、これから大きく外れた距離データ点を除いたデータ点群から再度平面を求めることで、よりノイズなどの影響を低減した検出結果を得ることができる。

更にまた、上記平面領域拡張手段は、線分により定まる平面と上記基準平面との誤差に基づき当該線分が該基準平面と同一平面に属するか否かを推定することができ、例えば平面方程式の2乗平均誤差などに基づきノイズの影響であるのか、異なる平面なのかを判別して更に正確に平面検出することができる。

本発明に係るロボット装置の動作制御方法は、移動手段により移動可能なロボット装置の動作制御方法において、3次元の距離データから環境内に含まれる1又は複数の平面を検出し、平面情報として出力する平面検出工程と、上記平面情報から移

動可能な平面を有する階段を認識し、該階段の踏面に関する踏面情報及び蹴り上げ情報を有する階段情報を出力する階段認識工程と、上記階段情報に基づき、階段昇降可能か否かを判断し、昇降動作が可能であると判断した場合には、その踏面に対して自律的に位置決めして階段昇降動作を制御する階段昇降制御工程とを有することを特徴とする。

本発明に係る移動装置は、移動手段により移動可能な移動装置において、3次元の距離データから環境内に含まれる1又は複数の平面を検出し、平面情報として出力する平面検出手段と、上記平面情報から移動可能な平面を有する階段を認識し、該階段の踏面に関する踏面情報及び蹴り上げ情報を有する階段情報を出力する階段認識手段と、上記階段情報に基づき、階段昇降可能か否かを判断し、昇降動作が可能であると判断した場合には、その踏面に対して自律的に位置決めして階段昇降動作を制御する階段昇降制御手段を有することを特徴とする。

本発明によれば、移動手段として例えば脚部などを備えて移動可能なロボット装置及び移動装置において、階段の踏面の例えば大きさや位置などに関する踏面情報から、その足底がその踏面に載せることができる大きさか否かを判断したり、階段の段差を示す蹴り上げの情報からその高さの踏面への移動が可能か否かを判断し、移動可能であると判断した場合には自律的に位置決めすることで階段を登ったり降ったりすることが可能となる。

本発明の更に他の目的、本発明によって得られる利点は、以下において図面を参照して説明される実施に形態から一層明らかにされるであろう。

図面の簡単な説明

- [0004] [図1]図1は、従来の昇降動作を説明する図である。
- [図2]図2は、従来の昇降動作を説明する図である。
- [図3]図3A、図3Bは、従来の昇降動作を説明する図である。
- [図4]図4は、本発明の実施の形態におけるロボット装置の概観を示す斜視図である。
- [図5]図5は、ロボット装置が具備する関節自由度構成を模式的に示す図である。
- [図6]図6は、ロボット装置の制御システム構成を示す模式図である。

[図7]図7は、ロボット装置がステレオデータから階段昇降動作を発現するまでの処理を実行するシステムを示す機能ブロック図である。

[図8]図8Aは、ロボット装置が外界を撮影している様子を示す模式図、図8Bは、ロボット装置の足底の大きさを示す図である。

[図9]図9は、階段検出を説明する図であって、図9Aは、階段を正面から見た図、図9Bは、階段を側面から見た図、図9Cは、階段を斜めから見た図である。

[図10]図10は、階段検出の他の例を示す説明する図であって、図10Aは、階段を正面から見た図、図10Bは、階段を側面から見た図、図10Cは、階段を斜めから見た図である。

[図11]図11は、図9の階段を検出した結果の一例を示す図であって、図11Aは、図9の階段を撮影した場合の画像を示す模式図、図11B乃至図11Dは、図11Aに示す画像から取得した3次元の距離データを示す図である。

[図12]図12は、図10の階段を検出した結果の一例を示す図であって、図12Aは、図10の階段を撮影した場合の画像を示す模式図、図12B乃至図12Dは、図12Aに示す画像から取得した3次元の距離データを示す図である。

[図13]図13Aは、階段を撮影した画像を示す模式図、図13Bは、図13Aから取得した3次元距離データから4つの平面領域A、B、C、Dを検出した結果を示す図である。階段を検出した結果の一例を示す図である。

[図14]図14は、階段認識器を示す機能ブロック図である。

[図15]図15は、階段検出処理の手順を示すフローチャートである。

[図16]図16A、図16Bは、多角形を示す模式図である。

[図17]図17は、Melkmanのアルゴリズムを説明するための模式図である。

[図18]図18A、図18Bは、Sklanskyのアルゴリズムにより多角形を求める方法を説明するための模式図である。

[図19]図19は、非凸多角形状の階段について発生する問題を説明するための模式図であって、図19Aは、入力される平面を示す図、図19Bは、凸包による非凸多角形状の階段の多角形表現結果を示す図である。

[図20]図20は、平滑化によって入力平面を包含する多角形を求める方法を示す模

式図であって、図20Aは、入力された平面を示す図、図20Bは、入力平面を示す多角形から不連続なギャップを除去し平滑化した多角形を示す図、図20Cは、図20Bで得られた多角形に対してラインフィッティングにより更に平滑化した多角形を示す図である。

[図21]図21は、ギャップ除去とラインフィットによる平滑化によって入力平面を包含する多角形を求める処理のプログラム例を示す図である。

[図22]図22A、図22Bは、階段パラメータの算出方法を説明するための模式図である。

[図23]図23は、最終的に認識される踏面及び階段パラメータを説明するための模式図である。

[図24]図24A、図24Bは、階段を示す模式図である。

[図25]図25は、階段統合処理の方法を示すフローチャートである。

[図26]図26は、オーバーラップしている階段データを統合する処理を説明するための模式図である。

[図27]図27は、アライン動作を説明するための図である。

[図28]図28は、アプローチ動作を説明するための模式図である。

[図29]図29は、階段昇降動作の手順を示すフローチャートである。

[図30]図30は、サーチ・アライン・アプローチ処理方法を示すフローチャートである。

[図31]図31は、昇降動作処理の方法を示すフローチャートである。

[図32]図32は、ロボット装置が認識しているか又は認識する予定の階段面を示す模式図である。

[図33]図33は、昇降動作処理の方法を示すフローチャートである。

[図34]図34は、昇降動作処理の方法を示すフローチャートである。

[図35]図35Aは、ロボット装置により認識されている踏面と足底の関係を説明するための図であり、図35Bは、各部の寸法を示す図である。

[図36]図36は、ロボット装置が昇降動作を行った様子を撮影したものをトレースした図である。

[図37]図37は、ロボット装置が昇降動作を行った様子を撮影したものをトレースした

図である。

[図38]図38は、単一の段部とロボット装置の足底の関係を示す図である。

[図39]図39は、単一の凹部とロボット装置の足底の関係を示す図である。

[図40]図40は、本変形例における平面検出装置を示す機能ブロック図である。

[図41]図41は、テクスチャを付与する手段を有しているロボット装置を説明するための図である。

[図42]図42は、本変形例における線分拡張法による平面検出方法を説明する図である。

[図43]図43は、線分拡張法による平面検出処理を示すフローチャートである。

[図44]図44は、本変形例における線分抽出部における処理の詳細を示すフローチャートである。

[図45]図45は、距離データ点の分布の様子を示す図であって、図45Aは、データの分布が線分に対してジグザグ形である場合、図45Bは、ノイズなどにより線分近傍に一樣に分布している場合を示す模式図である。

[図46]図46は、本変形例におけるZig-Zag-Shape判別方法を示すフローチャートである。

[図47]図47は、上記Zig-Zag-Shape判別処理のプログラム例を示す図である。

[図48]図48は、Zig-Zag-Shape判別処理を行う処理部を示すブロック図である。

[図49]図49は、本変形例における領域拡張処理を説明するための模式図である。

[図50]図50は、本変形例における領域拡張部における領域種を検索する処理及び領域拡張処理の手順を示すフローチャートである。

[図51]図51は、端点と直線との距離が等しくても平面方程式の2乗平均誤差rmsが異なる例を示す図であって、図51Aは、ノイズなどの影響により線分が平面からずれている場合、図51Bは、線分が属する他の平面が存在する場合を示す模式図である。

[図52]図52は、領域種の選択処理を示す図である。

[図53]図53は、領域拡張処理を示す図である。

[図54]図54Aは、ロボット装置が立った状態で床面を見下ろした際の床面を示す模

式図、図 5 4 B は、縦軸を x 、横軸を y 、各データ点の濃淡で z 軸を表現して 3 次元距離データ及び、行方向の画素列から線分抽出処理にて同一平面に存在するとされるデータ点群から直線を検出したものを示す図、図 5 4 C は、図 5 4 B に示す直線群から領域拡張処理により得られた平面領域を示す図である。

【図 5 5】図 5 5 は、床面に段差を一段置いたときの本変形例における平面検出方法と従来の平面検出方法との結果の違いを説明するための図であって、図 5 5 A は、観察された画像を示す模式図、図 5 5 B は、実験条件を示す図、図 5 5 C は、本変形例における平面検出方法により平面検出された結果を示す図、図 5 5 D は、従来の平面検出方法により平面検出された結果を示す図である。

【図 5 6】図 5 6 A は、床面を撮影した画像を示す模式図、図 5 6 B 及び図 5 6 C は、図 5 6 A に示す床面を撮影して取得した 3 次元距離データから水平方向及び垂直方向の距離データ点列から、それぞれ本変形例の線分検出により検出した線分及び従来の線分検出により検出した線分を示す図である。

発明を実施するための最良の形態

以下、本発明を適用した具体的な実施の形態について、図面を参照しながら詳細に説明する。この実施の形態は、本発明を、周囲の環境に存在する階段などの段差を認識する段差認識装置を搭載した自律的に動作可能なロボット装置に適用したものである。

本実施の形態におけるロボット装置は、ステレオビジョンなどにより得られた距離情報（距離データ）から抽出した複数平面から階段を認識し、この階段認識結果を利用して階段昇降動作を可能とするものである。

(1) ロボット装置

ここでは、まず、このようなロボット装置の一例として 2 足歩行タイプのロボット装置を例にとって説明する。このロボット装置は、住環境その他の日常生活上の様々な場面における人的活動を支援する実用ロボットであり、内部状態（怒り、悲しみ、喜び、楽しみ等）に応じて行動できるほか、人間が行う基本的な動作を表出できるエンターテインメントロボット装置である。なお、ここでは、2 足歩行型のロボット装置を例にとって説明するが、階段認識装置は、2 足歩行のロボット装置に限らず、脚式移動型のロボット装置に搭載すればロボット装置に階段昇降動作を実行させることができる。

図4は、本実施の形態におけるロボット装置の概観を示す斜視図である。図4に示すように、ロボット装置201は、体幹部ユニット202の所定の位置に頭部ユニット203が連結されると共に、左右2つの腕部ユニット204R/Lと、左右2つの脚部ユニット205R/Lが連結されて構成されている(ただし、R及びLの各々は、右及び左の各々を示す接尾辞である。以下において同じ。)

このロボット装置201が具備する関節自由度構成を図5に模式的に示す。頭部ユニット203を支持する首関節は、首関節ヨー軸101と、首関節ピッチ軸102と、首関節ロール軸103という3自由度を有している。

また、上肢を構成する各々の腕部ユニット204R/Lは、肩関節ピッチ軸107と、肩関節ロール軸108と、上腕ヨー軸109と、肘関節ピッチ軸110と、前腕ヨー軸111と、手首関節ピッチ軸112と、手首関節ロール軸113と、手部114とで構成される。手部114は、実際には、複数本の指を含む多関節・多自由度構造体である。ただし、手部114の動作は、ロボット装置201の姿勢制御や歩行制御に対する寄与や影響が少ないので、本明細書では簡単のため、ゼロ自由度と仮定する。したがって、各腕部は7自由度を有するとする。

また、体幹部ユニット202は、体幹ピッチ軸104と、体幹ロール軸105と、体幹ヨー軸106という3自由度を有する。

また、下肢を構成する各々の脚部ユニット205R/Lは、股関節ヨー軸115と、股関節ピッチ軸116と、股関節ロール軸117と、膝関節ピッチ軸118と、足首関節ピッチ軸119と、足首関節ロール軸120と、足底121とで構成される。本明細書中では、股関節ピッチ軸116と股関節ロール軸117の交点は、ロボット装置201の股関節位置を定義する。人体の足底121は、実際には多関節・多自由度の足底を含んだ構造体であるが、本明細書においては、簡単のためロボット装置201の足底は、ゼロ自由度とする。したがって、各脚部は、6自由度で構成される。

以上を総括すれば、ロボット装置201全体としては、合計で $3 + 7 \times 2 + 3 + 6 \times 2 = 32$ 自由度を有することになる。ただし、エンターテインメント向けのロボット装置201が必ずしも32自由度に限定されるわけではない。設計・制作上の制約条件や要求仕様等に応じて、自由度すなわち関節数を適宜増減することができるというまでも

ない。

上述したようなロボット装置201がもつ各自由度は、実際にはアクチュエータを用いて実装される。外観上で余分な膨らみを排してヒトの自然体形状に近似させること、2足歩行という不安定構造体に対して姿勢制御を行うこと等の要請から、アクチュエータは小型かつ軽量であることが好ましい。

このようなロボット装置は、ロボット装置全体の動作を制御する制御システムを例えば体幹部ユニット202等に備える。図6は、ロボット装置201の制御システム構成を示す模式図である。図6に示すように、制御システムは、ユーザ入力等に動的に反応して情緒判断や感情表現を司る思考制御モジュール200と、アクチュエータ350の駆動等、ロボット装置201の全身協調運動を制御する運動制御モジュール300とで構成される。

思考制御モジュール200は、情緒判断や感情表現に関する演算処理を実行するCPU (Central Processing Unit) 211や、RAM (Random Access Memory) 212、ROM (Read Only Memory) 213及び外部記憶装置 (ハード・ディスク・ドライブ等) 214等で構成され、モジュール内で自己完結した処理を行うことができる、独立駆動型の情報処理装置である。

この思考制御モジュール200は、画像入力装置251から入力される画像データや音声入力装置252から入力される音声データ等、外界からの刺激等に従って、ロボット装置201の現在の感情や意思を決定する。すなわち、上述したように、入力される画像データからユーザの表情を認識し、その情報をロボット装置201の感情や意思に反映させることで、ユーザの表情に応じた行動を発現することができる。ここで、画像入力装置251は、例えばCCD (Charge Coupled Device) カメラを複数備えており、これらのカメラにより撮像した画像から距離画像を得ることができる。また、音声入力装置252は、例えばマイクロホンを複数備えている。

思考制御モジュール200は、意思決定に基づいた動作又は行動シーケンス、すなわち四肢の運動を実行するように、運動制御モジュール300に対して指令を発行する。

一方の運動制御モジュール300は、ロボット装置201の全身協調運動を制御する

CPU311や、RAM312、ROM313及び外部記憶装置(ハード・ディスク・ドライブ等)314等で構成され、モジュール内で自己完結した処理を行うことができる独立駆動型の情報処理装置である。また、外部記憶装置314には、例えば、オフラインで算出された歩行パターンや目標とするZMP軌道、その他の行動計画を蓄積することができる。

この運動制御モジュール300には、図5に示したロボット装置201の全身に分散するそれぞれの関節自由度を実現するアクチュエータ350、対象物との距離を測定する距離計測センサ(図示せず)、体幹部ユニット202の姿勢や傾斜を計測する姿勢センサ351、左右の足底の離床又は着床を検出する接地確認センサ352、353、足底121の足底121に設けられる荷重センサ、バッテリー等の電源を管理する電源制御装置354等の各種の装置が、バス・インターフェース(I/F)310経由で接続されている。ここで、姿勢センサ351は、例えば加速度センサとジャイロ・センサの組み合わせによって構成され、接地確認センサ352、353は、近接センサ又はマイクロ・スイッチ等で構成される。

思考制御モジュール200と運動制御モジュール300は、共通のプラットフォーム上で構築され、両者間はバス・インターフェース210、310を介して相互接続されている。

運動制御モジュール300では、思考制御モジュール200から指示された行動を体現すべく、各アクチュエータ350による全身協調運動を制御する。すなわち、CPU311は、思考制御モジュール200から指示された行動に応じた動作パターンを外部記憶装置314から取り出し、又は、内部的に動作パターンを生成する。そして、CPU311は、指定された動作パターンに従って、足部運動、ZMP軌道、体幹運動、上肢運動、腰部水平位置及び高さ等を設定するとともに、これらの設定内容に従った動作を指示する指令値を各アクチュエータ350に転送する。

また、CPU311は、姿勢センサ351の出力信号によりロボット装置201の体幹部ユニット202の姿勢や傾きを検出するとともに、各接地確認センサ352、353の出力信号により各脚部ユニット205R/Lが遊脚又は立脚の何れの状態であるかを検出することによって、ロボット装置201の全身協調運動を適応的に制御することができる。更

に、CPU311は、ZMP位置が常にZMP安定領域の中心に向かうように、ロボット装置201の姿勢や動作を制御する。

また、運動制御モジュール300は、思考制御モジュール200において決定された意思通りの行動がどの程度発現されたか、すなわち処理の状況を、思考制御モジュール200に返すようになっている。このようにしてロボット装置201は、制御プログラムに基づいて自己及び周囲の状況を判断し、自律的に行動することができる。

(2)ロボット装置の動作制御方法

上述のロボット装置においては、頭部ユニット203にステレオビジョンシステムを搭載し、外界の3次元距離情報を取得することができる。次に、このようなロボット装置などに好適に搭載されるものであって、ロボット装置が、ステレオビジョンシステムにより周囲の環境から獲得した3次元距離データを使用して平面を検出し、この平面検出結果に基づき階段を認識し、この階段認識結果を使用して階段昇降動作を行う一連の処理について説明する。

図7は、ロボット装置がステレオデータから階段昇降動作を発現するまでの処理を実行するシステムを示す機能ブロック図である。図7に示すように、ロボット装置は、3次元の距離データを取得する距離データ計測手段としてのステレオビジョンシステム(Stereo Vision System)1と、ステレオビジョンシステム1からステレオデータD1が入力され、このステレオデータD1から環境内の平面を検出する平面検出器(Plane Segmentation/Extractor)2と、平面検出器2から出力される平面データD2から階段を認識する階段認識器(Stair Recognition)3と、階段認識部2により認識された認識結果である階段データD4を使用して階段昇降動作をするための動作制御指令D5を出力する階段昇降制御器(Stair Climber)4とを備える。

そして、ロボット装置は、まず、ステレオビジョンによって外界を観測し、両眼の視差によって算出される3次元距離情報であるステレオデータD1を画像として出力する。すなわち、人間の両眼に相当する左右2つのカメラからの画像入力を各画素近傍毎に比較し、その視差から対象までの距離を推定し、3次元距離情報を画像として出力(距離画像)する。この距離画像から平面検出器2によって平面を検出することで、環境内に存在する複数の平面を認識することができる。更に、階段認識器3によって、

これら平面からロボット装置が昇降可能な平面を抽出し、その平面から階段を認識し階段データD4を出力する。そして階段昇降制御器4が階段データD4を用いて階段の昇降動作を実現する動作制御指令D5を出力する。

図8Aは、ロボット装置201が外界を撮影している様子を示す模式図である。床面をx-y平面とし、高さ方向をz方向としたとき、図8Aに示すように、画像入力部(ステレオカメラ)を頭部ユニット203に有するロボット装置201の視野範囲は、ロボット装置201の前方の所定範囲となる。

ロボット装置201は、上述したCPU211において、画像入力装置251からのカラー画像及び視差画像と、各アクチュエータ350の全ての関節角度等のセンサデータなどが入力されて各種の処理を実行するソフトウェア構成を実現する。

本実施の形態のロボット装置201におけるソフトウェアは、オブジェクト単位で構成され、ロボット装置の位置、移動量、周囲の障害物、及び環境地図等を認識し、ロボット装置が最終的に取るべき行動についての行動列を出力する各種認識処理等を行うことができる。なお、ロボット装置の位置を示す座標として、例えば、ランドマーク等の特定の物体等に基づく所定位置を座標の原点としたワールド基準系のカメラ座標系(以下、絶対座標ともいう。)と、ロボット装置自身を中心(座標の原点)としたロボット中心座標系(以下、相対座標ともいう。)との2つの座標を使用する。

ステレオビジョンシステム1では、カラー画像及びステレオカメラによる視差画像などの画像データが撮像された時間において、センサデータから割り出した関節角を使用してロボット装置201が中心に固定されたロボット中心座標系を頭部ユニット203に設けられた画像入力装置251の座標系へ変換する。この場合、本実施の形態においては、ロボット中心座標系からカメラ座標系の同次変換行列等を導出し、この同次変換行列とこれに対応する3次元距離データからなる距離画像を出力する。

本実施の形態におけるロボット装置は、自身の視野内に含まれる階段を認識することができ、その認識結果(以下、階段データという。)を使用して階段昇降動作を可能とする。したがって、階段昇降動作のためには、ロボット装置は、階段の大きさが自身の足底の大きさより小さいか、階段の高さが登ることができる又は降りることができる高さか否かなどの、階段の大きさについて様々な判断を行う必要がある。

ここで、本実施の形態においては、ロボット装置の足底の大きさを図8Bとした場合について説明する。すなわち、図8Bに示すように、ロボット装置201の前進方向をx軸方向、床面に平行でx方向と直交する方法をy方向としたとし、ロボット装置201が直立した際の両足のy方向の幅をfeet base width足底の大きさであて足首(脚部と足底の接続部)から前側の部分を足底前幅foot_front_size、足首から後ろ側の部分を足底の足底後ろ幅foot_back_sizeとするものとする。

ロボット装置201が環境内から検出する階段としては、例えば図9、図10に示すようなものがある。図9A、図10Aは、階段を正面から見た図、図9B、図10Bは、階段を側面から見た図、図9C、図10Cは、階段を斜めから見た図である。

ここで、人間、ロボット装置などが階段を昇降するために使用する面(足又は可動脚部を載せる面)を踏面といい、一の踏面からその次の踏面までの高さ(階段1段の階段の高さ)を蹴り上げという。また、以下では、階段は、地面に近い方から登るに従って1段目、2段目とカウントすることとする。

図9に示す階段ST1は、段数が3段の階段であり、蹴り上げ4cm、1, 2段面の踏面の大きさは幅30cm、奥行き10cm、最上段である3段目の踏面のみ、幅30cm、奥行き21cmとなっている。また、図10に示す階段ST2も、段数が3段の階段であり、蹴り上げ3cm、1, 2段面の踏面の大きさは幅33cm、奥行き12cm、最上段である3段目の踏面のみ、幅33cm、奥行き32cmとなっている。ロボット装置がこれらの階段を認識した結果は後述する。

平面検出器2は、ステレオビジョン等の距離計測器から出力される距離情報(ステレオデータD1)から環境内に存在する複数の平面を検出し、平面データD2を出力する。平面の検出方法としては後述する線分拡張法その他、ハフ変換を利用した公知の平面検出技術を適用することができる。ただし、ノイズを含む距離データから階段のように複数平面を検出するには、後述する線分拡張法などによる平面検出を行うと正確に平面を検出することができる。

図11、図12は、階段を検出した結果の一例を示す図である。図11及び図12は、後述する平面検出方法により、それぞれ図9及び図10に示す階段を撮影した画像から3次元距離データを取得して平面検出した例である。すなわち、図11Aは、図9

の階段を撮影した場合の画像を示す模式図、図11B乃至図11Dは、図11Aに示す画像から取得した3次元の距離データを示す図である。また、図12Aは、図10の階段を撮影した場合の画像を示す模式図、図12B乃至図12Dは、図12Aに示す画像から取得した3次元の距離データを示す図である。図11及び図12に示すように、何れの場合も全ての踏面を平面として検出できている。図11Bは、下から1段目、2段目、3段目の踏面が平面検出されている例を示す。また、図12Bは、床面の一部も他の平面として検出成功していることを示す。

すなわち、図13Aに示すように、例えば階段ST2を撮影した距離画像から平面検出すると、図13Bに示すように、領域A〜Dがそれぞれ、床面、1段目、2段目、3段目の踏面を示す平面として検出される。各領域A〜Dに含まれる同一領域に示す点群は、それぞれ同一平面を構成すると推定された距離データ点群を示している。

階段認識器3には、こうして平面検出器2が検出した平面データD2が入力され、階段の形状、すなわち踏面の大きさ、階段の高さ(蹴り上げの大きさ)などを認識する。ここで、本実施の形態における階段認識器3は、詳細は後述するが、ロボット装置201が認識した踏面に含まれる領域(多角形)の手前側(ロボット装置に距離的に近い側)の境界(Front Edge)(以下、フロントエッジFEという。)と、踏面の奥側(ロボット装置に距離的に遠い側)の境界(Back Edge)(以下、バックエッジBEという。)とを階段データとして認識する。そして、階段昇降制御器4が階段データを利用して階段昇降動作を制御する。

次に、ロボット装置の階段昇降制御方法について具体的に説明する。なお、以下では、第1に、ロボット装置の階段認識方法、第2に認識した階段を利用して行う階段昇降動作、最後に平面検出方法の具体例として線分拡張法による平面検出方法の順にて説明する。

図14は、図7に示す階段認識器を示す機能ブロック図である。図14に示すように、階段認識器3は、平面検出器2から出力された平面データD2から階段を検出する階段検出器(Stair Extraction)5と、この階段検出器5が検出した階段データD3の時系列データ、すなわち異なる時間に検出された複数の階段データD3を統合することで更に正確に階段を認識する処理を行う階段統合器(Stair Merging)6とを有し、階段

統合器にて統合した階段データD4が階段認識器3の出力となる。

階段検出器5は、平面検出器2から入力される平面データD2から階段を検出するが、

平面検出器2から入力される平面データD2は、1つの平面につき以下に示す複数の情報を有し、ステレオビジョンシステム1によって取り込まれた画像から検出された複数の平面毎の平面データが入力される。

すなわち、平面データD2は、平面毎に

1-1: 平面を構成する点の数(number of supporting point)

1-2: 平面の中心となる点

1-3: 平面パラメータ(法線ベクトル、原点からの距離)

1-4: 平面を構成する多角形の境界

から構成される情報を有する。

この平面データに基づき、ロボット装置は、自身が接地している床面や踏面などの接地面と略水平な平面を選択し、下記の情報(以下、階段パラメータという。)を算出する。すなわち、

2-1: フロントエッジFE、バックエッジBE

2-2: 階段の高さ

である。

ロボット装置が認識するフロントエッジFE、バックエッジBEとは、上述した如く、階段の踏面の境界(線)を示すものであって、ロボット装置が対峙した場合に、多角形においてロボット装置に近い側の境界(手前側の境界)をフロントエッジFE、ロボット装置から距離が離れている側の境界(奥側の境界)をバックエッジBEとする。これは、後述するように、例えば平面を構成する点を全て含む最小の多角形を求め、その手前側又は奥側の境界とすることができる。フロントエッジFE及びバックエッジBEの情報としては、これらの端点の情報などとすることができる。また、上記多角形から階段の幅W(width)、階段の長さ(length)などの情報を得ることができる。また、階段の高さ(蹴り上げ)は、与えられた平面データD2の平面の中心点を利用して、2の平面の中心点間の高さ差としたり、上記多角形を求めた際の重心点を利用して2の重心点の

高さの差としたりすることができる。なお、蹴り上げは、前段のバックエッジBEと後段のフロントエッジFEとの高さの差としたりしてもよい。

また、本実施の形態においては、フロントエッジFE及びバックエッジBEに加えて更にフロントエッジFE及びバックエッジBEに挟まれた領域(安全領域)の左右に隣接する領域であって移動可能である確率が高いと推定される領域をマージン(領域)として認識する。これらの求め方については後述する。このマージンを求めることにより、移動可能であると推定する踏面の領域を広く認識することができる。更に、踏面を構成するデータ点群の数や、上述の重心点などを1つ規定した参照点の情報などの情報(階段パラメータ)のセットを階段データD3とすることができる。

以上の階段データから下記の条件を満たす平面(階段)を抽出する。

3-1: フロントエッジFE及びバックエッジBEの長さが所定の閾値以上

3-2: 階段の高さ(height)が所定の閾値以下

また、その他、

3-3: 階段の幅W (width) が所定の閾値以上

3-4: 階段の長さL (length) が所定の閾値以上

など同時に満たすものを抽出することが好ましい。

図15は、階段検出器5の階段検出処理の手順を示すフローチャートである。図15に示すように、先ず、入力された平面データの平面パラメータなどに基づき、入力された平面が例えば接地面と水平であるか否かなど、歩行又は移動可能である平面か否かを判断する(ステップS1)。ここで、どのような平面が水平であるか又は移動可能であるかの条件は、ロボット装置の機能に応じて設定すればよい。例えば、入力平面の平面ベクトルを $n(n_x, n_y, n_z)$ とした場合、 $|\sin^{-1} \frac{n_z}{|n|}| > \min_{th}$ であれば水平であると判断することができる。ここで、 $\min_{th}$ は、水平面を判断するための閾値であり、例えば、使用する距離データ、平面検出など精度を考慮して $\min_{th} = 80^\circ$ などとして、水平面に対して $\pm 10^\circ$ までの傾きであれば水平と判断して検出するなどとすることができる。または、例えば、 $\pm 30^\circ$ 程度の傾きがあっても歩行または移動可能であれば、それらの角度範囲の平面を抽出するようにすればよい。ステップS1にて水平でないと判断した場合(ステップS1:No)、検出に失敗したことを出力して処理を終了し、次

の平面データについての処理を実行する。

次に、平面が水平である場合(ステップS1:Yes)、平面の境界(形状)を認識するための処理を行う。ここでは、例えばSklanskyのアルゴリズム(J. Sklansky, "Measuring concavity on a rectangular mosaic", IEEE Trans Comput.21,1974,pp.1355-1364)やMelkmanのアルゴリズム(Melkman A., "On-line Construction of the Convex Hull of a Simple Polygon" Information Processing Letters 25,1987,p.11)などの凸包や、ノイズ除去による平滑化によって、入力平面を包含する多角形を求める(ステップS2)。そして、この多角形の前後の境界線をフロントエッジ及びバックエッジなどの階段パラメータを求める(ステップS3)。そして、フロントエッジ及びバックエッジの両境界線から、本実施の形態においては、階段踏面を示す平面における幅W(width)及び長さL(length)を求め、これらの値が所定の閾値より大きいかどうか判断する(ステップS4)。踏面の幅及び長さが所定の閾値以上でない場合(ステップS4:No)、ロボット装置が移動可能な平面ではないとし、次の平面データについて再びステップS1からの処理を繰り返す。

平面の幅及び長さが所定の閾値以上である場合(ステップS4:No)、移動可能な踏面であると判断し左右のマージン(Left Margin, Right Margin)を計算し(ステップS5)、これらの情報を階段データD3として出力する。

次に、ステップS2において凸包によって入力平面を包含する多角形を求める方法について説明する。図16は、凸多角形を示す模式図であって、図16Aは、入力された一の平面に属すると判断されたサポータリングポイント全て(同一平面であって連続した領域に含まれるとされる距離データ点群)が含まれる領域を示し、図16Bは、図16Aに示す図形から求めた凸多角形である。ここで示す凸多角形は、与えられた平面図形(サポータリングポイントが含まれる領域)を含む最小の凸集合を求める凸包(convex hull)を利用したものとすることができる。なお、Gで示す点は、後述するが、踏面の幅Wを求める際に使用するもので、例えばサポータリングポイントが含まれる領域の重心などの点(参照点)を示す。

このような凸包を利用して凸多角形を求めるアルゴリズムの例として、上述したように、Melkmanのアルゴリズム、Sklanskyのアルゴリズムなどがある。図17は、Melkmanの

アルゴリズムを説明するための模式図である。図17に示すように、与えられた図形に含まれる点を3点P1, P2, P3抽出し、点P1, P2を結ぶ線分を引き、点P1, P3, P2, P3を通る直線を引く。これにより、3点P1, P2, P3からなる三角形AR4を含む5つの領域AR1〜AR5に区画される。そして、次に選択した点P4がどの領域に含まれるかを判断して多角形を形成しなおすという処理を繰り返して凸多角形を更新していく。例えばP4が、領域AR1に存在する場合、P1, P2, P4, P3の順序で結んだ線分に囲まれた領域が更新された凸多角形となる。また、領域AR3, AR4に点P4が存在する場合には、それぞれP1, P2, P3, P4の順序で結んだ線分に囲まれた領域、P1, P4, P2, P3の順序で結んだ線分に囲まれた領域として凸多角形を更新する。一方、領域AR4、すなわち凸多角形内部に点Pが含まれる場合は、凸多角形は更新せず、また、領域AR2に点P4がある場合は、点P3は除きP1, P2, P3の順序で結んだ線分に囲まれた領域として凸多角形を更新する。本実施の形態においては、全てのサポーティングポイントに対して、各点に含まれる領域を考慮して凸多角形を生成することができる。

図18は、Sklanskyのアルゴリズムにより多角形を求める方法を説明するための模式図である。Sklanskyのアルゴリズムにより抽出される多角形は、Weakly Externally Visible Polygonと呼ばれるものであるが、上述のSklanskyのアルゴリズムに比べて計算量が少なく、したがって高速演算が可能である。

与えられた図形を包含する凸多角形を求める場合、図18Aに示すように、与えられた図形131の境界上の任意の点xから図形131を含む円132に対して半直線を引く。このとき、点xから円132に引いた半直線のうち、図形131を横切らない半直線を引くことができた場合、この点は凸多角形の境界を構成する点であるとする。一方、図18Bに示すように、与えられた図形133の境界上任意の他の点yから図形133を含む円134に半直線を引いた場合には、図形133を横切らない半直線を引くことができない。この場合は、この他の点yは凸多角形の境界を構成しないものとする。以上のようにして、順次各点が凸多角形の境界を構成するか否かを判断して選択された点のみからの図形を求めると、図16Aに示すような図形が得られる。

この図形を包含する凸多角形を求めることで、図16Bの凸多角形を得ることができ

る。ここで、本実施の形態においては、ステレオビジョンシステム1の精度、特性などを考慮し、図16Aから凸多角形を求める場合には、図16Bに示すように、図16Aの図形に外接する凸多角形を求めるものとして説明するが、カメラの精度、特性などを考慮し、例えば図16Aの図形に内接する凸多角形を求めるようにしてもよいことは勿論である。また、これらの方法を平面の傾き度合いや、周囲の状況に応じて使い分けられるようにしてもよい。

また、図15のステップS2において凸包によって入力平面を包含する多角形を求めた場合、非凸多角形形状の階段について問題が発生する。図19は、この問題を示す模式図であって、図19Aは、入力される平面でありstep0は非凸多角形形状の階段である。図19Bは、凸包によるstep0の多角形表現結果であり、非凸部分に関して望ましい結果と大きな乖離が生じている。このような非凸多角形を扱う方法として、ギャップ除去とラインフィットによる平滑化によって多角形を求める方法が考えられる。

ギャップ除去とラインフィットによる平滑化によって入力平面を包含する多角形を求める方法について説明する。図20は、平滑化を示す模式図であって、図20Aは、入力された一の平面に属すると判断されたサポータリングポイント全て(同一平面であって連続した領域に含まれるとされる距離データ点群)が含まれる領域である入力多角形(input polygon)を示し、図20Bは、入力平面を示す多角形から不連続なギャップを除去し(close gaps)平滑化した多角形(ギャップ除去多角形:gaps closed polygon)を示し、図20Cは、図20Bで得られた多角形に対してラインフィッティングにより(fit line segments)更に平滑化した多角形(平滑化多角形:smoothed polygon)である。ここで、図21は、ギャップ除去とラインフィットによる平滑化によって入力平面を包含する多角形を求める処理のプログラム例を示す図であり、多角形から不連続なギャップを除去するClose gaps処理、及び得られた多角形に対してラインフィッティングにより更に平滑化するFit line segments処理を示している。

まず、ギャップ除去の方法について説明する。多角形を示す頂点から連続する3つの頂点を選び、この中央の点が端点を結んだ直線から大きく離れている場合にこの中央の点を除去する。残った頂点について、この処理を除去する点が無くなるまで続ける。

次に、ラインフィットの方法について説明する。多角形を示す頂点から連続する3つの頂点を選び、最小二乗法によりこれらの3点を近似する直線とこの直線と3点の誤差を求める。求められた全ての近似直線と誤差について、誤差の小さい順に並べ、誤差がある閾値より小さい場合に、中央の点を除去し、端点の位置を近似直線によって再計算する。この処理を除去する点が無くなるまで続ける。

次に、図15のステップS3における処理について説明する。まず、得られた多角形(図16B、図20C)から上述した階段パラメータを算出する。図22は、階段パラメータの算出方法を説明するための模式図である。図22Aに示すように、得られた多角形140が、点141〜点147により囲まれる領域であるものとする。ここで、ロボット装置201からみて多角形140の前側の境界を構成する線分がフロントエッジFE、奥側の境界を構成する線分がバックエッジBEである。

階段踏面の幅Wは、フロントエッジFEの中心点 C_{FE} と参照点Gとを結ぶ線分の長さを d_1 、バックエッジBEの中心点 C_{BE} と参照点Gとを結ぶ線分の長さを d_2 としたとき、幅 $W = d_1 + d_2$ とすることができる。

ここで、参照点Gは、踏面となる平面の略中心であればよく、例えば、全サポーティングポイントの中心点としたり、多角形140の重心としたり、フロントエッジFE及びバックエッジBEの端点を結んだ図22Bに示す安全領域152の重心としたりすることができる。

こうして得られたフロントエッジFE、バックエッジBE、参照点Gに基づき、階段の長さL及び幅Wを求めステップS4の処理を実行する。階段の長さLは、フロントエッジFE及びバックエッジBEの長さのうち例えば短い方としたり、以下に示す左右のマーヅィンを含めたフロントエッジFE及び左右のマーヅィンを含めたバックエッジBEのうち長い方としたりすることができる。

次に、ステップS5におけるマーヅィン算出方法について説明する。図23は、最終的に認識される踏面及び階段パラメータを説明するための模式図である。本実施の形態においては、図22B、図23に示すように、安全領域152の左右端部にマーヅィン M_1 、 M_2 を設け、この左右のマーヅィン M_1 、 M_2 を含んだ領域151を最終的に踏面として認識するものとする。左右のマーヅィン M_1 、 M_2 は、フロントエッジFE及びバックエッジ

BEにて規定される安全領域152の外側に多角形がはみ出している場合、先ず、それらの点を選択する。図22Aにおいて、例えば右マージン M_2 を求める場合であれば、点142, 143, 146である。安全領域152の右側に隣接した右マージン M_2 を求める際は、安全領域152から最も離れた点である点142を選択し、この点142からフロントエッジFE、バックエッジBEに対して垂線を下ろす。そして、これら垂線及びフロントエッジFE、バックエッジBEに囲まれる領域151を踏面として認識するものとする。なお、このマージンの求め方としては、点142を通り、フロントエッジFE又はバックエッジBEと交差する線分を引くのみでもよい。

ここで、図22Bに示すように、フロントエッジFEと同一直線における左マージン M_1 の長さを l_{mf} とし、バックエッジBEと同一直線における左マージン M_1 の長さを l_{bm} とする。同様に、フロントエッジFE、バックエッジBEとそれぞれ同一直線における右マージン M_2 の長さを r_{fm} , r_{bm} とする。

このようにして多角形からフロントエッジFE、バックエッジBEを求めて階段を認識することの効果について説明しておく。図24A及び図24Bは、2種類の階段を示す模式図である。図24Aに示すのは、図9、図10に示したような踏面が長方形の形状の階段であるが、図24Bは、螺旋(spiral)状の階段である。この図24Bのような螺旋階段の場合、フロントエッジFEに対してバックエッジBEは平行でない。したがって、例えば単に検出した平面から長方形領域を抽出してしまうようなアルゴリズムを適用できない場合がある。したがって、本実施の形態のように、検出した平面から多角形を求め、フロントエッジFE及びバックエッジBEを求めることにより、このような螺旋階段であってもロボット装置が昇降動作することが可能となる。

次に、図14に示す階段統合器6について説明する。階段統合器6は、階段検出器5によって検出された階段データ(階段パラメータ)D3を入力とし、それらの階段データD3を時間的に統合することで、より正確かつ高域な階段の情報を推定するものである。例えば、ロボット装置の視野が狭い場合などにおいて、一度に階段全体を認識できない場合がある。そのような場合、例えば、例えば前フレームなどの古い階段データと例えば現フレームなどの新しい階段データとの中で、空間的にオーバーラップしている階段の組を探し、オーバーラップしている階段を統合することにより、新しい

仮想的な階段を定義する。この作業をオーバーラップする階段がなくなるまで続けることによって正確な階段を認識することが可能となる。

図25は、階段統合器6における階段統合処理の方法を示すフローチャートである。まず、現在の階段データ(New Stairs)と古い階段データ(Old Stairs)とを入力とし(ステップS11)、新しい階段データ及び古い階段データの全てを1つの集合(union)とする(ステップS12)。これら合わされた階段データ集合において、空間的にオーバーラップしている階段データを検索し(ステップS13)、オーバーラップしている組がある場合には(ステップS14:Yes)、それらの階段を統合して当該階段データ集合に登録する(ステップS15)。そして、ステップS13、14の処理を空間的にオーバーラップする階段の組がなくなるまで続け(ステップS14:No)、最終的に更新された階段データ集合を階段データD4として出力する。

図26は、オーバーラップしている階段データを統合するステップS13における処理を説明するための模式図である。図26は、空間的にオーバーラップしている階段データST11, ST12を示す。空間的にオーバーラップしているかどうかの判断には、例えば2つの階段データST11, ST12の参照点Gにおける高さの差(距離)と、左右のマーヅンを含めた踏面の領域が重なっている面積の大きさを利用することができる。すなわち、2つの階段の重心 G_{11} , G_{12} の高さの差が閾値(maxdz)以下であり、かつ、オーバーラップしている面積の大きさが閾値(minarea)以上ある場合には、これらの2つの階段データが示す踏面はオーバーラップしていると判断することができる。その場合には図26の下図に示すように、階段データST11, ST12を統合して重心 G_{13} の階段データST13とする。

ここで、統合に際しては、階段データST11, ST12を含む外枠の領域をステップS13とし、統合前の階段データST11及び階段データST12の左右のマーヅンを除いた安全領域を含む領域を統合後の新たな安全領域165とし、階段データST13からこの安全領域165を除いた領域をマーヅン M_1 , M_2 とする。統合後の安全領域165により統合されたフロントエヅジFE及びバックエヅジBEを求めることができる。

すなわち、統合後の階段データST13におけるフロントエヅジFEの両端点は、階段データST11のフロントエヅジFEと階段データST12のフロントエヅジFEの左右の端

点を比較し、右端点163はより右側にある方、左端点はより左側にある方とされる。また、フロントエッジFEのラインの位置は、階段データST11のフロントエッジFEと階段データST12のフロントエッジFEを比較してよりロボット装置に近い方(手前側)のライン位置とされる。同様にバックエッジBEもより奥側にある方の位置が選択され、より左右に広がるようにその左右の端点161, 162が選択される。

なお、統合方法はこれに限るものではない。本実施の形態においては、ロボット装置の視界などを考慮してフロントエッジFE及びバックエッジBEにより決まる四角形の領域、統合データST13が共に最も大きくなるように統合するものとしているが、例えば、十分視界が広い場合や、距離データの精度が十分高い場合においては、2つの階段データを単に合わせた領域を統合後の階段データとするなどしてもよい。また、統合後の参照点Gは、階段データST11と階段データST12に含まれるサポーティングポイントの数の比に応じて重み付き平均をとって求めることができる。

次に、図7に示す階段昇降制御器4について説明する。階段昇降制御器4は、階段検出器及び階段統合器6によって統合されて検出された階段データD4を用いて実際にロボット装置が階段の昇降動作を行うための制御を行う。この昇降制御には、階段を探す動作も含むものとする。

本実施の形態において実現した階段昇降動作は、次の5つのステートマシンとして構築することができる。

4-1:サーチ(Search)動作

ロボット装置の頭部に搭載されたステレオビジョンシステムにて距離画像を取得するため、首を振って周囲を見廻し、環境の情報を集める。

4-2:アライン(Align)動作

階段に対して正対し、かつ、決められた一定量の距離位置に移動する。図27は、アライン動作を説明するための図である。図27において、領域170がロボット装置の認識した階段踏面の1段目であるとする。アライン動作においては、踏面170のフロントエッジFEの中心点から直交する方向に所定距離ad (align_distance)離れた目的とする位置(以下、アライン位置という。)に移動する。この場合、現在のロボット装置の位置が点171であって、目的とするアライン位置が点172である場合、両者の距離

が所定の閾値 max_d 以上離れている場合及びフロントエッジFEに直行する方向とロボット装置の向く方向との角度差が所定の閾値 max_a 以上ある場合、ロボット装置は目的とするアライン位置172に移動を開始し、これらの条件を満たしたとき、アライン動作が完了したと判断するものとする。これらの閾値としては、例えば $\text{max_d}=3(\text{cm})$ 、 $\text{max_a}=2^\circ$ とすることができる。

4-3: アプローチ (Approach) 動作

アプローチ動作では、ロボット装置が階段の直前まで移動する。図28は、アプローチ動作を説明するための模式図である。図28に示すように、階段と正対しAlign_distanceだけ離れた目的位置であるアライン位置172に移動してアライン動作を終了したロボット装置201は、階段の昇降を行うために、踏面170のフロントエッジFEの中心点 C_{FE} とロボット装置201とが正対し、かつその距離が所定の値 $\text{ax}(\text{approach_x})$ となる目的の位置(以下、アプローチ位置という。)に移動する。

4-4: クライム (Climb) 動作

階段認識によって得られた階段データを元に階段昇降動作をする。次の踏面(段)に移動した場合であって、次の段の踏面が観測されている場合には、更に昇る又は下る動作を続ける。この動作を次の段がなくなるまで続けて行うことにより階段昇降動作が実現される。

4-5: フィニッシュ (Finish) 動作

クライム動作にて階段を登る動作をした場合は、最上段にいることを確認し、最上段の中央へ移動する。クライム動作にて降りる動作をした場合、又は図9に示すような階段ST1である場合には例えば向きを変えるなどして降りる動作をした場合には、床面に降りたことを認識することで階段昇降動作は終了される。

以上の階段昇降処理の動作方法について更に詳細に説明する。図29は、階段昇降動作の手順を示すフローチャートである。図29に示すように、階段昇降制御の処理が開始されると、サーチ (Search)・アライン (Align)・アプローチ (Approach) 動作によって、階段を検索し、検索した階段に対して対峙した所定位置に移動(アライン)し、階段の1段目に近づくアプローチ動作を実行する(ステップS21)。このサーチ・アライン・アプローチ動作が成功した場合(ステップS22: Yes)、後述する方法にて階段

昇降を行い(ステップS23)、成功したことを出力する。近づくことに失敗した場合(ステップS22:No)、失敗したことを出力して処理を終了する。この場合は、もう一度、ステップS21の処理から繰り返すなどする。

ここで、ステップS21におけるサーチ・アライン・アプローチのシーケンスは通常ロボット装置が物体や目的地に到達するために用いる制御と同じである。具体的には例えば以下に示すような方法がある。図30は、サーチ・アライン・アプローチ処理方法を示すフローチャートである。

図30に示すように、サーチ・アライン・アプローチが開始されると、サーチ動作(1)を実行する(ステップS32)。サーチ動作(1)では、首を振りできるだけ広い範囲の情報を集める動作とする。その結果、周囲に昇降可能な階段があるかどうかを判断する(ステップS32)。ここで、検出された階段の中で1段目の踏面を構成する平面の高さ n_z を利用し、高さが $\text{step_min_z} < n_z < \text{step_max_z}$ を満たす場合に、昇降可能であると判断する。昇降可能な階段があった場合(ステップS32:Yes)、階段を近くで認識しなおすために、階段に対して決められた距離(align_distance)まで移動するアライン動作を実行し(ステップS33)、昇降しようとしている階段を再度認識する(ステップS34)。このステップS34の動作がサーチ動作(2)である。

そして、昇降可能な階段であるか否かを再度確認し(ステップS35)、サーチ動作(2)が成功している場合には、再認識した階段に対して対峙しかつ所定の距離のアライン位置に移動完了できているか否か、すなわちステップS33のアライン動作が成功しているか否かを確認し(ステップS36)、昇降可能な階段があり、かつアラインされている場合には(ステップS35、S36:Yes)、初段の階段のフロントエッジまで進むアプローチ動作を実行する(ステップS37)。一方、ステップS35にて昇降可能な階段がない場合にはステップS31の処理に戻り、ステップS36にてアライン動作が成功していない場合にはステップS33の処理から繰り返す。

次に、ステップS22における階段昇降動作について説明する。階段昇降動作は、一段上又は下の段(以下、次段という。)を現在の踏面からロボット装置自身が認識できる場合の昇降動作処理1と、現在の移動面から次段は認識できない場合等であって、2段以上、上の段又は下の段(以下、2段以上先の段という。)を認識できる場合

の昇降動作処理2と、複数段の踏面が認識可能な場合の昇降動作処理3とがある。図31、図33、図34は、それぞれ昇降動作処理1〜3の処理方法を示すフローチャートである。ここで、以下の説明においては、現在移動中の段(床面を含む)をstep-0、次段の段をstep-1、その更に次の段をstep-2、m段先の段をstep-mとする。

まず、次段(step-1)の踏面を観測・認識できる場合の昇降動作処理について説明する。図31に示すように、昇降動作処理1が開始されると、階段を登る／降りる動作(クライム動作(1))を実行する(ステップS41)。このクライム動作(1)では、上述のステップS32において、階段の高さ n_z を認識できているため、この高さ n_z の正負の判断をすることにより、昇降モードを切り替える。すなわち、 $n_z < 0$ であれば降りる動作であり、 $n_z > 0$ であれば登る動作となる。登る場合と降りる場合では、後述するように、クライム動作に使用する制御パラメータの値が異なる。すなわち、制御パラメータを切り替えるのみで、登る動作と降りる動作を切り替え実行することができる。

そして、クライム動作(1)が成功したか否かを判断し(ステップS42)、成功した場合(ステップS42:Yes)は、サーチ動作(3)を実行する(ステップS43)。このサーチ動作(3)は、ステレオビジョンシステムが搭載された頭部ユニットを動かし、周囲の距離データを取得して次の階段を検出する処理であり、サーチ動作(2)などとはほぼ同様の動作処理である。

そして、ステップS35と同様、サーチ動作(3)が成功したら(ステップS44:Yes)、再びステップS41の処理を繰り返す。階段が検索されなかった場合(ステップS44:No)、階段を登り切ったか、又は下りきったものと判断し、階段面の中央など所定の位置まで移動するなどのフィニッシュ動作を実行し(ステップS45)、処理を終了する。

次に、ステップS41のクライム動作(1)について詳細に説明する。図32は、ロボット装置が認識しているか又は認識する予定の階段面を示す模式図である。図32に示すように、例えば、現在移動中のロボット装置の足底121L/Rが踏面181にあるとする。また、この図32においては、フロントエッジFE及びバックエッジBEに挟まれる安全領域及びこれに隣接する左右のマージン M_1 、 M_2 を踏面と認識することとする。このとき、ロボット装置は、次の次段(step-2)の踏面182を認識できている。ここで、踏面181と踏面182との間にはその蹴り上げなどによりギャップ184が存在するものと

する。昇降動作(1)では、現在の段(step-0)の踏面181から次の段(step-1)の踏面182に移動(クライム動作)可能であるかを判断するが、例えば、以下の基準を満たすものを移動可能と判断するものとする。

5-1: 次段(step-1)の踏面182のフロントエッジFE₁に十分近く、角度のずれが所定の閾値以下

5-2: 次段(step-1)の踏面182の大きさが十分に大きい

5-3: フロントエッジFE₁から足底121L/Rの後端までの距離front_xが指定された昇降モードにおける制御パラメータfront_x_limitより大きい

5-4: バックエッジBE₁から足底121L/Rの前端までの距離back_xが昇降モードにおける制御パラメータback_x_limitより大きい

ここで、更に次段(step-2)の踏面183を観測できた場合、次段(step-1)の踏面182の参照点における高さz1と更に次の段(step-2)の踏面183の参照点における高さz2の差(z2-z1)から、次段(step-1)の踏面182からその次の段(step-2)の踏面183へのクライム動作が登りであるか下りであるかを判断することができる。なお、2段先の段(step-2)の踏面183が認識できない場合は、現在の昇降状態が維持されるものとする。

上記の5-1〜5-4からクライム動作可能と判断した場合は、階段を登る又は降りるクライム動作を実行する。ロボット装置は、現在の段(step-0)の踏面181において、踏面181のバックエッジBE₀にアラインしている。ここで、踏面181と踏面182との間のギャップ184が大きい場合には、次段(step-1)の踏面182のフロントエッジFE₁にアラインする動作をした後に、次段に移動し、そのバックエッジBE₁にアラインする。そして、次のクライム動作にて、次の段(step-2)の踏面183のフロントエッジFE₂にアラインし、踏面183に移動し、そのバックエッジBE₂にアラインする。すなわち、例えば、次の段の踏面のフロントエッジFEにアラインして、昇降動作し、その移動した踏面のバックエッジBEにアラインするまでをクライム動作とする。

また、ギャップ184が小さい場合には、現在の段(step-0)の踏面181のバックエッジBE₀と次段の踏面182におけるフロントエッジFE₁がほぼ一致するものとして、何れか一方のエッジにのみアライン動作するようにしてもよい。すなわち、現在の踏面181

のバックエッジ BE_0 にアラインして次段の踏面182に移動し、そのバックエッジ BE_1 にアラインするか、次段の踏面182のフロントエッジ FE_1 にアラインして次段の踏面に移動し、更に次の段の踏面183のフロントエッジ FE_1 にアラインすればよい。この場合、クライム動作とは、次段のフロントエッジ FE にアランする処理を省略し、次段に移動して移動した踏面のバックエッジ BE にアライン動作を実行する処理となる。

以上の昇降動作処理1は、ロボット装置が階段を昇降動作中に次に移動可能な段(step-1)の踏面を観測できる場合に適用することができる。例えば2足歩行ロボット装置であれば、自身の足元を見下ろすことが可能なステレオビジョンシステム1を搭載しておく必要がある。

次に、次段の踏面を観測・認識可能な場合の昇降動作処理1とは異なり、ロボット装置の頭部ユニットと体幹部ユニットとの接続部の可動角の制約などにより、現在の段(step-0)における踏面から次段(step-1)の踏面を観測・認識することができず、2段先(step-2)又はそれ以上先の段(step-m)の段の踏面を認識可能な場合の昇降動作処理2、3について説明する。まず、2段先の段(step-2)の踏面を認識可能な場合について説明すると、図32に示すように、昇降動作処理2が開始されると、上述と同様に、サーチ動作(4)を実行する(ステップS51)。このサーチ動作(4)では、ステレオビジョンシステム1により2段先の段(step-2)の踏面を観察・認識する。このサーチ動作(4)は、2段先の段(step-2)の踏面を認識する以外、上述のステップS43などと同様の処理である。

そして、クライム動作(2)を実行する(ステップS52)。クライム動作(2)は、クライム動作(1)と同様の動作である。この場合においても、クライム動作における階段を登る動作と降りる動作の切り替えは、同じく次段(step-1)の踏面の高さ n_z により判断することができる。次段(step-1)の踏面とは、現在の段(step-0)の踏面より時間的に前に移動していた段の踏面において観測された踏面である。

そして、クライム動作が成功した場合(ステップS53:Yes)、ステップS51にてサーチした結果、2段先の階段(step-2)が検出されていれば(ステップS54:Yes)、それを次の階段として更新し(step-1=step-2)(ステップS56)、ステップS51からの処理を繰り返す。ステップS51のサーチ動作(4)にて2段先の段(step-2)の踏面が検出さ

れていない場合(ステップS55:No)、フィニッシュ動作を実行し(ステップS55)、処理を終了する。

なお、ここでは2段先の段の踏面を観測できることとして説明したが、3段以降先の踏面を観測できる場合であっても同様に処理すればよい。

次に、複数段(以下、 m 段)先までの踏面が観測・認識可能な場合についての昇降動作を昇降動作3として説明する。

図34に示すように、 $1 \sim n$ 段が観測・認識できている場合に昇降動作を行う場合、まず、サーチ動作(5)を実行する。サーチ動作(5)は、基本的には、ステップS51と同様であるが、認識可能な m 段先の段(step- m)までの踏面を観測対象としている点異なる。

そして、クライム動作(3)を k 段分実行する(ステップS62)。クライム動作(3)においても、現在までに観測されている複数の踏面の高さの差から昇降モードを決定することができる。すなわち、 i 番目、 $i-1$ 番目の踏面の高さ $z_i - z_{i-1}$ が負であれば階段を下りる動作となり、0又は正であれば階段を登る動作モードとすればよい。このクライム動作(3)において移動する踏面の情報は、現在の段より時間的に m 段前にて観測済みのデータである。

そして、昇降動作が成功した場合(ステップS63:Yes)、現在の踏面より先の階段データが観測できたか否か、すなわち $m+n > k$ であるか否かが判定され(ステップS64)、 $m+n > k$ であれば(ステップS64:Yes)、step- $(k+1) \sim$ step- $(n+m)$ を、step-1 $\sim$ step- $(m+n-k)$ ($n=m$)として、更新し(ステップS66)、ステップS61からの処理を繰り返す。一方、 $m < 0$ であれば(ステップS64:No)、次に移動する対象となる踏面が存在しないため、フィニッシュ動作を実行し(ステップS65)、処理を終了する。

以上説明したように、階段昇降動作処理1 $\sim$ 3においては、クライム動作における登る動作と降りる動作とで使用する制御パラメータを変更するのみで、階段を登ることも、降りることも同様の手順にて実行することができる。階段昇降動作に使用する制御パラメータは、ロボット装置の足底の現在の踏面に対する位置を規制するためのものである。

図35Aは、ロボット装置により認識されている踏面と足底の関係を説明するための

図、図35Bは、クライム動作に使用する制御パラメータの一例を示す図である。

図35Aに示す各制御パラメータは、以下を示す。

step_min_z: 現在の段と次段の段との高さの差(蹴り上げ)の昇降可能最小値

step_max_z: 現在の段と次段の段との高さの差(蹴り上げ)の昇降可能最大値

ad (align_distance): アライン位置におけるフロントエッジFEとロボット装置との間の距離

ax (approach_x): アプローチ位置におけるフロントエッジFEとロボット装置との間の距離

front_x_limit: 踏面におけるフロントエッジFEと足底121の後端部との距離の限界値 (minimal x-value)

back_x_limit: 踏面におけるバックエッジBEと足底121の前端部との距離の限界値 (maximal x-value)

back_x_desired: バックエッジBEと足底121の前端部との距離の欲求値(desired value)

階段を登る場合には、蹴り上げがstep_min_zより小さい場合は段差がある(階段)と見なさないものとし、蹴り上げがstep_max_zより大きい場合には、階段昇降不可と判断する。同様に、階段を降りる場合には、蹴り上げがstep_max_zより大きい場合には、階段と見なさないものとし、step_min_zより大きい場合には、階段昇降不可と判断するものとする。

align_distanceは、アライン動作をする場合にのみ使用するパラメータであり、階段を登る／降りる動作を実行する階段昇降動作処理を開始する場合、すなわち、初段の段に対する昇降動作を行う場合に使用される。同様に、approach_xも、アプローチ動作をする場合にのみ使用するパラメータであり、階段を登る／降りる動作を実行する階段昇降動作処理を開始する場合に使用される。

front_x_limit及びback_x_limitは、ロボット装置が認識している踏面と、ロボット装置の足底の関係を規定するもので、踏面のバックエッジBE及びフロントエッジFEと、足底の端部との距離、すなわちその踏面に移動した場合における踏面のあまり部分がこれらの値より小さい場合には、その踏面に移動できない、又は移動できたとしても次

の昇降動作が不能であると判断される。ここで、登り動作において、front_x_limit及びback_x_limitの値が何れも負であることは、踏面が足底より小さいことを許すことを示す。すなわち、登り動作においては、踏面は足底より小さい場合であっても移動可能と判断することができる。

back_x_desiredは、現在の踏面におけるバックエッジBEに対してロボット装置がアラインしたい位置におけるバックエッジBEと足底前端部との距離を示すものであり、図35Bに示すように、通常、登る場合であれば、back_x_desiredは、バックエッジBEよりも手前、本実施の形態においては、バックエッジBEから15mm手前の位置となっており、一方、降りる動作であれば、バックエッジBEより足底がはみ出した位置、本実施の形態においては、5mmはみ出した位置となる。これは、登る動作であれば、次段に足上げして移動するまでにある程度の距離が必要であるのに対し、降りる動作においては、そのような距離が不要であると共に、踏面からはみ出すような位置の方が次段又はそれ以降の踏面の観測・認識が容易となるためである。

図36及び図37は、図35に示す制御パラメータを使用して実際にロボット装置が昇降動作を行った様子を撮影したものをトレースした図である。図36は、ロボット装置が階段を登る動作を示している。最上段から番号順に、ステップS31のサーチ動作(1)実行の様子(No. 1)、ステップS33のアライン動作実行の様子(No. 2)、ステップS32のサーチ動作(2)実行の様子(No. 3)、ステップS37のアプローチ動作実行の様子(No. 4)、ステップS51のサーチ動作(4)実行の様子(No. 5)、ステップS52のクライム動作(2)実行の様子(No. 6)、ステップS52のクライム動作(2)の続きであって現在の踏面のバックエッジBEにアラインしている様子(No. 7)、ステップS51のサーチ動作(4)実行の様子(No. 8)、・・・を示し、サーチ動作(4)をして次段の踏面が観測されなかった場合(No. 17)に、階段昇降動作終了(フィニッシュ)動作を行っている様子(No. 18)を示す。

図37は、降りる動作を示すものであり、図36の登る動作と同様に、サーチ動作(No. 1、No. 4、No. 7、No. 10、No. 13、No. 16)、クライム動作(アライン動作含む)を繰り返し(No. 5、No. 6、No. 8、No. 9、No. 11、No. 12、No. 14、No. 15)、次段の踏面が観測されなくなった時点でフィニッシュ動作を行って(No. 18)、昇降

動作を終了する。

次に、本実施の形態における変形例について説明する。以上の説明においては、階段を登る動作、降りる動作について説明したが、本実施の形態の制御方法に係るアルゴリズムを適用すれば、複数の段からなる階段のみならず、1段のみからなる段部や、1段の凹部が存在しても、その段差が所定の値以下であれば、移動を可能にするものである。

まず、単一の段部に登る動作について説明する。図38は、単一の段部とロボット装置の足底の関係を示す図である。図38において、191で示すのは、床面($z=0$)からの高さが $z_1=30$ である段部を示し、ここでは、ロボット装置が次段(step1)となる段部191の紙面下側から上側に移動する場合について説明する。段部191に対し紙面下側における移動面の高さが $z_0=0$ 、紙面上側における移動面の高さが $z_2=0$ である場合であって、現在、高さ z_0 の移動面を移動しているとした場合、 $z_1-z_0=30>0$ であるので次段の段部191への移動は登り動作と判断し、 $z_2-z_1=-30<0$ であるので段部191から次の領域への移動は降りる動作であると判断することができる。この判断に応じてクライム動作において、上述した制御パラメータの値を変更すればよい。ここで、図35Bに示す制御パラメータを使用した場合、登る動作の場合の $ront_x_limit$ 及び $back_x_limit$ は、何れも負の値であって、ロボット装置の足底121が図38に示すように、段部191からはみ出した状態であっても移動可能と判断することを示す。

次に、単一の窪み(凹部)に降りる動作について説明する。図39は、単一の凹部とロボット装置の足底の関係を示す図である。図39において、192で示すのは、床面($z=0$)からの高さが $z_1=-30$ である凹部を示し、ここでは、ロボット装置が次段(step1)となる凹部192の紙面下側から上側に移動する場合について説明する。凹部192に対し紙面下側における移動面の高さが $z_0=0$ 、紙面上側における移動面の高さが $z_2=0$ である場合であって、現在、高さ z_0 の移動面を移動しているとした場合、 $z_1-z_0=-30<0$ であるので次段の凹部192への移動は降りる動作と判断し、 $z_2-z_1=30>0$ であるので凹部192から次の領域への移動は登る動作であると判断することができる。したがって、段部191と同様、この判断に応じてクライム動作において、上述

した制御パラメータの値を変更すればよい。ここで、図35Bに示す制御パラメータを使用した場合、降りる動作の場合のront_x_limit及びback_x_limitは、何れも正の値であって、ロボット装置の足底121が図39に示すように、凹部191より小さい場合にのみ移動可能と判断する。

本実施の形態においては、検出された平面から例えば水平など、移動可能と判断できる平面を抽出し、その領域を含む多角形から階段の踏面を認識する。そして、多角形のフロントエッジFE及びバックエッジBEなどの踏面に関する情報と、その床面からの高さを含む階段情報を利用して階段昇降動作を行う。昇降動作においては、移動する踏面をサーチ動作を実行し、サーチできた踏面のフロントエッジFE又は現在の移動面におけるバックエッジBEに対してアライン動作を実行し、次段の移動面と現在の移動面との高さの違いから登る動作か降りる動作かを判断して制御パラメータを切り替えることにより、通常の矩形の踏面からなる階段のみならず、スパイラル形状の階段などの昇降動作が可能となると共に、登る動作も降りる動作も制御パラメータを変更するのみで同一の手順にて実行することができる。したがって、階段のみならず、単一の段部や、単一の凹部などへの移動も同一の制御方法にて移動可能となる。

また、階段認識においては、認識した階段を時間的に統合していくため、例えばロボット装置の大きさに対して階段が大きい、ロボット装置に搭載されるステレオビジョンシステムの位置の制約などの理由で視野が限られたロボット装置などにおいても、高域に亘って階段情報を認識することができる。また、この階段情報を利用して昇降動作する際、同じくステレオビジョンシステムの位置の制約などの理由で次段の踏面が観測・認識できない場合であっても、過去に観測・認識してある階段情報を利用して昇降動作を同様に行うことができる。

本変形例における平面検出装置は、線分拡張法により、視野内において支配的な平面だけでなく、例えば階段など複数の平面が存在する場合であっても確実に複数平面を検出することができ、平面を検出する際に抽出する線分抽出において、距離データの点の分布に応じて適応的に線分をフィッティングさせることにより計測ノイズに対してロバストな平面検出結果を得ることができるものである。

図40は、本変形例における平面検出装置を示す機能ブロック図である。図40に示すように、平面検出装置100は、3次元の距離データを取得する距離データ計測手段としてのステレオビジョンシステム(Stereo Vision System) 1と、3次元の距離データからなる距離画像に存在する平面を線分拡張法により検出する平面検出部2とを有する。平面検出部2は、画像を構成する距離データ点から同一平面にあると推定される距離データ点群を選択し、この距離データ点群毎に線分を抽出する線分抽出部2aと、画像内に含まれる、線分抽出部2aによって抽出された全線分からなる線分群から、該画像内に存在する1又は複数の平面領域を検出する領域拡張部2bとを有する。領域拡張部2bは、線分群から同一平面上に存在すると推定される任意の3本の線分を選択し、これらから基準平面を求める。そして、選択した3本の線分に隣接する線分がこの基準平面と同一平面に属するか否かを判定し、同一平面に属すると判定した場合にはその領域拡張用線分としての線分により基準平面を更新すると共に基準平面の領域を拡張する。

線分抽出部2aは、その距離画像における列または行毎の各データ列において、3次元空間内で同一平面上にあると推定される距離データ点群を抽出し、この距離データ点群から距離データ点群の分布に応じて1以上の線分を生成する。すなわち、分布に偏りがあると判断された場合には、データ点群は同一平面上にないと判断し、データ点群を分割し、分割したデータ点群それぞれについて再度分布に偏りがあるかを判断する処理を繰り返し、分布に偏りがない場合にはそのデータ点群から線分を生成する。全てのデータ列について以上の処理を行い、生成した線分群D11を領域拡張部2bに出力する。

領域拡張部2bは、この線分群D11において、同一の平面に属すると推定される線分を3本選択し、これらから基準平面としての種となる平面を求める。この種となる平面の領域(領域種:seed region)に対して、該領域種と同一平面に属する線分を順次統合していくことで拡張していく領域拡張によって距離画像を複数の平面に分割し、平面群D2を出力する。

ロボット装置201は、障害物回避や階段昇降など平面の情報が必要なとき、または定期的にこれらの処理を行うことによって、階段や床面、壁といった歩行に重要な平

面の情報を取得する。

ここで、このような3次元距離データをステレオビジョンシステム1によって取得するためには、階段ST2の表面に模様(テクスチャ)が必要となる。すなわち、2台のカメラによる視差により得ることができるため、模様がないものは視差が算出できず、正確に距離を計測することができない。すなわち、ステレオビジョンシステムにおける距離データの計測精度は、計測対象のテクスチャに依存することになる。なお、視差とは、空間中のある点が左目及び右目に写像される点の違いを示し、そのカメラからの距離に応じて変化するものである。

そこで、図41に示すように、ロボット装置の頭部ユニットに、ステレオビジョンシステムを構成するステレオカメラ11R/Lを備えると共に、例えば同じく頭部ユニットなどに投射手段としての例えば赤外光などを出力する光源12を設ける。この光源12は、模様がない階段ST3、その他テクスチャがないか少ない物体、壁などの対象物に対してこれを投射(照射)し、ランダムなパターンPTを付与する模様付与手段として作用する。なお、ランダムパターンPTを形成して距離画像を取得できるものであれば、ランダムパターンPTを付与する手段は赤外光を投射する光源などには限らず、例えばロボット装置自ら対象物に模様を書いたりしてもよいが、赤外光であれば、人間の目にはみえないものの、ロボット装置に搭載されるCCDカメラなどにおいては観測可能なパターンを付与することができる。

次に、平面検出装置100の平面検出部2について説明する。この平面検出部2は、線分拡張法を使用して平面を検出するものであり、図42は、線分拡張法による平面検出方法を説明する図である。線分拡張法による平面検出では、図42に示すように、まず、焦点Fから撮影された画像11において、行方向または列方向のデータ列における処理をする。画像内の例えば行方向の画素列(image row:イメージロウ)において、距離データ点が同一の平面に属するならば直線となることを利用し、同一平面に属すると推定される距離データ点からなる線分を生成する。そして、得られた複数の線分からなる線分群において、同一平面を構成するとされる線分群に基づき平面を推定、検出する方法である。

図43は、線分拡張法による平面検出処理を示すフローチャートである。図43に示

すように、まず、距離画像を入力し(ステップS71)、距離画像の行方向(又は列方向)の各画素列において同一平面に属すると推定されるデータ点から線分を求める(ステップS72)。そして、これらの線分群の中から同一平面に属すると推定される線分を抽出し、これらの線分からなる平面を求める(ステップS73)。このステップS73では、まず、平面の種となる領域(以下、領域種(seed region)という。)を選び、該当する領域種を選択する。この選択においては、上下隣接する行方向(又は左右隣接する列方向)の1ラインを含む3本の線分が同一平面にあることを条件とする。ここで、選択した3本の線分からなる領域種が属する平面を基準平面とし、3本の線分から平均して求まる平面を求めておく。また、3本の線分からなる領域を基準平面領域とする。

そして、選択した領域種に隣接する行方向(又は列方向)の画素列からなる直線と上記基準平面とが同じ平面であるかどうかを空間的な距離を比較することで判断し、同じ平面である場合には、その隣接する線分を基準平面領域に追加し(領域拡張処理)、追加した線分を含めたものとして上記基準平面を更新し(平面更新処理)、これを平面領域に隣接するデータ列に同一平面の線分が存在しなくなるまで繰り返し行う。そして、以上領域種を検索して平面更新及び領域拡張処理を、種となる領域(3本の線分)が存在しなくなるまで繰り返し実行する。最後に、得られた複数の領域群の中から同一平面を構成するものを連結する。そして、本変形例においては、得られた平面に属する線分群のうち、平面から所定の閾値以上外れる線分を除いて再度平面を求める平面再算出処理をステップS74として更に設け、最終的な平面とするが、詳細は後述する。

ここで、3次元距離データから線分を検出し、これを同一平面毎にまとめた領域を1つの平面とする処理は従来の線分拡張法による平面検出処理であるが、本変形例においては、ステップS72における線分抽出方法が従来とは異なる。すなわち、上述したように、距離データ点から線分を求めて距離データ点にできるだけフィットするように線分を生成しようとしても、距離データの精度に応じて閾値を変更しなければ、over-segmentation又はunder-segmentationなどものの問題が生じてしまう。そこで、本変形例においては、この線分抽出において、距離データの分布を解析することで、距離データの精度、ノイズに応じて適応的に閾値を変更する手法を導入するものと

する。

以下、図43に示す線分拡張法による平面検出方法について更に詳細に説明する。線分抽出部(Line Extraction)2aは、上述したように、ステレオビジョンシステム1からの3次元距離画像を入力とし、距離画像の各列または各行毎に3次元空間内で同一平面上にあると推定される線分を検出する。この線分抽出において、計測ノイズなどによる、上述したover-segmentationやunder-segmentationの問題、すなわち、本来は複数の平面であるのに1つの平面として認識してしまったり、本来は1つの平面であるのに、複数の平面として認識してしまったりする問題を回避するため、データ点の分布に応じて適応的に線分フィッティングさせるアルゴリズム(Adaptive Line Fitting)を導入する。Adaptive Line Fittingは、線分抽出部2aにおいて、先ず比較的大きい閾値を使用して大まかに第1の線分としての線分を抽出し、次に抽出された第1の線分に属するデータ点群から後述する最小二乗法によって得られる第2の線分としての線分に対する該データ点群の分布を解析する。すなわち、同一平面上に存在するか否かを大まかに推定してデータ点群を抽出し、抽出したデータ点群におけるデータ点の分布の偏りがあるか否かを解析して同一平面上に存在しているか否かを再度推定する。

本変形例においては、このデータ点の分布を解析し、データ点群が後述するジグザグ形(zig-zag-shape)に当てはまる場合には、分布に偏りがあるとしてデータ点群を分割する処理を行い、これを繰り返すことによって、データ点群に含まれるノイズに対して適応的に線分の抽出を行うアルゴリズムを使用するものとする。

図44は、線分抽出部2aにおける処理、すなわち、図43におけるステップS72の処理の詳細を示すフローチャートである。まず、線分抽出部2aには、距離データが入力される。入力された距離データのうち、例えば行方向の画素列(データ点列)において、3次元空間上で同一平面上に存在すると推定されるデータ点群を抽出する。3次元空間上で同一平面上に存在すると推定されるデータ点群は、例えば隣接するデータ点の距離が、例えば6cm以下など、データ点間の3次元空間における距離が所定の閾値以下のものからなるデータ点の集合などとして行うことができ、これをデータ点群($P[0 \cdots n-1]$)として抽出する(ステップS81)。そして、このデータ点群 $P[0 \cdots$

$n-1$]に含まれるサンプル数 n が処理に最低限必要なサンプル数(必要最小値) $\min_n$ より多いか否かをチェックし(ステップS82)、データ数 n が必要最小値 $\min_n$ より少ない場合(S82:YES)には、検出結果として空集合を出力して処理を終了する。

一方、サンプル数 n が必要最小値 $\min_n$ 以上である場合(S82:NO)、データ点群 $P[0 \cdots n-1]$ の一方の端点 $P[0]$ と他方の端点 $P[n-1]$ とを結ぶ線分(弦) $L1$ を第1の線分として生成する。そして、データ点群 $P[0 \cdots n-1]$ から、この線分 $L1$ との距離が最も大きいデータ点を着目点 brk として検索し、その距離 $dist$ を算出する(ステップS83)。最大距離 $dist$ がデータ点群分割の閾値 $\max_d$ より大きい場合には(S84:YES)、データ点群 $P[0 \cdots n-1]$ を着目点(分割点) brk にて2つのデータ点群 $P[0 \cdots brk]$ 及び $P[brk \cdots n-1]$ に分割する(ステップS88)。

一方、最大距離 $dist$ がデータ点群分割の閾値 $\max_d$ より小さい場合には(S84:NO)、データ点群 $P[0 \cdots n-1]$ から後述する最小二乗法によって最適な線分の方程式 $line$ を求め(ステップS85)、この方程式 $line$ が示す線分 $L2$ を第2の線分として生成する。そして、データ点群 $P[0 \cdots n-1]$ がこの線分 $L2$ に対して後述するZig-Zag-Shapeであるかどうかを調べる(ステップS86)、Zig-Zag-Shapeでない場合(S86:NO)、得られた線分の方程式 $line$ を線分抽出結果リストに追加し(ステップS87)、処理を終了する。

また、ステップS86においてステップS85で求めた線分がZig-Zag-Shapeである判断された場合(S86:YES)、上述のステップS84と同様、ステップS88に進み、ステップS83において距離 $dist$ を求めた着目点 brk にてデータ点群を2つのデータ点群 $P[0 \cdots brk]$ 及び $P[brk \cdots n-1]$ に分割する。このステップS88にて2つのデータ点群が得られた場合には、それぞれを再帰的に再度ステップS81からの処理を行う。そして、この処理を分割された全てのデータ点について分割されなくなるまで、すなわち全てのデータ点群がステップS87を経るまで処理を繰り返し、これにより、全ての線分が登録された線分抽出結果リストを得る。このような処理によって、データ点群 $P[0 \cdots n-1]$ からノイズの影響を排除し複数の線分からなる線分群を精度よく検出することができる。

なお、ステップS83にてデータ点群 $P[0 \cdots n-1]$ の端点を結ぶ線分 $L1$ を生成するも

のとして説明したが、例えばデータ点群 $P[0 \cdots n-1]$ の分布、性質など必要に応じてデータ点群 $P[0 \cdots n-1]$ から最小二乗により線分 $L1$ を求めてもよい。また、本変形例においては、着目点 brk は、端点を結んだ線分 $L1$ との距離が最大の点1つとしているが、例えば、上記のように最小二乗により求めた線分との距離が最大の点としたり、距離がデータ点群分割の閾値 max_d 以上のものが複数ある場合はそれら全ての点又は選択した1つ以上にてデータ点群 $P[0 \cdots n-1]$ を分割するようにしてもよい。

次に、ステップS85における最小二乗による線分生成方法(Least-Squares Line Fitting)について説明する。ある n 個のデータ点群 $P[0 \cdots n-1]$ が与るとき、データ点群に最もフィットした直線の方程式を求める方法を示す。直線の方程式のモデルを下記式(1)で表す。

[数1]

$$x \cos \alpha + y \sin \alpha + d = 0 \quad \cdots(1)$$

この場合、 n 個のデータ点群 $P[0 \cdots n-1]$ の1点 (x_i, y_i) において、直線方程式のモデルとデータ点との誤差の総和は下記式(2)で表すことができる。

[数2]

$$E_{fit} = \sum_i (x_i \cos \alpha + y_i \sin \alpha + d)^2 \quad \cdots(2)$$

データ点群に最もフィットした直線は、上記式(2)の誤差の総和を最小化することによって求められる。上記式(2)を最小にする α 及び d は、データ点群 P の平均及び分散共分散行列を用いて下記(3)のように求めることができる。

[数3]

$$\alpha = \frac{1}{2} \tan^{-1} \frac{-2S_{xy}}{S_{y^2} - S_{x^2}}, \quad d = -(\bar{x} \cos \alpha + \bar{y} \sin \alpha) \quad \cdots(3)$$

ただし、

$$\begin{aligned} \bar{x} &= \frac{1}{n} \sum_i x_i & \bar{y} &= \frac{1}{n} \sum_i y_i \\ S_{x^2} &= \sum_i (x_i - \bar{x})^2 & S_{y^2} &= \sum_i (y_i - \bar{y})^2 \\ S_{xy} &= \sum_i (x_i - \bar{x})(y_i - \bar{y}) \end{aligned}$$

次に、ステップS86におけるジグザグ形(Zig-Zag-Shape)判別方法について説明する。このZig-Zag-Shape判別では、ある n 個のデータ点群 $P[0 \cdots n-1]$ と直線 $Line(\alpha, d)$ 、 $x \cos \alpha + y \cos \alpha + d = 0$ が与えられたとき、そのデータ点群 $P[0 \cdots n-1]$ が、図45Aに示すように直線 $Line$ に対して交差するか、図45Bに示すように、例えばノイズなどの影響によりデータ点が一様に分布しているかを判別するものである。基本的には、直線 $Line$ の一方にデータ点群 $P[0 \cdots n-1]$ が連続して現れる数をカウントし、ある一定数を超えて連続して現れる場合には、zig-zag-shapeであると判断することができる。図45Aの場合には、データ点群 $P[0 \cdots n-1]$ によりよくフィットする直線 $Line$ を求めるためにデータ点群 $P[i]$ を分割する必要がある。図46は、Zig-Zag-Shape判別方法を示すフローチャートである。

まず、データ点群 $P[0 \cdots n-1]$ と直線 $Line(\alpha, d, \sigma)$ とを入力する(ステップS90)。ここで、 σ は、点列の標準偏差を示す。次に、この標準偏差 σ が所定の閾値 th_σ より大きいかな否かを判断する。この標準偏差 σ が閾値 th_σ より小さい場合(ステップS91:No)は、演算器の浮動小数点演算誤差による誤差検出の影響を回避するため、判別を終了する。そして、標準偏差 σ が閾値 th_σ より大きい場合のみ判別処理を継続する。次に、データ点群 $P[0 \cdots n-1]$ のうちの最初のデータ点 $P[0]$ が直線のどちら側にあるかを $\text{sing}(\text{sdist}(P[0]))$ によって判断し、この結果を val_0 に代入すると共に val_0 と同じ側にあるデータ点の連続数をカウントするカウンタ(以下、連続点カウンタといい、このカウント値をカウント値countという。)のカウント値countを1に設定する(ステップS92)。ここで、 $\text{sign}(x)$ は、 x の値の符号(+又は-)を返す関数であり、 $\text{sdist}(i)$ は、 $P[i] \cdot x \cos \alpha + P[i] \cdot y \cos \alpha + d$ として計算された直線 $Line$ において、 i 番目のデータ点との正負の距離を示す。すなわち、 Val_0 には、データ点 $P[0]$ が直線 $Line$ のどちら側にあるかで+又は-の符号が代入される。

次に、データ点をカウントするためのカウンタ(以下、データ点カウンタといい、このカウント値をカウント値 i という。)のカウント値 i を1とする(ステップS93)。そして、データ点カウンタのカウント値 i がデータ数 n より小さい場合(ステップS94:YES)、その次のデータ(以下、 i 番目とする。)のデータ点であるデータ点 $P[i]$ が直線のどちら側にあるかを $\text{sing}(\text{sdist}(P[i]))$ によって判断し、この結果を val に代入する(ステップS95)。

そして、ステップ92にて求めた val_0 とステップS95にて求めた val_0 とを比較し、 val_0 と val_0 とが異なる場合(ステップS96:NO)、 val_0 に val を代入し、連続点カウンタのカウンタ値countに1を代入し(ステップS98)、データ点カウンタのカウンタ値 i をインクリメントして(ステップS100)ステップS94からの処理に戻る。

一方、ステップS96において、 val_0 と val_0 が同じ場合(ステップS96:YES)、データ点 $P[i-1]$ と $P[i]$ は、直線Lineに対して同じ側にあると判断され、連続点カウンタのカウンタ値countを1つインクリメントする(ステップS97)。更に、連続点カウンタのカウンタ値countがZig-Zag-Shapeと判定されるための最小のデータ点数 min_c より大きいかな否か判定し(ステップS99)、大きい場合には(ステップS99:YES)、Zig-Zag-Shapeと判断し、TRUEを出力して処理を終了する。一方、連続点カウンタのカウンタ値countが最小のデータ点数 min_c より小さい場合には(ステップS99:NO)、ステップS100に進み、データ点カウンタのカウンタ値 i をインクリメントして(ステップS100)、ステップS94からの処理を繰り返す。

そして、このステップS94からの処理を、データ点カウンタのカウンタ値 i がデータ点数 n に到達するまで続け、カウンタ値 $i \geq n$ となったところで、FALSEを出力して処理を終了する。

このようなジグザク形判別処理によって、 n 個のデータ点群 $P[0 \cdots n-1]$ と直線Line(α, d): $x \cos \alpha + y \cos \alpha + d = 0$ が与えられたとき、このデータ点群が直線Lineに対してzig-zagに交差しているかどうかを判断することができる。これによって、上述したように、ステップS86にてデータ点群を分割するべきかどうかを判断することができ、最小二乗により求めた直線に対し、データ点群がzig-zagに交差していると判断した場合にはデータ点群を分割すべきと判断してステップS88の処理へ進み、着目点brkを分割点としてデータ点群を分割することができる。なお、上記ステップS91〜ステップS100までの処理は図47のように表現することも可能である。

また、このようなZig-Zag-Shape判別処理は、演算器のみならずハードウェアで行うことも可能である。図48は、Zig-Zag-Shape判別処理を行う処理部を示すブロック図である。図48に示すように、Zig-Zag-Shape判別処理部20は、 n 個のデータ点群 $P[0 \cdots n-1]$ が入力され、順次各データ点 $P[i]$ が直線Lineの何れ側に位置するかを判別

し、その判別結果Valを出力する方向判別部21と、1つ後のデータと方向判別部21の結果を比較させるための遅延部22と、データ点P[i]における方向判別結果Valとデータ点P[i-1]における方向判別結果Val₀とを比較する比較部23と、比較部23においてVal=Val₀の場合に、カウント値をインクリメントする連続点カウンタ24と、連続点カウンタ24のカウント値countと最小データ点数格納部26から読み出した最小データ点数min_cとを比較する比較部25とを有する。

このZig-Zag-Shape判別処理部における動作は以下のようになる。すなわち、方向判別部21は、データ点群P[0...n-1]から最小二乗法により直線Lineを求め、各データ点P[i]と直線Lineとの正負の距離を求め、その正負の符号を出力する。遅延部22は、データ点P[i-1]の直線Lineまでの距離に対する正負の符号が入力されると1つ後のデータ点P[i]の正負の符号が入力されるタイミングまでデータを格納する。

比較部23は、データ点P[i]とデータ点P[i-1]の上記正負の符号を比較し、同じ符号である場合にはカウンタ24のカウント値countをインクリメントする信号を出力し、正負の符号が異なればカウント値countに1を代入する信号を出力する。比較部25は、カウント値countと最小データ点数min_cとを比較し、最小データ点数min_cよりカウント値countが大きい場合には、データ点群P[0...n-1]がジグザグであることを示す信号を出力する。

次に、図40に示す領域拡張部 (Region Growing) 2bについて説明する。領域拡張部2bは、線分抽出部2aによって得られた線分群を入力とし、それらの線分それぞれがどの平面に属しているかを点列の平面への当てはめ (Plane Fitting) により判断し、与えられる線分群からなる領域を複数の平面 (平面領域) に分離する。複数の平面に分離するために、以下の手法をとる。

まず、与えられた線分群から、同じ平面上にあると推定される隣接する3本の線分を検索する。この3本の線分により求められる平面 (基準平面) が、平面の種となるものであり、この3本の線分が含まれる領域を領域種 (seed region) という。そして、この領域種に隣接する線分を順次、基準平面と同一平面上にある線分か否かを点列の平面への当てはめ (Plane Fitting) により判断し、隣接する線分が同じ平面に含まれると判断された場合には、この線分を領域拡大用の線分として領域種に追加してその

領域を拡大すると共に、基準平面の方程式を上記領域拡大用の線分を含めて再度算出し直す。このような処理によって、全ての線分を何れかの領域(平面)に配分する。

図49は、領域拡張処理を説明するための模式図である。図49に示すように、画像30内に複数の平面からなる階段31が存在する場合、例えば太線で示す32a〜32cの3本の線分が領域種として選択されたとする。これら3本の線分32a〜32cからなる領域が領域種となる。先ず、この3つの線分32a〜32cにより1つの平面(基準平面)Pを求める。次に、領域種の最も外側の線分32a又は32cに領域種外にて隣接するそれぞれデータ列33又は34において、平面Pと同一の平面である線分を選択する。ここでは、線分33aが選択されたとする。次に、これら4本の線分群からなる平面P'を求め、基準平面Pを更新する。次に、線分34aが選択されれば、5本の線分群からなる平面P''を求め、平面P'を更新する。これを繰り返すことにより、階段31の2段目の踏面が、破線で囲まれる平面45として求められる。このようにして、選択された領域種を種として追加する線分がなくなるまで領域拡大処理する。そして、追加する線分がなくなった場合、再び画像30内から領域種となる3つの線分を検索して領域拡大処理を実行するというような処理を繰り返し、領域種となる3つの線分がなくなるまで図43のステップS3の処理を繰り返す。

次に、データ点群 $P[0 \cdots n-1]$ から構成される平面の方程式を推定する手法(Plane Fitting)、これを使用して領域種を選択する方法(Selection of seed region)、領域種から領域を拡大していく領域拡張処理(Region growing)、及び得られた平面方程式から誤差が大きいものなどを除いて再度算出する後処理(Post processing)について説明する。

3次元空間内の点Pは $P = (x_i, y_i, z_i)$ により表され、平面の方程式はその法線ベクトル $n(n_x, n_y, n_z)$ と非負の定数 d によって下記式(4)で表される。

[数4]

$$xn_x + yn_y + zn_z + d = 0 \quad \cdots(4)$$

ここで、ステレオカメラでは、焦点を通る平面を観測することができない、すなわち、

平面は焦点を通らないため、 $d \neq 0$ とすることができる。したがって、平面は、最小二乗法により下記式(5)に示す値を最小にする値として求めることができる。

[数5]

$$fit(\mathbf{n}, d) = \sum_i (\mathbf{p}_i^T \mathbf{n} + d)^2. \quad \dots(5)$$

最適解は $\mathbf{n} = \mathbf{m} / \|\mathbf{m}\|$, $d = -1 / \|\mathbf{m}\|$ として求まる。ここで、 $\|\cdot\|$ は、ベクトルの大きさ、 $\mathbf{m}$ は、行列式によって連立一次方程式を解くクラメールの法則(Cramer's rule)を使用して下記(6-1)のように容易に得られる線形システムの解である。

[数6]

$$\mathbf{A} \cdot \mathbf{m} = \mathbf{b} \quad \dots(6-1)$$

ここで、

$$\mathbf{A} = \sum_i \mathbf{p}_i \mathbf{p}_i^T, \quad \mathbf{b} = \sum_i \mathbf{p}_i \quad \dots(6-2)$$

この解は、新たなデータ点を加えられたり、又はデータ点削除されたりした場合であっても、上記式(6-2)に示す $\mathbf{A}$ と $\mathbf{b}$ の値を更新するのみで、平面パラメータを再計算することができる。更に、本変形例における線分抽出方法の場合は n 個のデータ点群の2つのモーメント(1次モーメント:平均、2次モーメント:分散) $E(\mathbf{p})$ 、 $E(\mathbf{p}\mathbf{p}_T)$ が既知であり、これらを使用して、下記(7)に示すように $\mathbf{A}$ 、 $\mathbf{b}$ を更新することができ、 n 個のデータ点群における平面更新処理に拡張することができる。

[数7]

$$\mathbf{A} \leftarrow \mathbf{A} + nE(\mathbf{p}\mathbf{p}_T), \quad \mathbf{b} \leftarrow \mathbf{b} + nE(\mathbf{p}) \quad \dots(7)$$

また、一度平面パラメータ $\mathbf{n}$ 、 d を算出すれば、求まった平面方程式から、 n 個のデータ点群の平面方程式からの外れ度合いを示す平面方程式の2乗平均誤差(RMS (root mean square) residual) (以下、rmsという。)を下記式(8)により算出することができる。この場合も、 n 個のデータ点の上記2つのモーメントを使用して下記式(8)を求めることができる。

[数8]

$$rms(p_1 \cdots p_n) = \sqrt{\frac{1}{n} \sum_i (p_i^T n + d)^2} > th_rms \quad \cdots (8)$$

上記(8)に示すように、各データ点が求めた平面上にあれば平面方程式の2乗平均誤差 $rms(p_1 \cdots p_n)$ は0になる値であり、この値が小さいほど各データ点が平面によくフィットしていることを示す。

次に、領域種(seed region)を検索する方法及び領域種から領域を拡大すると共に平面を更新する方法について説明する。図50は、領域種を検索する処理及び領域拡張処理の手順を示すフローチャートである。図50に示すように、領域種を選択には、先ず、線分抽出の際に使用した行方向又は列方向のデータ列が隣接する3つの線分 (l_1, l_2, l_3) であって、互いの線分 (l_1, l_2) , (l_2, l_3) における画素位置が上記データ列とは直交する方向にて重複したものを検索する(ステップS101)。各データ点は画像内における画素位置を示すインデックス(index)を有しており、例えば行方向のデータ列における線分である場合、このインデックスを比較して列方向にて重複しているか否かを比較する。この検索に成功した場合(ステップS102: YES)、上記式(7)を使用して上記(6-1)を算出する。これにより、平面パラメータ n, d を決定でき、これを使用して上記式(8)に示す平面方程式の2乗平均誤差 $rms(l_1, l_2, l_3)$ を計算する(ステップS103)。そして、この平面方程式の2乗平均誤差 $rms(l_1, l_2, l_3)$ が例えば1cmなどの所定の閾値 th_rms より小さい場合には、この3つの線分を領域種として選択する(ステップS104)。所定の閾値 th_rms より大きい場合には、再びステップS101に戻り、上記条件を満たす線分を検索する。また、領域種に選ばれた線分は、線分群のリストから除くことで、他の平面拡張などの際に使用されないようにしておく。

こうして選択された領域種から線分拡張法により領域を拡張する。すなわち、先ず、領域種の領域に追加する候補となる線分を検索する(ステップS105)。なお、この領域は、領域種が既に更新されている場合の、後述する更新された領域種も含む。候補となる線分は、領域種の領域に含まれる線分(例えば l_1)に隣接する線分(l_4)であって、上述同様、これらの線分の画素位置が相互に重なりあうことを条件とする。検索が成功した場合(ステップS106: YES)、その平面方程式の2乗平均誤差 $rms(l_4)$ を

算出し、これが所定の閾値 th_{rms} より小さいか否かを判定し(ステップS107)、小さい場合には平面パラメータを更新し(ステップS108)、再びステップS105からの処理を繰り返す。ここで、候補となる線分がなくなるまで処理を繰り返し、候補となる線分がなくなったら(ステップS106:NO)、ステップS101の処理に戻り、再び領域種を検索する。そして、線分群に含まれる領域種がなくなった場合(ステップS102:NO)、今まで得られている平面パラメータを出力して処理を終了する。

ここで、本変形例においては、領域種を検索し、3つの線分が同一平面に属するか否かの判定、及び領域拡張処理を行う際に基準平面又はこれを更新した更新平面に属するか否かの判定には、上記式(8)を使用する。すなわち、平面方程式の2乗平均誤差rmsが所定の閾値(th_{rms})未満である場合にのみその線分(群)を同一平面に属するものと推定し、その線分を含めた平面として再び平面を算出する。このように平面方程式の2乗平均誤差rmsを使用して同一平面に属するか否かを判定することにより、更にノイズにロバストでかつ、細かい段差を含んでいるような場合にも正確に平面を抽出することができる。以下にその理由について説明する。

図51は、その効果を示す図であって、端点と直線との距離が等しくても平面方程式の2乗平均誤差rmsが異なる例を示す模式図である。ここで、非特許文献4のように、領域拡張処理する際、注目の直線(線分)の端点(end point)と平面Pとの距離Dの値が所定の閾値より小さい場合に、当該注目の線分が平面Pと同一平面であるとして領域拡張処理を行うと、平面Pに交差する直線La(図51A)と、平面Pと平行で所定距離ずれているような直線Lb(図51B)とが同様に平面Pの更新に使用されることとなる。ここで、平面方程式の2乗平均誤差rmsを求めると、図51Aの直線Laから求まる平面方程式の2乗平均誤差rms(La)に比して図51Bの直線Lbから求まる平面方程式の2乗平均誤差rms(Lb)の方が大きい。すなわち、図51Aのように、直線Laと平面Pとが交差する場合は、平面方程式の2乗平均誤差rmsが比較的小さくノイズの影響である場合が多いのに対し、図51Bのような場合、平面方程式の2乗平均誤差rmsが大きく、直線Lbは平面Pと同一平面ではなく異なる平面P'である確率が高い。したがって、複数の平面が含まれるような環境から平面を精確に求める必要がある場合などにおいては、本変形例のように、平面方程式の2乗平均誤差rmsを算出し、この

値が所定の閾値未満である場合に同一平面と判断することが好ましい。なお、環境や距離データの性質に応じて、従来と同様、線分の端点と平面との距離が所定の閾値以下の場合は当該線分を平面に含めるようにしたり、これらを組み合わせてもよい。

また、面パラメータ n, d を一旦算出すれば、平面方程式の2乗平均誤差 rms は、データ点群について線分抽出の際に求めた2つのモーメントの値から平面方程式を更新し、上記式(8)にて簡単に算出することができる。

また、上述の領域種の選択方法は、図52のようにも表現することができる。 $overlap(l_j, l_k)$ は、各イメージロウに含まれる直線ベクトル l_j と l_k における端点間の位置が直線ベクトルとは直交する位置にて重なっている場合に $true$ を出力する関数である。また、 $fitPlane(l_1, l_2, l_3)$ は、上記式(4)～(7)により $Am=b$ の解を求め平面パラメータ n, d を計算し、上記式(8)により算出された A, b により、直線ベクトル l_1, l_2, l_3 を平面にフィッティングさせる関数である。

$rms(l_1, l_2, l_3)$ は、上記式(6)を使用して3本の直線全てにおいて、平面方程式の2乗平均誤差 rms の値を算出する関数である。また、 $remove(l_1, l_2, l_3)$ は、 $lines[i], lines[i+1]l_2, lines[i+2]$ から領域種を構成するとして選択されたそれぞれ直線 l_1, l_2, l_3 を除くことを意味し、これにより、再びこれらの直線が計算に使用されることを防止する。

また、領域拡張処理は、図53のように表現することもできる。図53において、 A 及び b は、上記式(6-1)に示すそれぞれ行列及びベクトルである、また、 $add(A, b, l)$ は、上記式(8)により、 A と b に直線 $line$ のモーメントを加える関数である。 $Solve(A, b)$ は、 $Am=b$ を満たす m を求め、上記式(4)～(7)により平面パラメータ n, d を計算する。 $select(open)$ は、例えば最初の1つなど、 $open$ の中から任意に1つのエレメントを選択する関数である。また、 $index(l_1)$ は、画素列又は行における l_1 のインデックスを返す関数である。また、 $neighbor(index)$ は、与えられたインデックスに隣接したインデックス、例えば $\{index-1, index+1\}$ を返す関数である。

また、上述したように、本変形例においては、図43のステップS73において領域拡張処理を行って平面方程式を更新した後、ステップS74において平面方程式を再度算出する処理(Post processing)を行う。この再度算出する処理では、例えば上述の

ように更新され最終的に得られた平面方程式が示す平面に属するとされた距離データ点又は線分の平面からのずれを計算し、所定の値以上平面から外れる距離データ点又は線分は除き、再度平面方程式を更新することで、ノイズの影響を更に低減することができる。

次に、このステップS74について詳細に説明する。ここでは、2つのステップにより、平面方程式を再度算出する方法について説明する。まず、ステップS73にて検出された各平面の境界の距離データ点(pixels)において、現在属している平面よりも、隣接する平面までの距離が近いデータ点が検出された場合は、当該データ点を隣接する平面の方に含める処理をする。また、何れの平面にも属していなく、かつ距離が例えば1.5cmなど比較的大きい閾値以下である平面が存在するデータ点が検出できた場合は、当該データ点をその平面に含める処理をする。これらの処理は各平面領域の境界近傍のデータ点を検索することで実行することができる。以上の処理が終了したら、再度平面方程式を算出する。

次に、上述のようにして再度算出された平面の各領域の境界近傍において、各データ点と平面との距離が例えば0.75cmなど比較的小さい閾値を超える場合は、それらのデータ点を捨てる処理を実行する。これにより、その平面領域は若干小さくなるものの更に精確な平面を求めることができる。距離データ点を削除後、再び平面を求め、この処理を繰り返す。このことにより、極めて精密に平面を求めることができる。

次に各処理によって得られる結果を示す。図54Aは、ロボット装置が立った状態で床面を見下ろした際の床面を示す模式図、図54Bは、縦軸をx、横軸をy、各データ点の濃淡でz軸を表現して3次元距離データを示す図であり、更に、行方向の画素列から線分抽出処理にて同一平面に存在するとされるデータ点群から直線を検出したものを示す。図54Bに示す直線群から領域拡張処理によりえられた平面領域を図54Cに示す。このように、ロボット装置の視野内には、1つの平面(床面)のみが存在する、すなわち、床面が全て同じ平面として検出されていることがわかる。

次に、床面に段差を一段置いたときの結果を図55に示す。図55Aに示すように、床面Fには、1段の段差ST3が載置されている。図55Bは、実験条件を示す図であり、着目点と直線(線分)との距離が $\max_d$ を超える場合は、データ点群を分割する。ま

た、抽出の成否(水平) (correct extraction(horizontal))は、行方向のデータ列毎に、合計10回の線分抽出を行う線分拡張法による平面検出を行って成功した回数を示し、抽出の成否(垂直) (correct extraction(vertical))は、列方向のデータ列毎についての抽出の成否を示す。また、No. 1〜No. 5は、上述したZig-Zag-Shape判別処理を取り入れていない従来の線分拡張法による平面検出処理の条件、No. 6は、Zig-Zag-Shape判別処理を行った本変形例における平面検出方法の条件を示す。

図55C及び図55Dは、線分拡張法により平面検出した結果を示す図であって、それぞれ本変形例における手法により平面検出した結果、従来の線分拡張法により平面検出した結果(比較例)を示す。図55Bに示すように、従来の手法においては、線分抽出(Line Fitting)において推定のための閾値パラメータmax_dを大きくする(max_d=25, 30)と検出率が下がり、閾値max_d小さくする(max_d=10, 15)と検出率が向上する。これに対して、本発明のように、ジグザグ形検証処理を導入することにより、大きな閾値max_d=30を設定しても、優れた検出結果を示すことがわかる。

すなわち、閾値max_dを大きくすると、ノイズの影響が少なくなるものの、線分抽出が難しくなり、閾値max_dを小さくすると、ノイズの影響を受けて誤検出が多くなってしまう。図56Bに示す床面を撮影した画像から3次元距離データを取得した場合を図56B及び図56Cに示す。何れ左図は、行方向の画素列(距離データ列)から線分を抽出した例、右図は列方向の画素列(距離データ列)から線分を抽出した例を示す。図56Bに示すように、閾値max_dを小さくすると、ノイズの影響が大きくなり、ノイズの影響が大きい遠方などにおいては特に、線分をうまく検出することができない。一方、図56Cに示すように、従来の線分抽出に更にジグザグ形判別処理を加えた場合、閾値max_dを大きくしても、更にノイズの影響が大きい遠方の領域であっても線分が検出されていることがわかる。

これにより、上述した如く、それぞれ異なる階段を撮影した画像から3次元距離データを取得して平面検出することができる。例えば、図11及び図12に示すように、何れの場合も全ての階面を平面として検出できている。なお、図12Bでは、床面の一部も他の平面として検出成功している。

本変形例によれば、線分拡張法による平面検出を行う際、始めは大きな閾値を設

定して線分を分割し、次にZig-Zag-Shape判別処理により、閾値を超えるデータ点を持たない直線であってもジグザグ形である場合には、ノイズではなく、複数平面からなる直線であるとして線分を分割するようにしたので、ノイズを含む距離情報から複数の平面を精度よく検出することが可能となる。

このように、小さい段差も精度よく検出することができるため、例えばロボット装置が移動可能な環境内の階段などを認識することができ、二足歩行ロボット装置であれば、この検出結果を利用して階段昇降動作が可能となる。

更に、複数の平面によって構成されている凸凹の床面を歩行可能な平面だと誤認識することがなくなり、ロボット装置の移動などが更に簡単になる。

なお、本発明は上述した変形例のみに限定されるものではなく、本発明の要旨を逸脱しない範囲において種々の変更が可能であることは勿論である。また、上述した平面検出処理、階段認識処理、階段昇降制御処理のうち1以上の任意の処理は、ハードウェアで構成しても、演算器(CPU)にコンピュータプログラムを実行させることで実現してもよい。コンピュータプログラムとする場合には、記録媒体に記録して提供することも可能であり、また、インターネットその他の伝送媒体を介して伝送することにより提供することも可能である。

請求の範囲

- [1] 1. 移動手段により移動可能なロボット装置において、
3次元の距離データから環境内に含まれる1又は複数の平面を検出し、平面情報として出力する平面検出手段と、
上記平面情報から移動可能な平面を有する階段を認識し、該階段の踏面に関する踏面情報及び蹴り上げ情報を有する階段情報を出力する階段認識手段と、
上記階段情報に基づき、階段昇降可能か否かを判断し、昇降動作が可能であると判断した場合には、その踏面に対して自律的に位置決めして階段昇降動作を制御する階段昇降制御手段を有する
ことを特徴とするロボット装置。
- [2] 2. 上記3次元の距離データを取得する距離計測手段を有する
ことを特徴とする請求の範囲第1項記載のロボット装置。
- [3] 3. 上記移動手段として脚部を備え、この脚部により移動可能である
ことを特徴とする請求の範囲第1項記載のロボット装置。
- [4] 4. 上記階段認識手段は、
与えられた平面情報から移動可能な平面を有する階段を検出して統合前階段情報を出力する階段検出手段と、
上記階段検出手段から出力される時間的に異なる複数の統合前階段情報を統計的に処理することにより統合した統合済階段情報を上記階段情報として出力する階段統合手段とを有する
ことを特徴とする請求の範囲第1項記載のロボット装置。
- [5] 5. 上記階段検出手段は、上記平面情報に基づき踏面の大きさ及び空間的な位置を認識し、この認識結果である踏面情報を上記統合前階段情報として出力し、
上記階段統合手段は、時間的に前後する踏面情報から、所定の閾値より大きい重複領域を有しかつ相対的な高さの違いが所定の閾値以下である2以上の踏面からなる踏面群を検出した場合、当該踏面群を何れをも含む一の踏面となるよう統合する
ことを特徴とする請求の範囲第4項記載のロボット装置。
- [6] 6. 上記階段認識手段は、上記平面情報に基づき踏面の大きさ及び空間的な位置を

認識し上記踏面情報とする

ことを特徴とする請求の範囲第1項記載のロボット装置。

- [7] 7. 上記踏面情報は、少なくとも移動方向に対して該踏面の手前側の境界を示すフロントエッジ及び奥側の境界を示すバックエッジの情報を含む

ことを特徴とする請求の範囲第6項記載のロボット装置。

- [8] 8. 上記踏面情報は、上記フロントエッジ及びバックエッジに挟まれた領域である安全領域の左右両側に隣接した領域であって移動可能である確率が高いと推定されるマージン領域を示す右側マージン情報及び左側マージン情報を有する

ことを特徴とする請求の範囲第7項記載のロボット装置。

- [9] 9. 上記踏面情報は、上記平面情報に基づき踏面と推定された領域の重心を示す参照点情報を有する

ことを特徴とする請求の範囲第7項記載のロボット装置。

- [10] 10. 上記参照点情報は、上記フロントエッジ及びバックエッジに挟まれた領域を示す安全領域の重心、該安全領域及びその両側に隣接した領域であって移動可能である確率が高いと推定されるマージン領域からなる踏面領域の重心、又は踏面となる平面を構成する点群から求まる中心点の何れかからなる参照点の位置情報を有する

ことを特徴とする請求の範囲第9項記載のロボット装置。

- [11] 11. 上記踏面情報は、踏面となる平面を構成する点群の3次元座標情報を有する

ことを特徴とする請求の範囲第7項記載のロボット装置。

- [12] 12. 上記階段認識手段は、上記平面情報に基づき平面の境界を抽出して多角形を算出し、該多角形に基づき上記踏面情報を算出する

ことを特徴とする請求の範囲第1項記載のロボット装置。

- [13] 13. 上記多角形は、上記平面情報に基づき抽出された平面の境界に外接する凸多角形領域である

ことを特徴とする請求の範囲第12項記載のロボット装置。

- [14] 14. 上記多角形は、上記平面情報に基づき抽出された平面の境界に内接する凸多角形領域である

ことを特徴とする請求の範囲第12項記載のロボット装置。

- [15] 15. 上記多角形は、上記平面情報に基づき抽出された平面の境界を平滑化した非凸多角形領域であることを特徴とする請求の範囲第12項記載のロボット装置。
- [16] 16. 上記平面情報は、一の平面毎に、法線ベクトル、境界を示す境界情報、重心位置を示す重心情報、大きさを示す面積情報、及び平面度から選択される1以上の情報を有する
ことを特徴とする請求の範囲第1項記載のロボット装置。
- [17] 17. 上記階段昇降制御手段は、現在移動中の移動面におけるバックエッジに対峙した所定位置に移動した後、昇降動作を実行するよう制御する
ことを特徴とする請求の範囲第3項記載のロボット装置。
- [18] 18. 上記階段昇降制御手段は、現在移動中の移動面におけるバックエッジが確認できない場合は、次に昇降動作の対象となる次段の踏面におけるフロントエッジに対峙した所定位置に移動した後、昇降動作を実行するよう制御する
ことを特徴とする請求の範囲第17項記載のロボット装置。
- [19] 19. 上記階段昇降制御手段は、次に移動対象となる踏面を検出し、当該移動対象となる踏面に対峙した所定位置に移動する一連の動作を行って昇降動作を実行するよう制御する
ことを特徴とする請求の範囲第3項記載のロボット装置。
- [20] 20. 上記階段昇降制御手段は、現在位置から次に移動対象となる次段又は次段以降の踏面が検出できない場合、過去に取得した階段情報から当該移動対象となる次段の踏面を検索する
ことを特徴とする請求の範囲第19項記載のロボット装置。
- [21] 21. 上記階段昇降制御手段は、現在の移動面におけるバックエッジに対峙した所定位置に移動した後、次の移動対象となる踏面を検出し、当該踏面におけるフロントエッジに対峙した所定位置に移動し、当該踏面に移動する昇降動作を実行するよう制御する
ことを特徴とする請求の範囲第3項記載のロボット装置。
- [22] 22. 上記階段昇降制御手段は、踏面に対する上記移動手段の位置を規定したパラメータを使用して昇降動作を制御する

ことを特徴とする請求の範囲第3項記載のロボット装置。

- [23] 23. 上記パラメータは、上記脚部の足上げ高さ又は足下げ高さに基づき決定されることを特徴とする請求の範囲第22項記載のロボット装置。

- [24] 24. 階段を登る動作と降りる動作とで上記パラメータの数値を変更するパラメータ切り替え手段を有する

ことを特徴とする請求の範囲第22項記載のロボット装置。

- [25] 25. 上記平面検出手段は、3次元空間で同一平面上にあると推定される距離データ点群毎に線分を抽出する線分抽出手段と、上記線分抽出手段によって抽出された線分群から同一平面に属すると推定される複数の線分を抽出し該複数の線分から平面を算出する平面領域拡張手段とを有し、

上記線分抽出手段は、距離データ点の分布に応じて適応的に線分を抽出することを特徴とする請求の範囲第1項記載のロボット装置。

- [26] 26. 上記線分抽出手段は、上記距離データ点間の距離に基づき同一平面上にあると推定される距離データ点群を抽出し、該距離データ点群における距離データ点の分布に基づき、当該距離データ点群が同一平面上にあるか否かを再度推定することを特徴とする請求の範囲第25項記載のロボット装置。

- [27] 27. 上記線分抽出手段は、上記同一平面上にあると推定される距離データ点群から線分を抽出し、該距離データ点群のうち該線分との距離が最も大きい距離データ点を着目点とし、当該距離が所定の閾値以下である場合に該距離データ点群における距離データ点の分布に偏りがあるか否かを判別し、該偏りがある場合には該着目点にて該距離データ点群を分割する

ことを特徴とする請求の範囲第25項記載のロボット装置。

- [28] 28. 上記線分抽出手段は、上記同一平面上にあると推定される距離データ点群から第1の線分を抽出し、該距離データ点群のうち該第1の線分との距離が最も大きい距離データ点を着目点とし、当該距離が所定の閾値以下である場合に該距離データ点群から第2の線分を抽出し、該第2の線分の一方側に距離データ点が所定の数以上連続して存在するか否かを判定し、所定の数以上連続して存在する場合に該距離データ点群を該着目点にて分割する

ことを特徴とする請求の範囲第25項記載のロボット装置。

- [29] 29. 上記平面領域拡張手段は、同一の平面に属すると推定される1以上の線分を選択して基準平面を算出し、該基準平面と同一平面に属すると推定される線分を該線分群から拡張用線分として検索し、該拡張用線分により該基準平面を更新すると共に該基準平面の領域を拡張する処理を繰り返し、更新が終了した平面を更新済平面として出力する

ことを特徴とする請求の範囲第25項記載のロボット装置。

- [30] 30. 上記更新済平面に属する距離データ点群において、当該更新済平面との距離が所定の閾値を超える距離データ点が存在する場合、これを除いた距離データ点群から再度平面を算出する平面再算出手段を更に有する

ことを特徴とする請求の範囲第29項記載のロボット装置。

- [31] 31. 上記平面領域拡張手段は、線分により定まる平面と上記基準平面との誤差に基づき当該線分が該基準平面と同一平面に属するか否かを推定する

ことを特徴とする請求の範囲第25項記載のロボット装置。

- [32] 32. 移動手段により移動可能なロボット装置の動作制御方法において、
3次元の距離データから環境内に含まれる1又は複数の平面を検出し、平面情報として出力する平面検出工程と、

上記平面情報から移動可能な平面を有する階段を認識し、該階段の踏面に関する踏面情報及び蹴り上げ情報を有する階段情報を出力する階段認識工程と、

上記階段情報に基づき、階段昇降可能か否かを判断し、昇降動作が可能であると判断した場合には、その踏面に対して自律的に位置決めして階段昇降動作を制御する階段昇降制御工程と

を有することを特徴とするロボット装置の動作制御方法。

- [33] 33. 上記3次元の距離データを取得する距離計測工程を有する

ことを特徴とする請求の範囲第32項記載のロボット装置の動作制御方法。

- [34] 34. 上記移動手段は、機体脚部である

ことを特徴とする請求の範囲第32項記載のロボット装置の動作制御方法。

- [35] 35. 上記階段認識工程は、

与えられた平面情報から移動可能な平面を有する階段を検出して統合前階段情報を出力する階段検出工程と、

上記階段検出工程にて出力される時間的に異なる複数の統合前階段情報を統計的に処理することにより統合した統合済階段情報を上記階段情報として出力する階段統合工程とを有する

ことを特徴とする請求の範囲第32項記載のロボット装置の動作制御方法。

- [36] 36. 上記階段検出工程では、上記平面情報に基づき踏面の大きさ及び空間的な位置を認識し、この認識結果である踏面情報を上記統合前階段情報として出力し、

上記階段統合工程では、時間的に前後する踏面情報から、所定の閾値より大きい重複領域を有しかつ相対的な高さの違いが所定の閾値以下である2以上の踏面からなる踏面群を検出した場合、当該踏面群を何れも含む一の踏面となるよう統合する

ことを特徴とする請求の範囲第35項記載のロボット装置の動作制御方法。

- [37] 37. 上記階段認識工程では、上記平面情報に基づき踏面の大きさ及び空間的な位置を認識し上記踏面情報とする

ことを特徴とする請求の範囲第32項記載のロボット装置の動作制御方法。

- [38] 38. 上記踏面情報は、少なくとも移動方向に対して該踏面の手前側の境界を示すフロントエッジ及び奥側の境界を示すバックエッジを示す情報を含む

ことを特徴とする請求の範囲第37項記載のロボット装置の動作制御方法。

- [39] 39. 上記階段認識工程では、上記平面情報に基づき平面の境界を抽出して多角形を算出し、該多角形に基づき上記踏面情報を算出する

ことを特徴とする請求の範囲第32項記載のロボット装置の動作制御方法。

- [40] 40. 上記階段昇降制御工程では、現在移動中の移動面におけるバックエッジに対峙した所定位置に移動した後、昇降動作を実行するよう制御する

ことを特徴とする請求の範囲第34項記載のロボット装置の動作制御方法。

- [41] 41. 上記階段昇降制御工程では、現在移動中の移動面におけるバックエッジが確認できない場合は、次に昇降動作の対象となる次段の踏面におけるフロントエッジに対峙した所定位置に移動した後、昇降動作を実行するよう制御する

ことを特徴とする請求の範囲第40項記載のロボット装置の動作制御方法。

- [42] 42. 上記階段昇降制御工程では、次に移動対象となる踏面を検出し、当該踏面に対峙した所定位置に移動する一連の動作を行って昇降動作を実行するよう制御することを特徴とする請求の範囲第34項記載のロボット装置の動作制御方法。
- [43] 43. 上記階段昇降制御工程では、現在位置から次に移動対象となる次段又は次段以降の踏面が検出できない場合、過去に取得した階段情報から当該移動対象となる次段の踏面を検索することを特徴とする請求の範囲第42項記載のロボット装置の動作制御方法。
- [44] 44. 上記階段昇降制御工程では、現在の移動面におけるバックエッジに対峙した所定位置に移動した後、次の移動対象となる踏面を検出し、当該踏面におけるフロントエッジに対峙した所定位置に移動し、当該踏面に移動する昇降動作を実行するよう制御することを特徴とする請求の範囲第34項記載のロボット装置の動作制御方法。
- [45] 45. 上記階段昇降制御工程では、踏面に対する移動手段の位置を規定するためのパラメータを使用して昇降動作を制御することを特徴とする請求の範囲第34項記載のロボット装置の動作制御方法。
- [46] 46. 階段を登る動作と降りる動作とで上記パラメータの数値を変更するパラメータ切り替え工程を有することを特徴とする請求の範囲第45項記載のロボット装置の動作制御方法。
- [47] 47. 移動手段により移動可能な移動装置において、
3次元の距離データから環境内に含まれる1又は複数の平面を検出し、平面情報として出力する平面検出手段と、
上記平面情報から移動可能な平面を有する階段を認識し、該階段の踏面に関する踏面情報及び蹴り上げ情報を有する階段情報を出力する階段認識手段と、
上記階段情報に基づき、階段昇降可能か否かを判断し、昇降動作が可能であると判断した場合には、その踏面に対して自律的に位置決めして階段昇降動作を制御する階段昇降制御手段とを有することを特徴とする移動装置。

[図1]

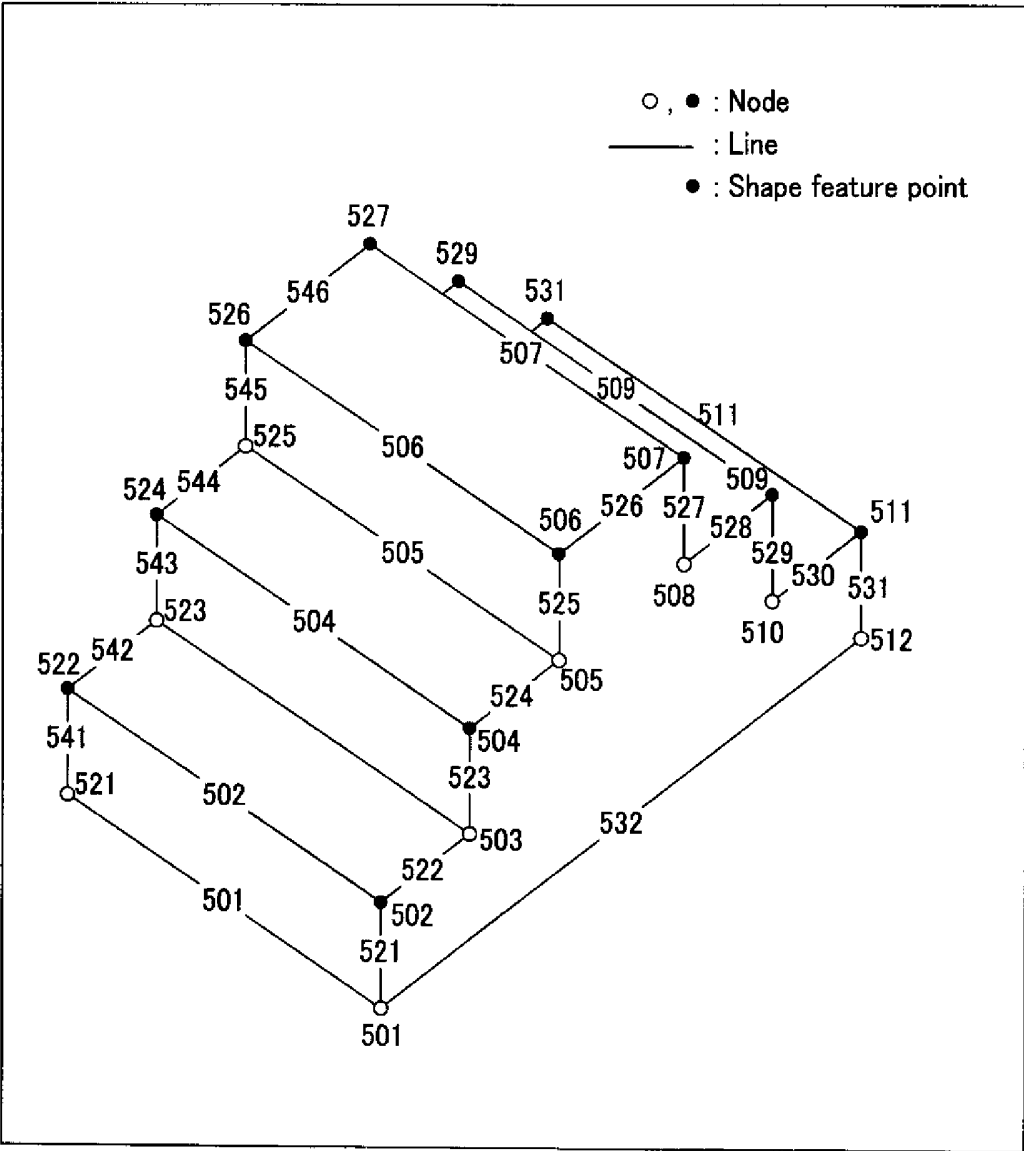
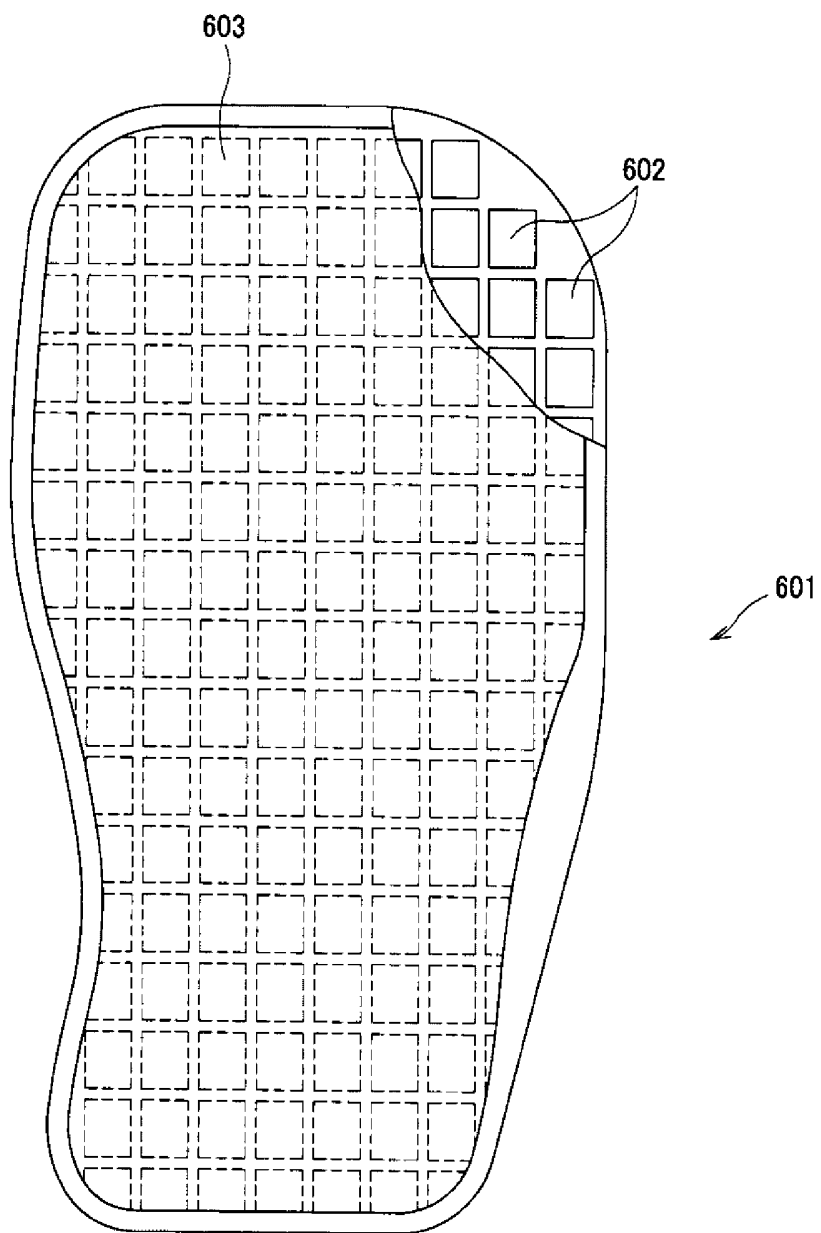


FIG. 1

[図2]

**FIG.2**

[図3]

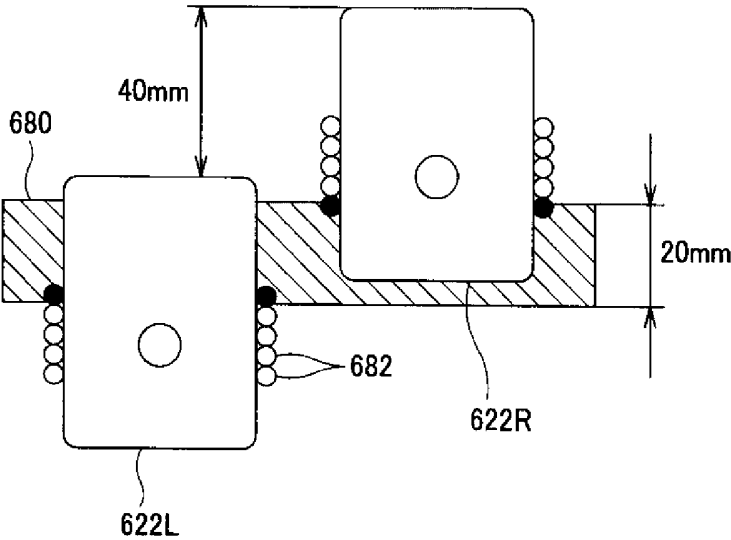


FIG.3A

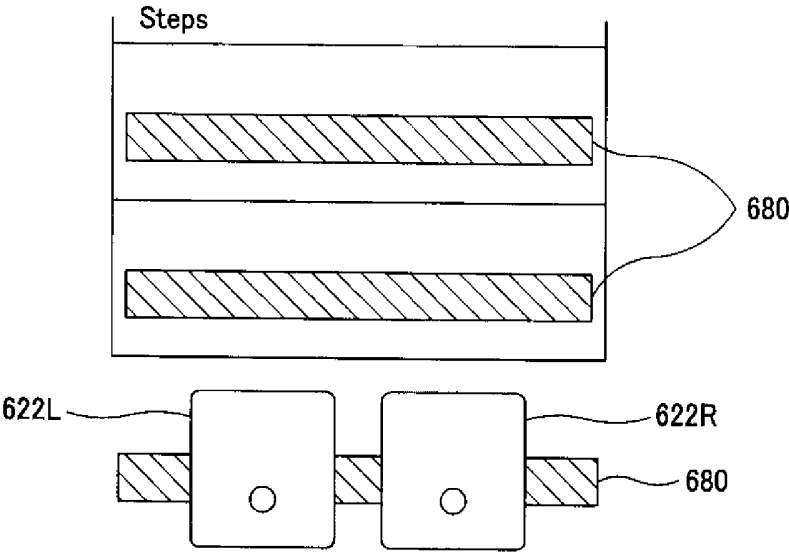
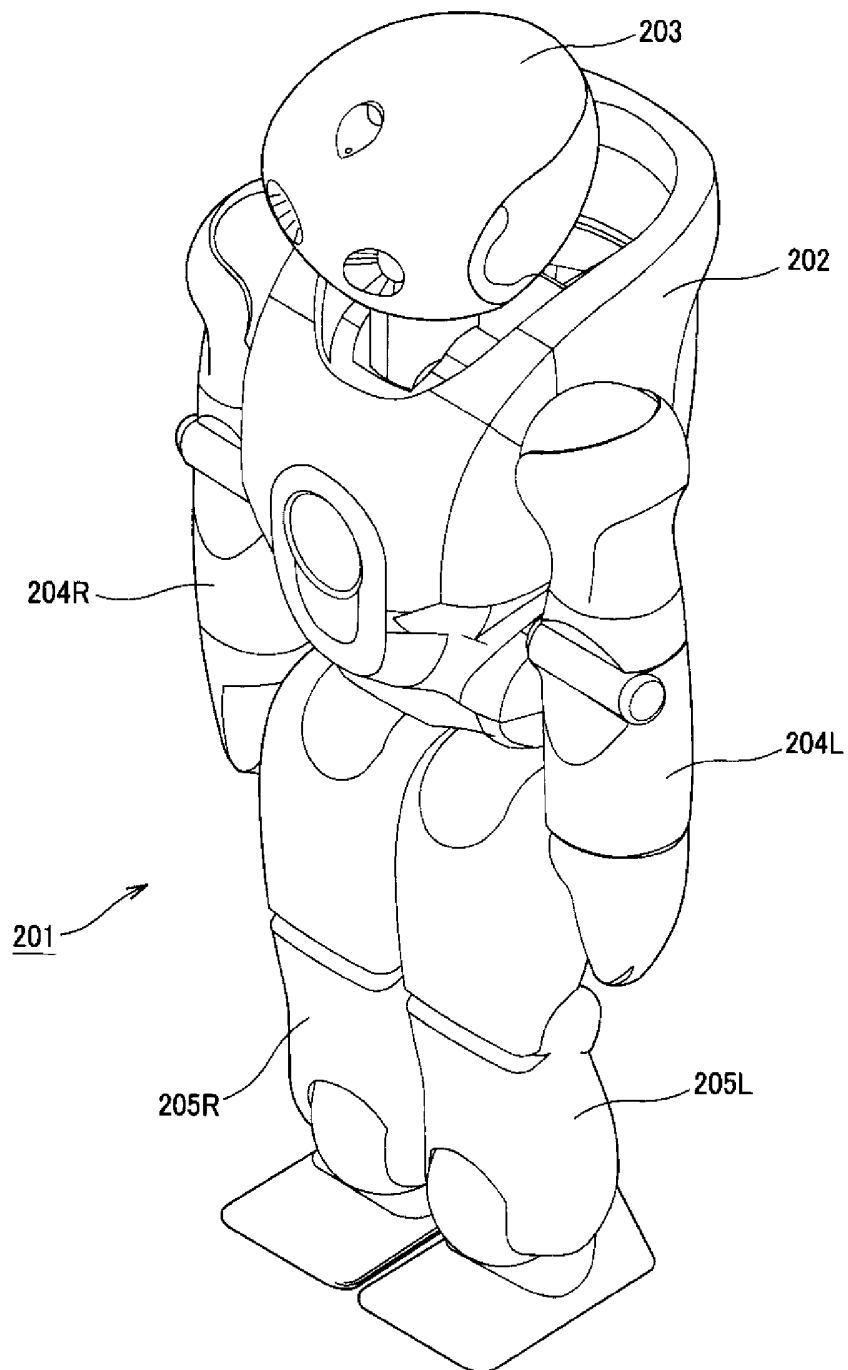


FIG.3B

[図4]

**FIG. 4**

[図5]

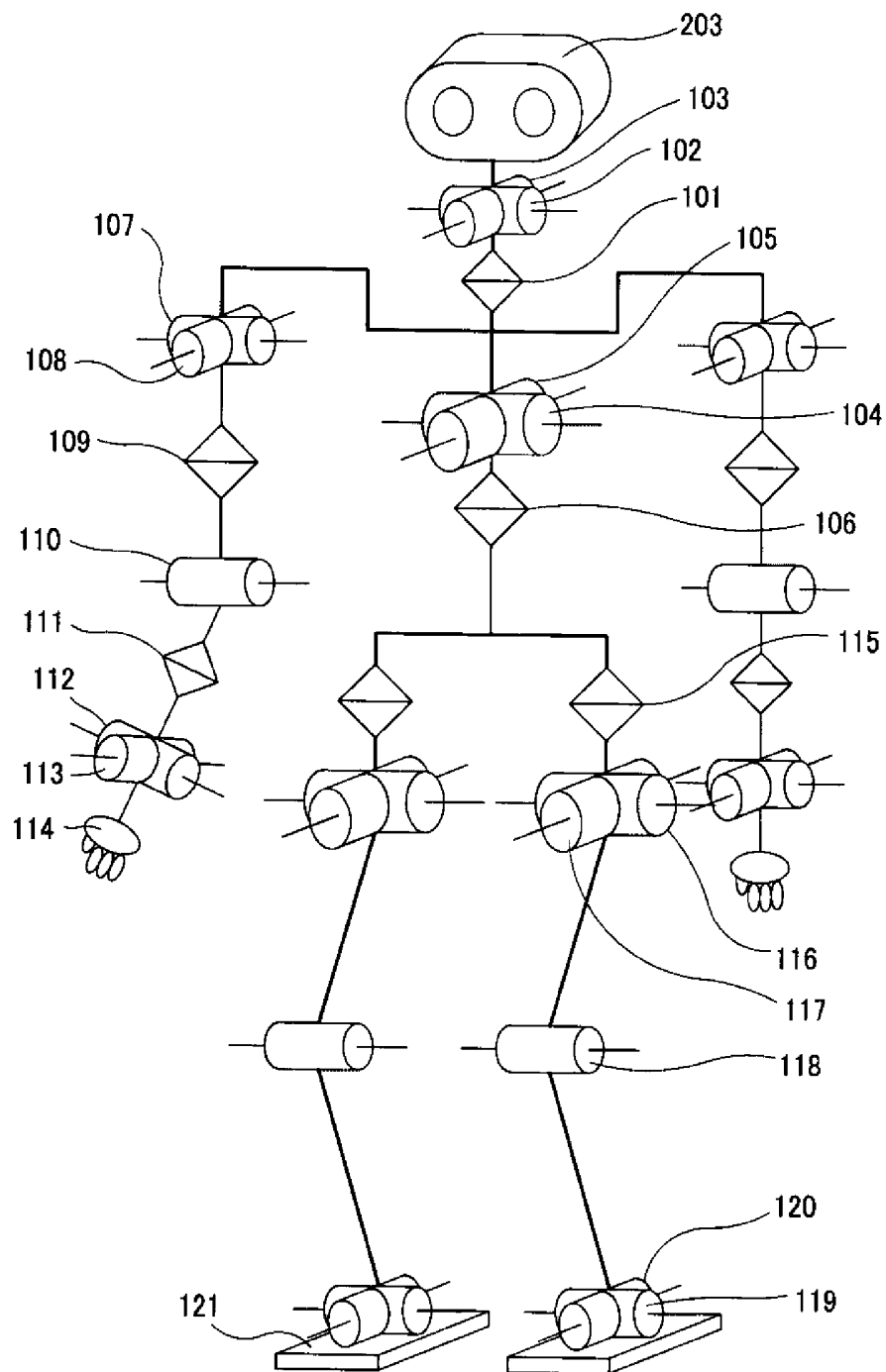


FIG.5

[図6]

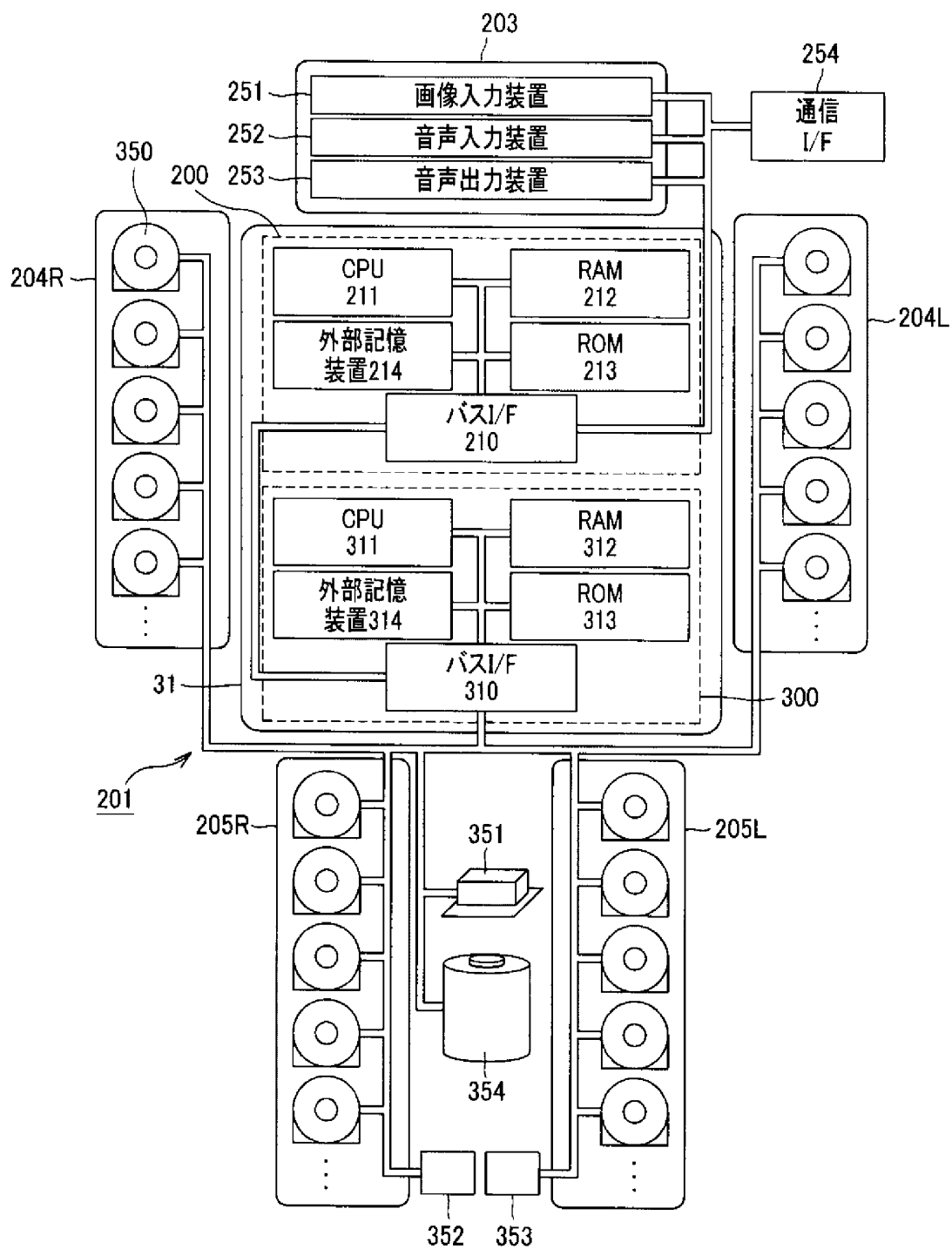


FIG. 6

[図7]

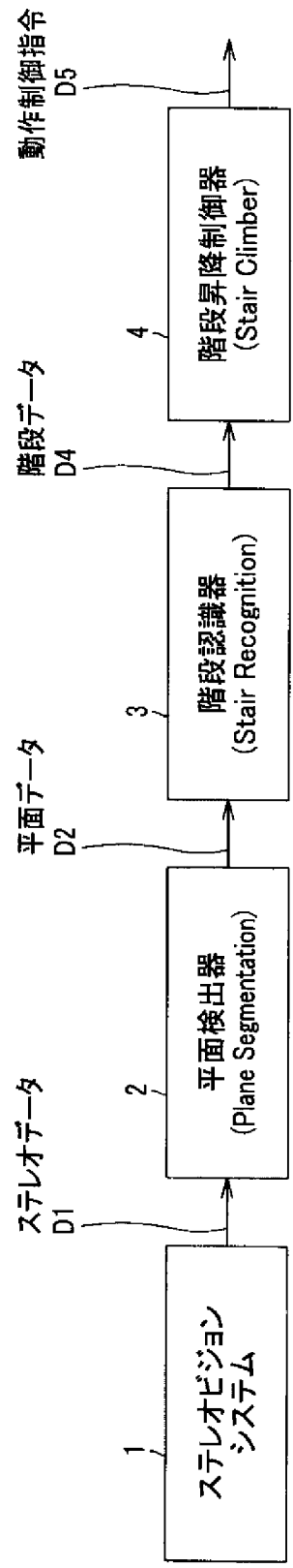


FIG.7

[図8]

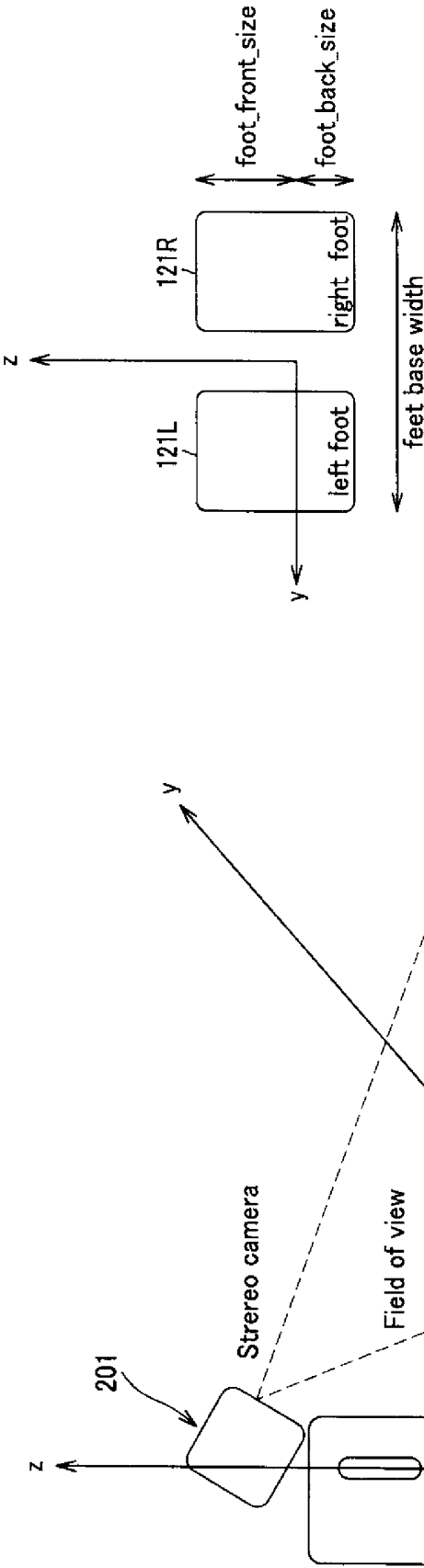


FIG.8B

FIG.8A

[図9]

FIG.9A

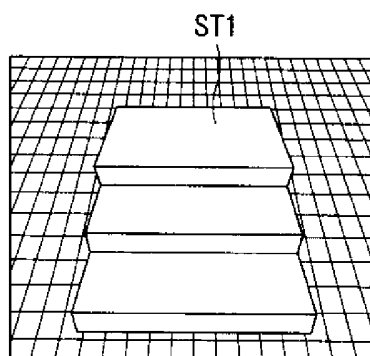


FIG.9B

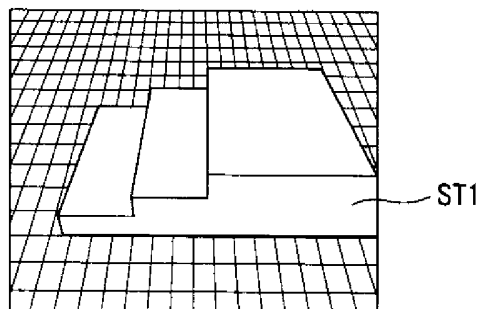
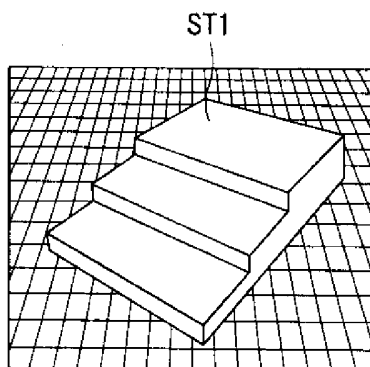


FIG.9C



4cm × 30cm × 10cm/21cm

[図10]

FIG.10A

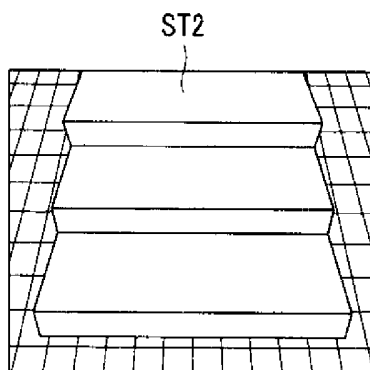


FIG.10B

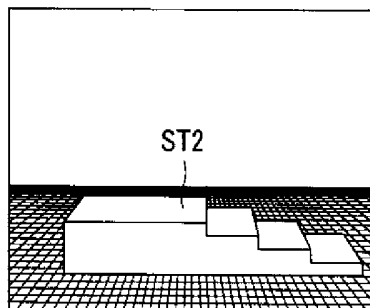
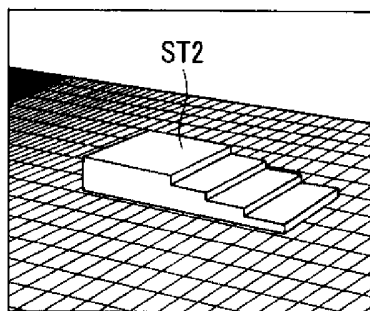


FIG.10C



3cm × 33cm × 12cm/32cm

[図11]

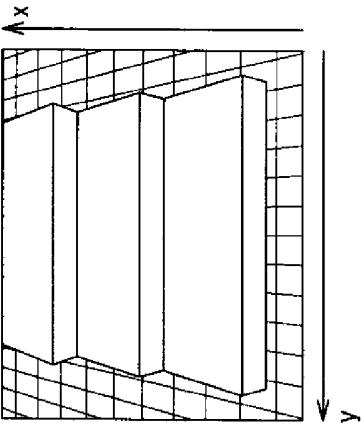


FIG. 11A

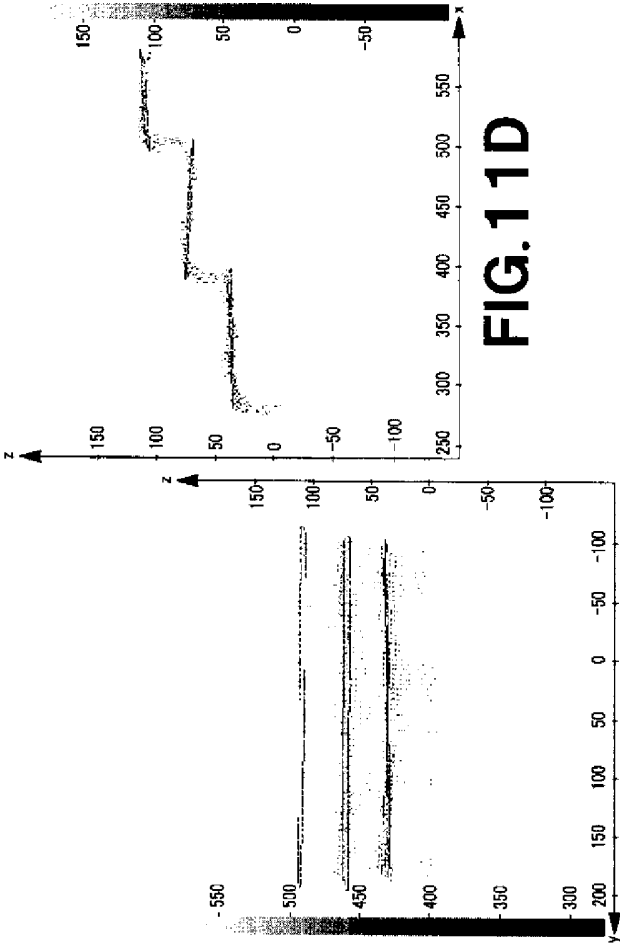


FIG. 11C

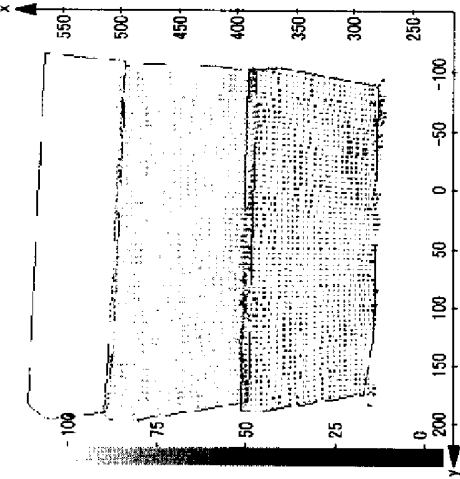


FIG. 11B

FIG. 11D

[図12]

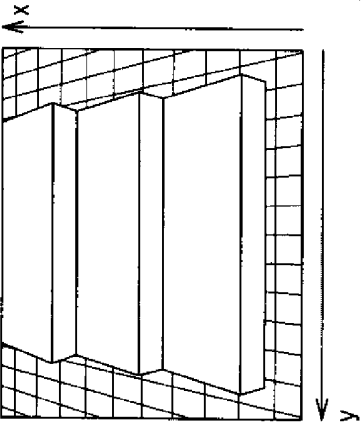


FIG. 12A

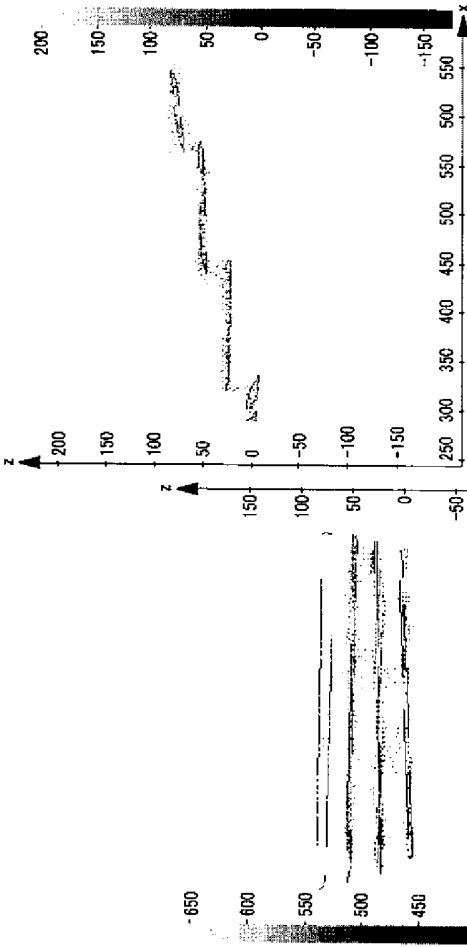


FIG. 12C

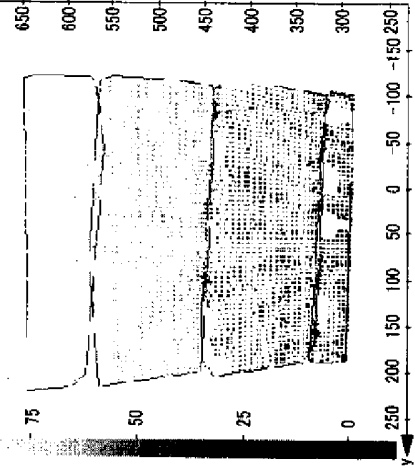


FIG. 12B

FIG. 12D

[図13]

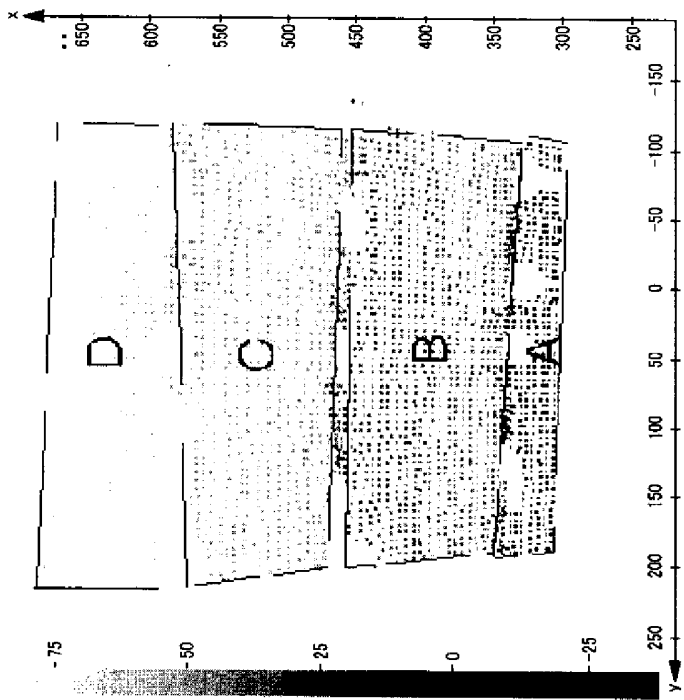


FIG.13B

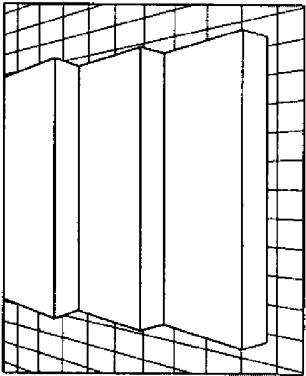


FIG.13A

[図14]

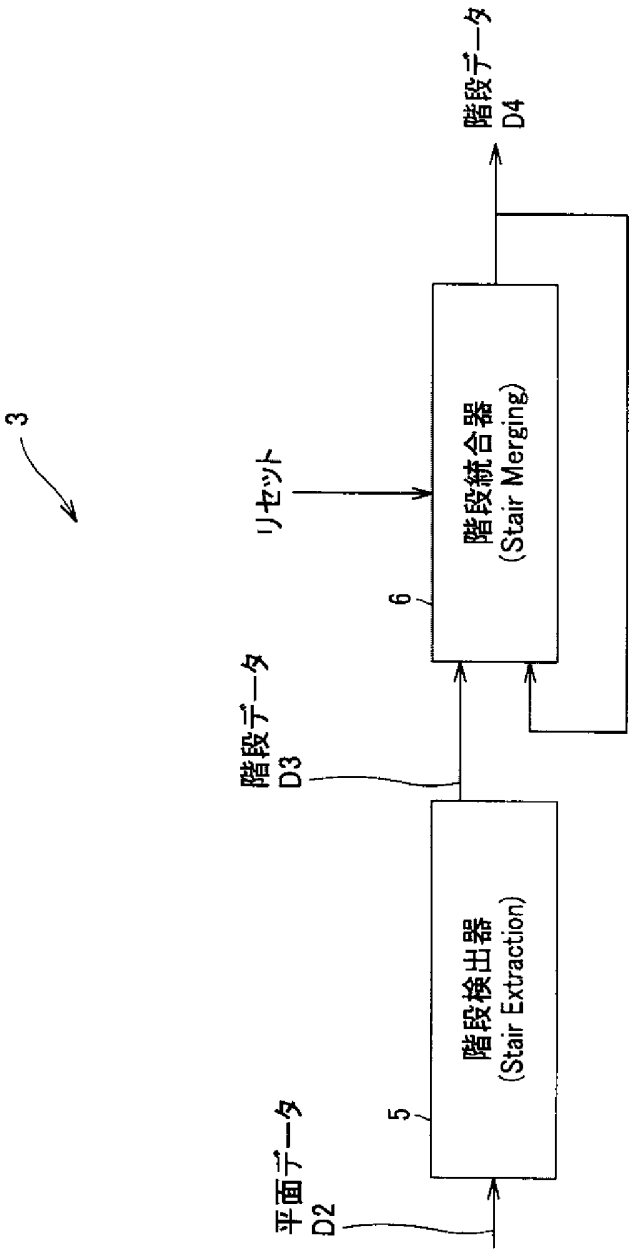


FIG.14

[図15]

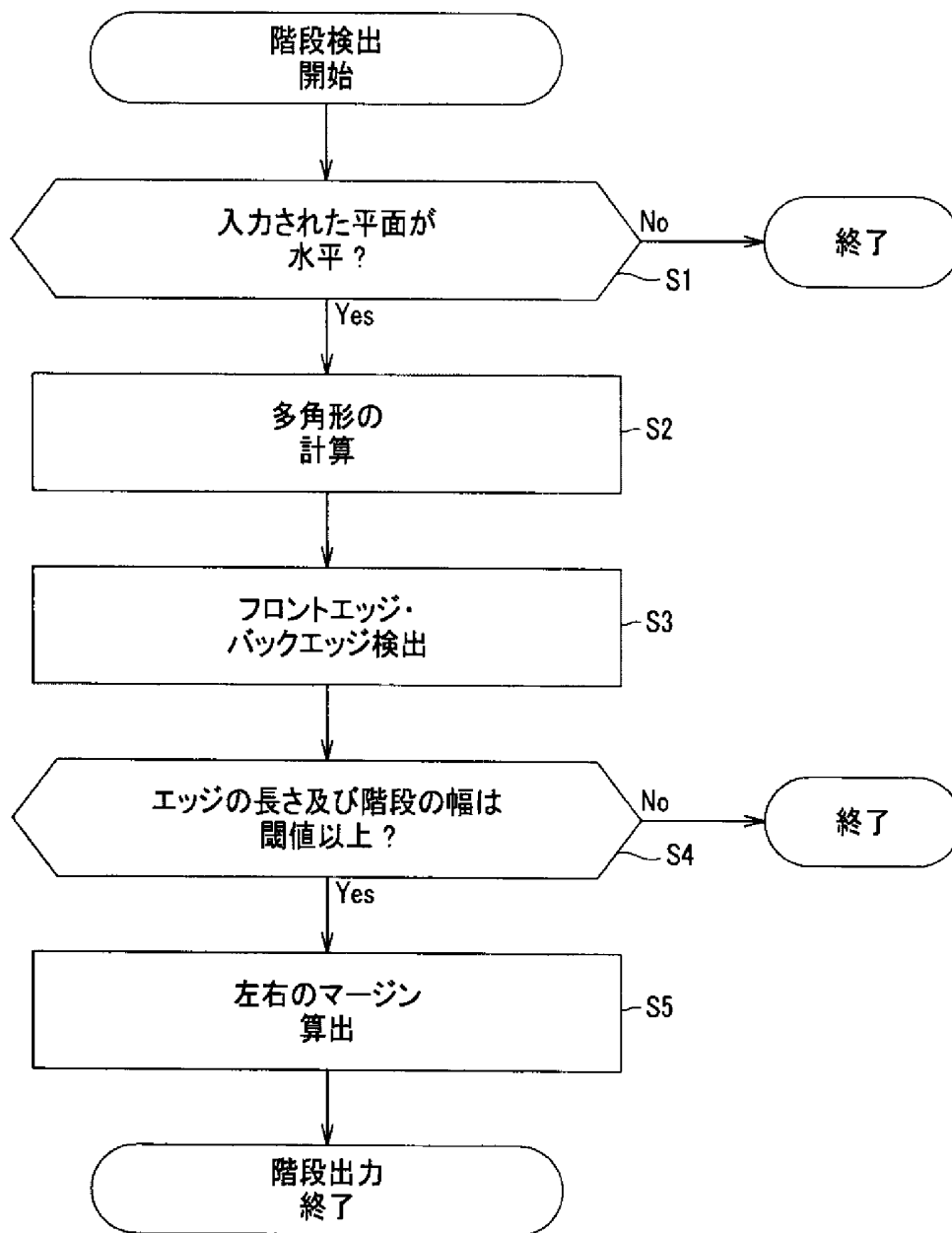


FIG.15

[図16]

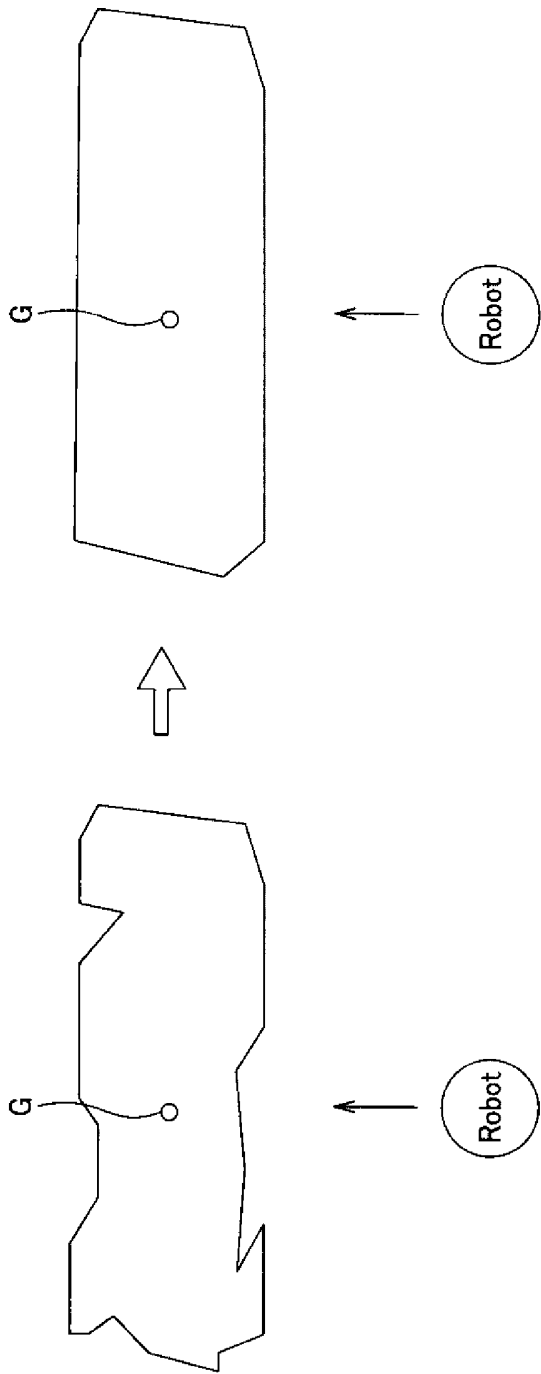
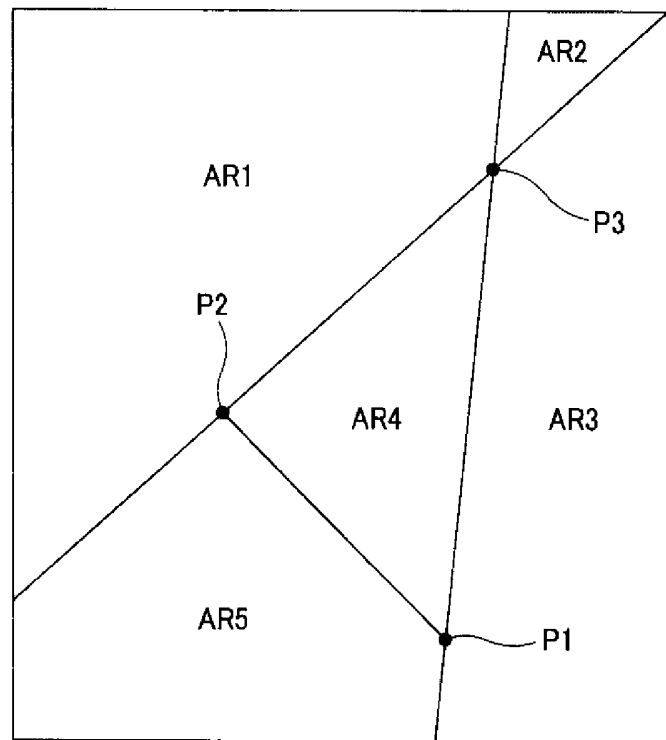


FIG.16B

FIG.16A

[図17]

**FIG.17**

[図18]

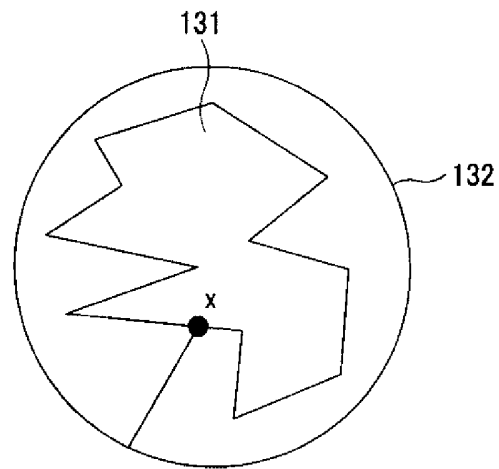


FIG.18A

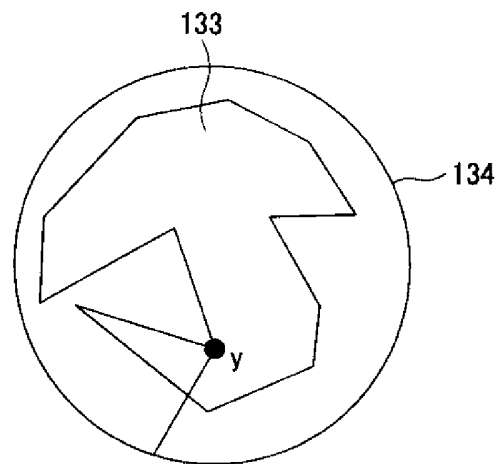


FIG.18B

[図19]

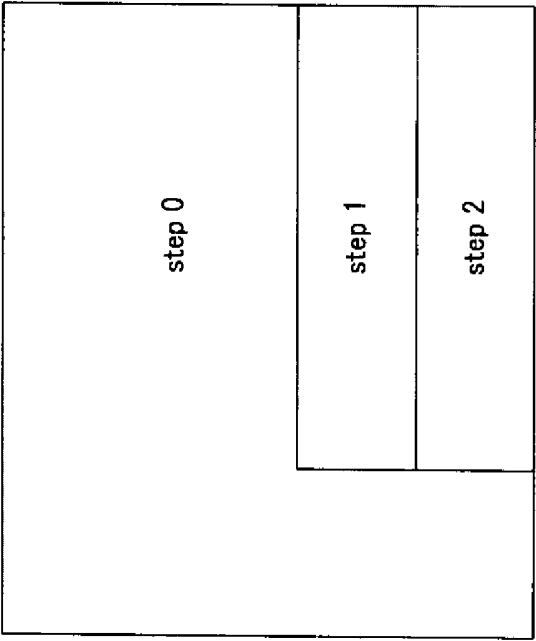
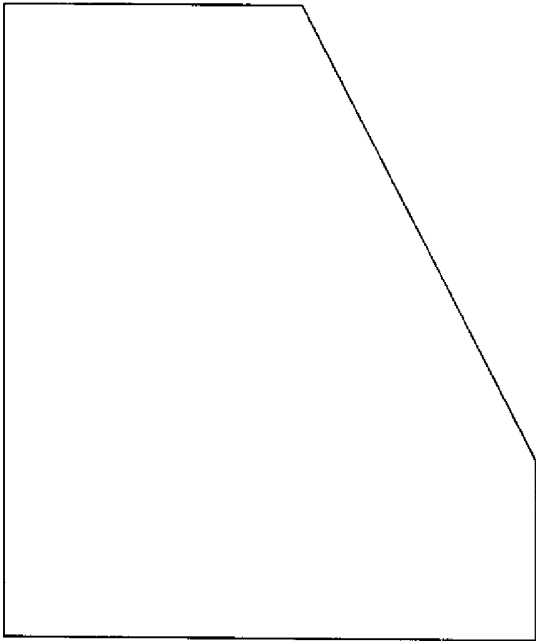
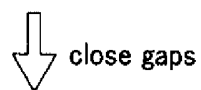
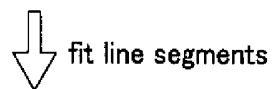
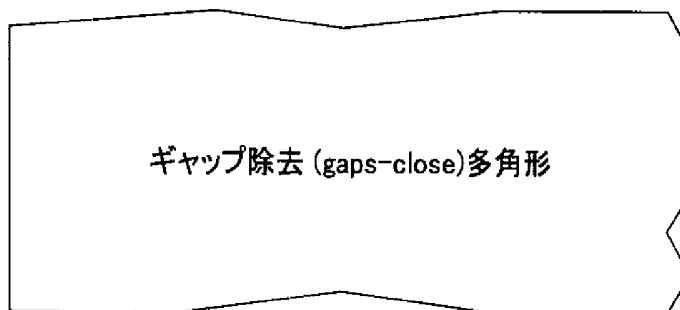
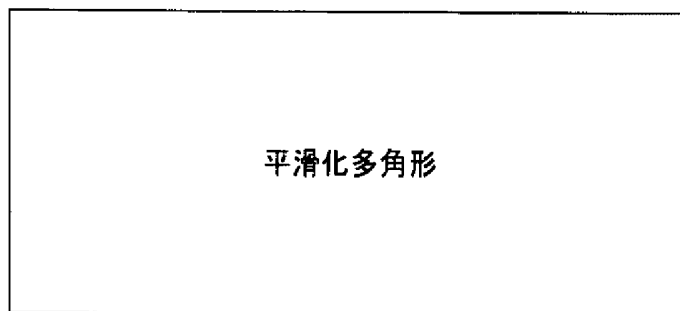


FIG.19B

FIG.19A

[図20]

FIG.20A**FIG.20B****FIG.20C**

[図21]

Close gaps :

```
for each triple of consecutive points do
  if distance first to last point  $< d_{\text{gap}}$  then
    remove middle point
  end
until no more changes
```

Fit line segments :

```
for each triple of consecutive points do
  compute least-squares line fit with error
end
for each fitted line, sorted by error do
  if error  $< e_{\text{fit}}$  then
    remove middle point
    update first and last point
    re-compute fitted lines for points
    preceding and following line
  end
until no more changes
```

FIG.21

[図23]

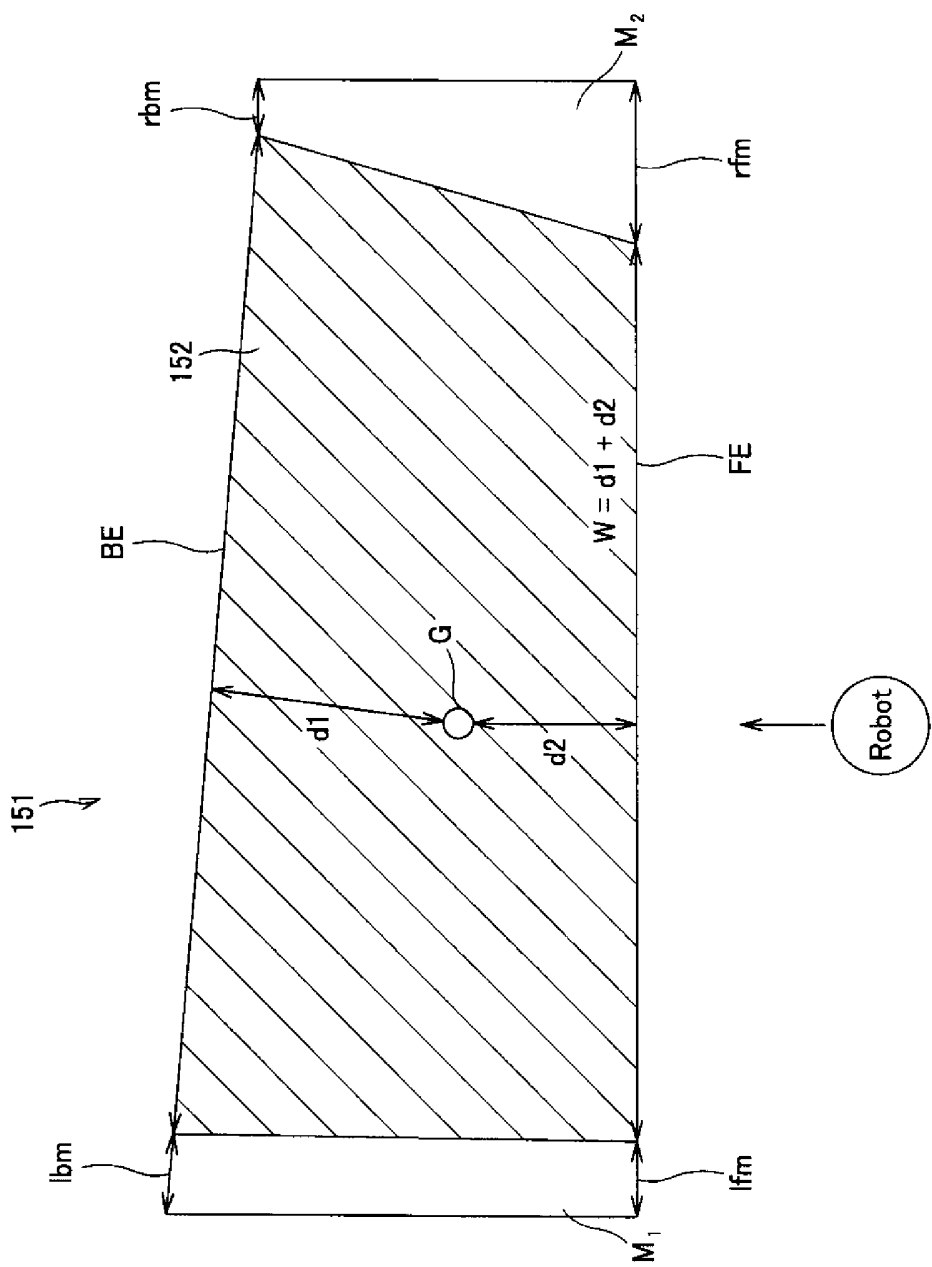


FIG.23

[図24]

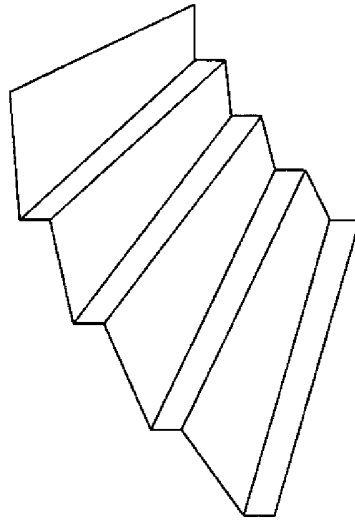


FIG.24B

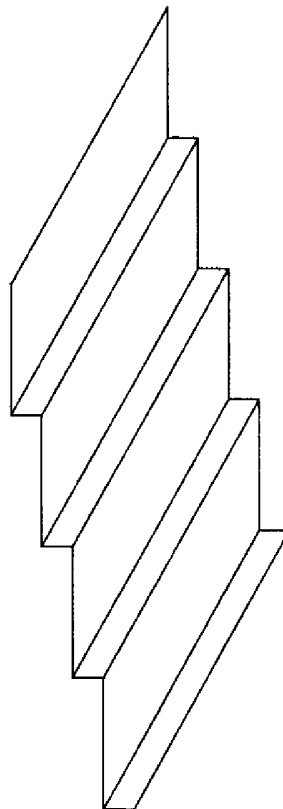


FIG.24A

[図25]

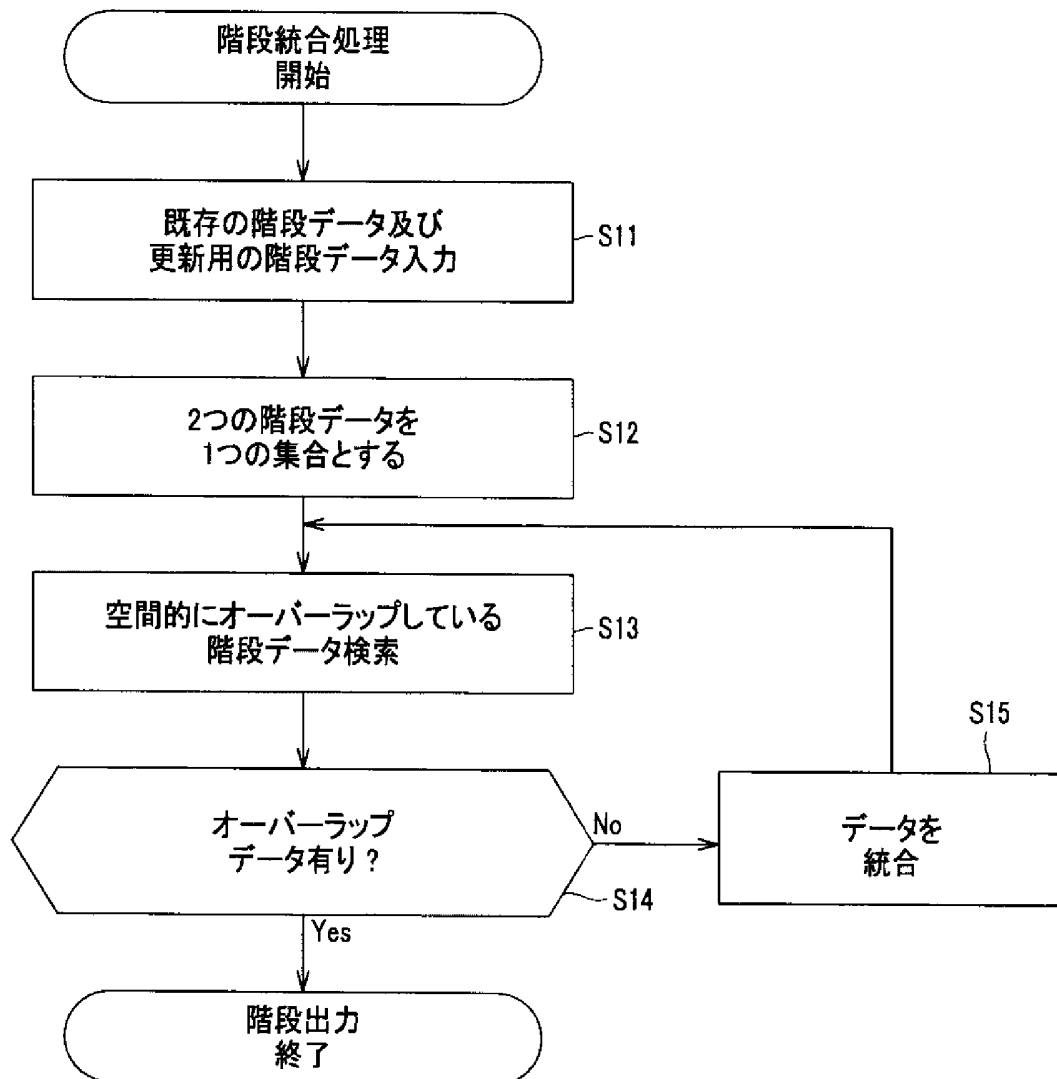


FIG.25

[図26]

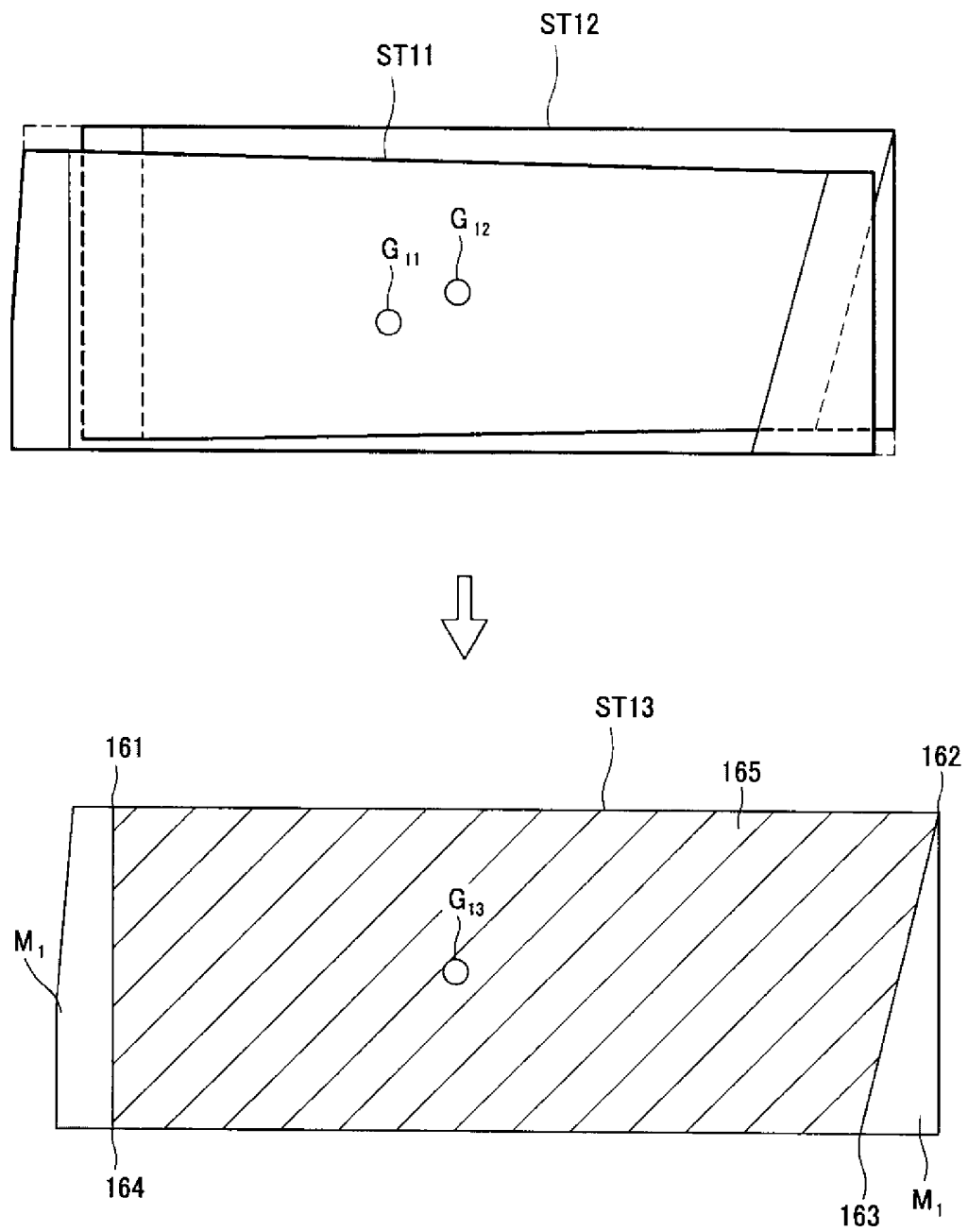


FIG.26

[図27]

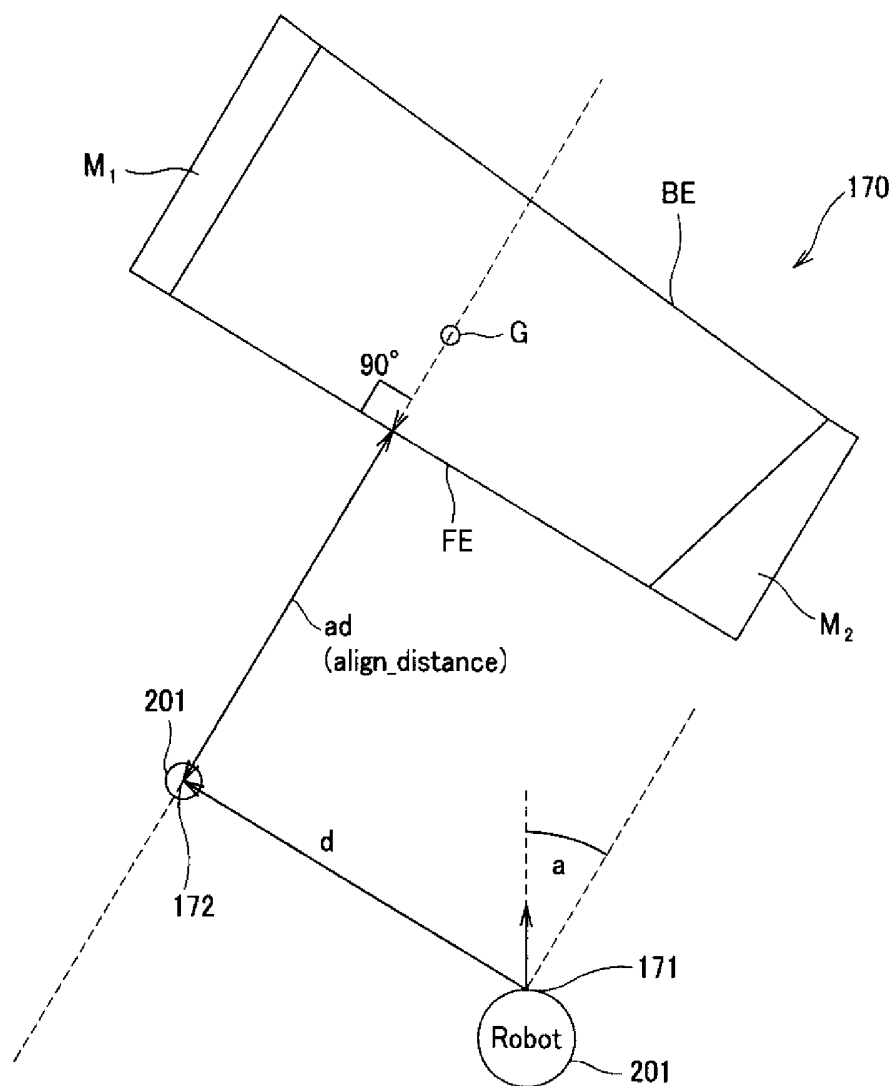


FIG.27

[図28]

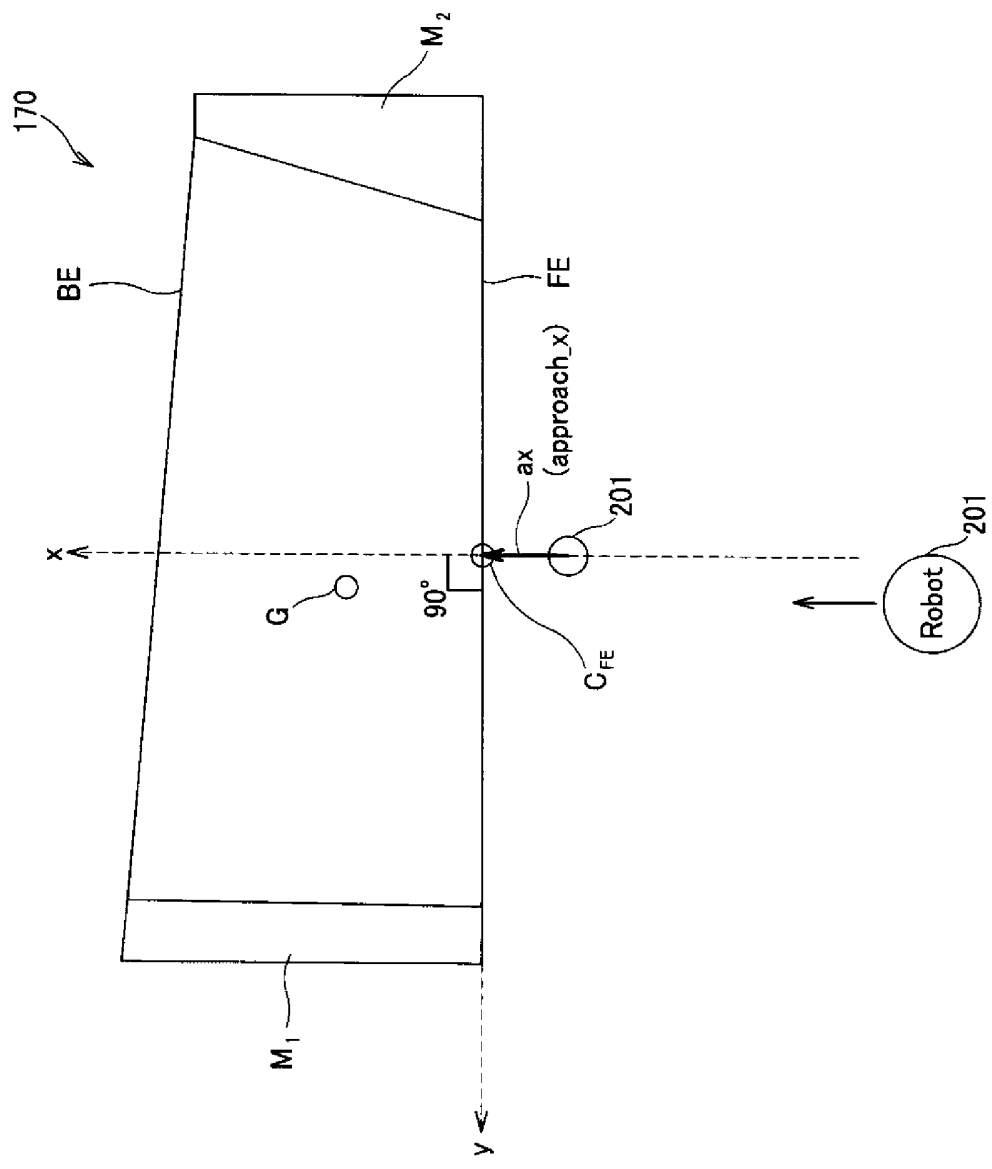


FIG.28

[図29]

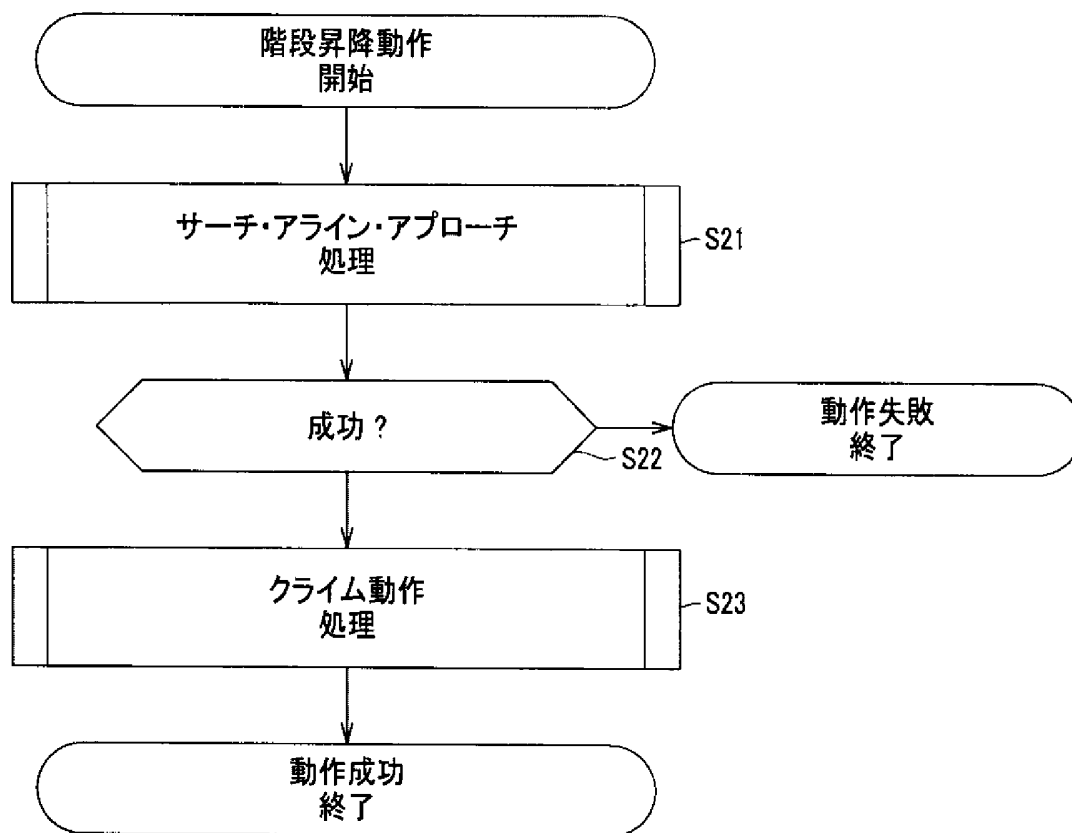


FIG.29

[図30]

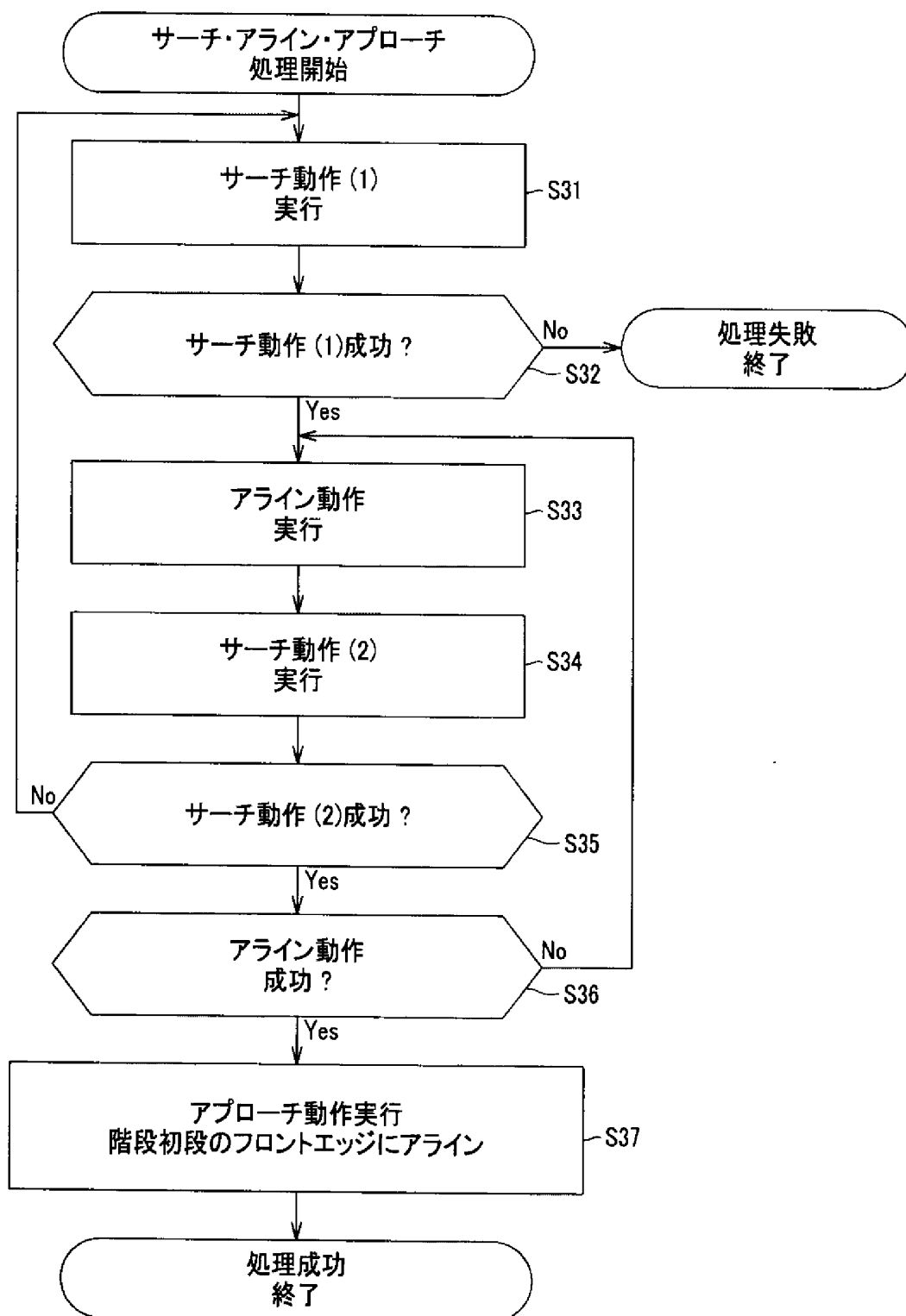


FIG.30

[図31]

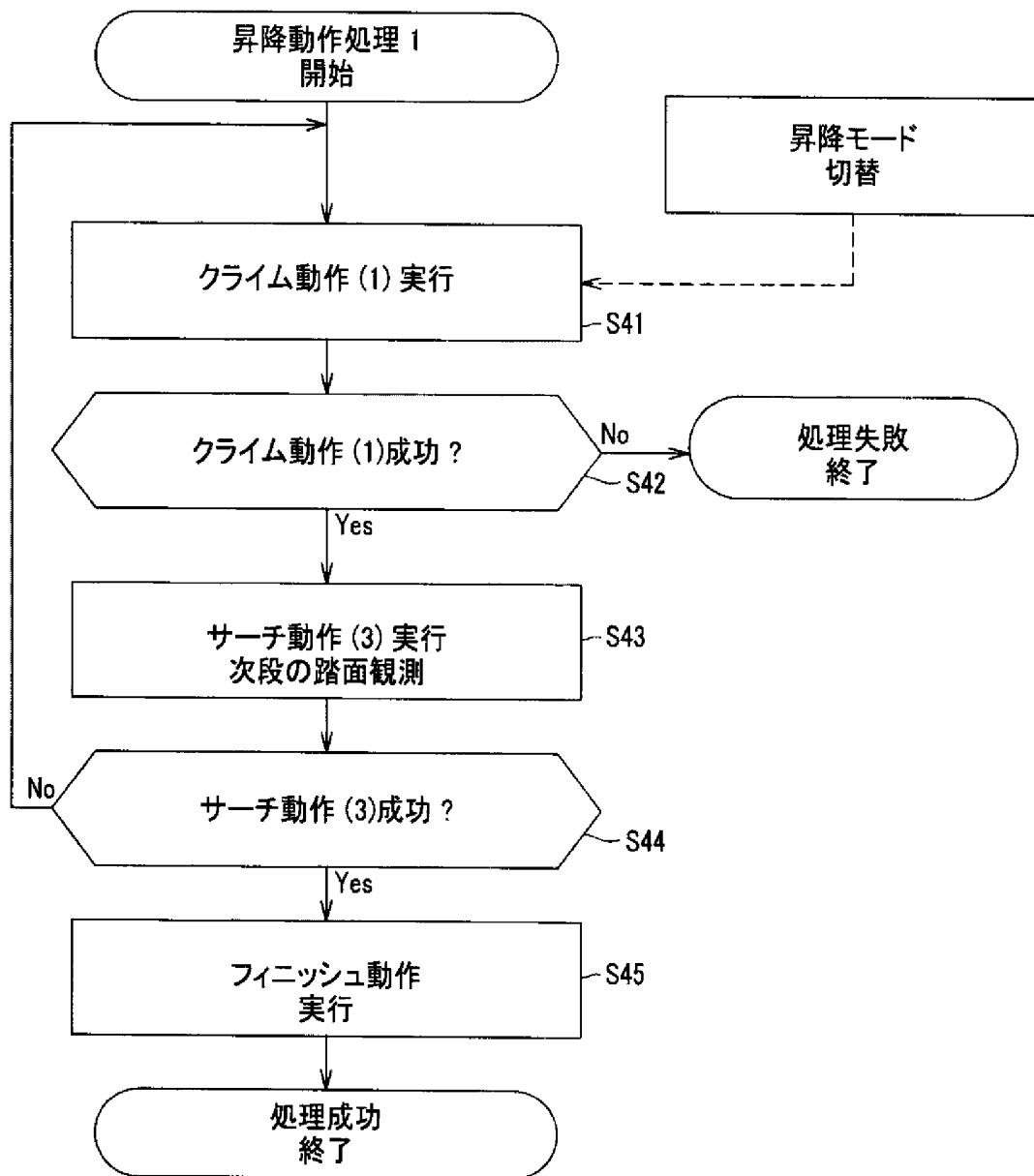


FIG.31

[図32]

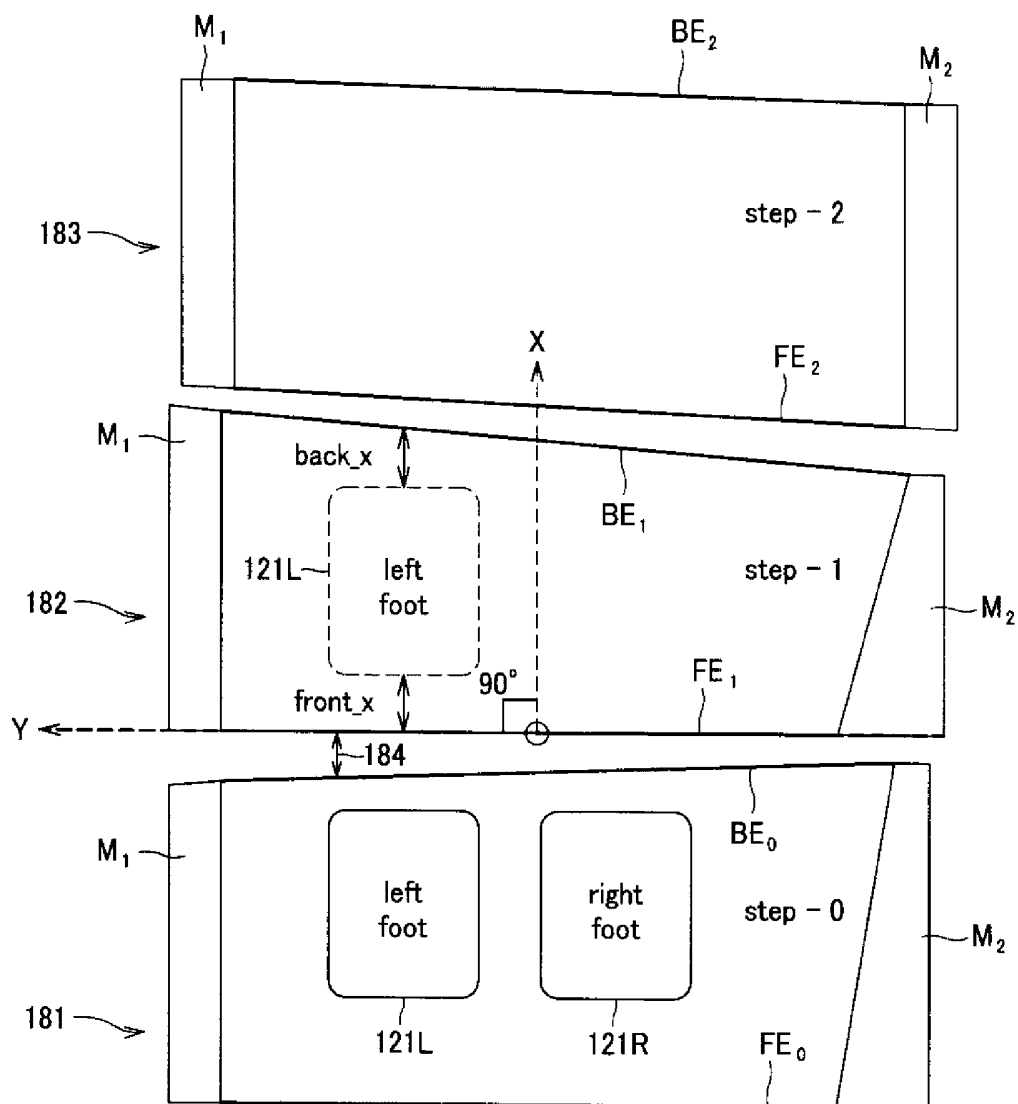


FIG.32

[図33]

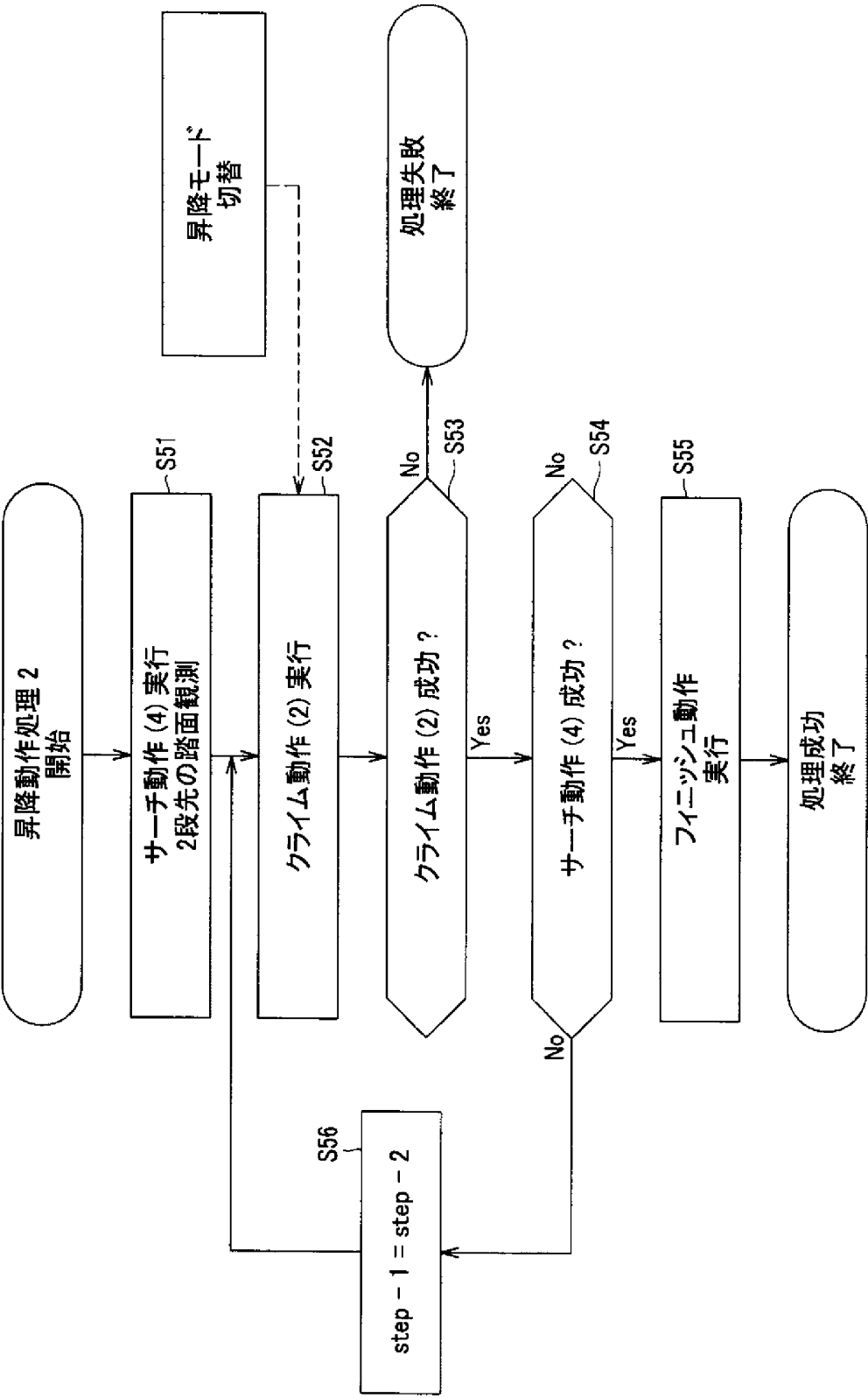


FIG.33

[図34]

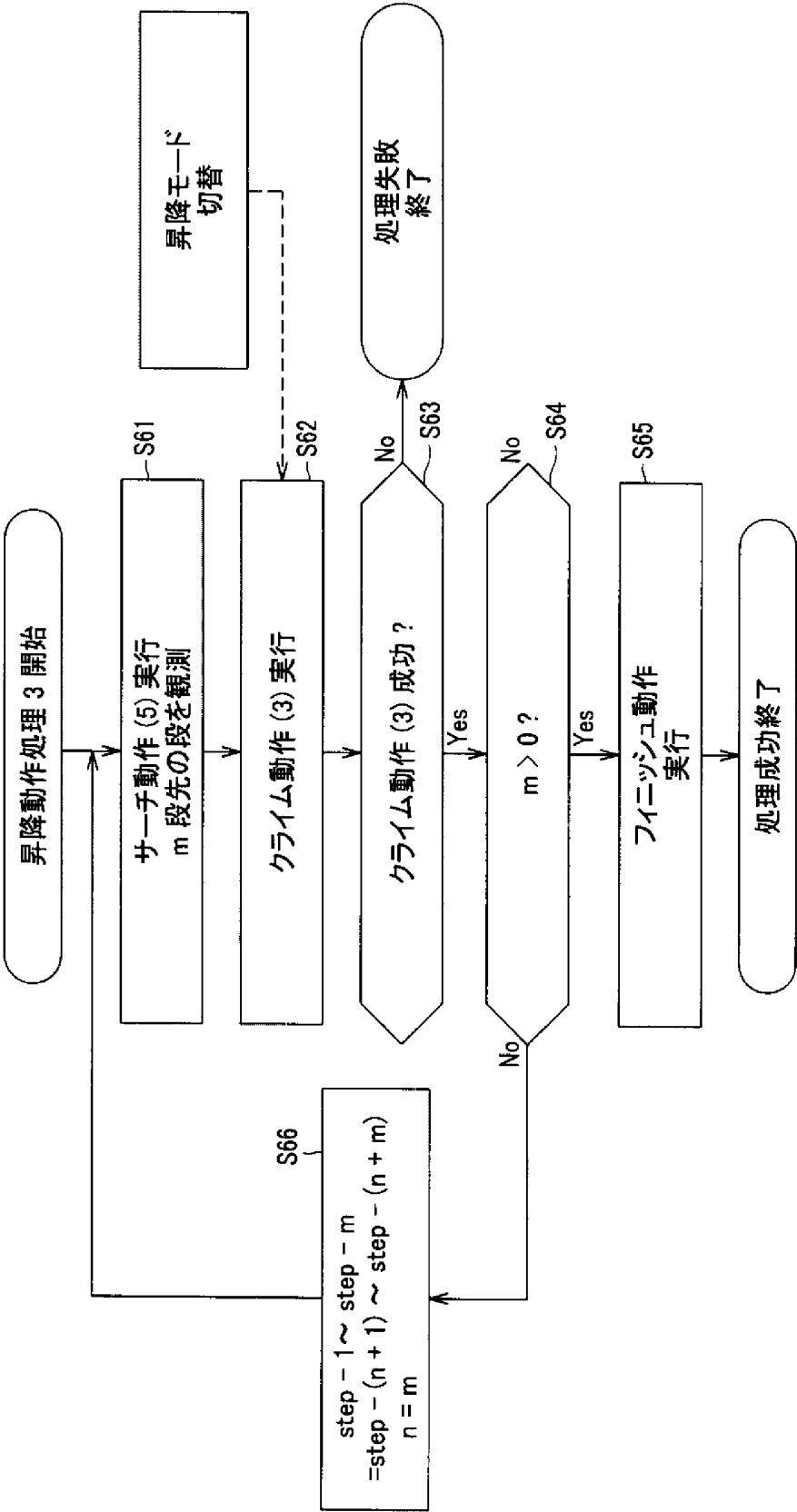


FIG.34

[図35]

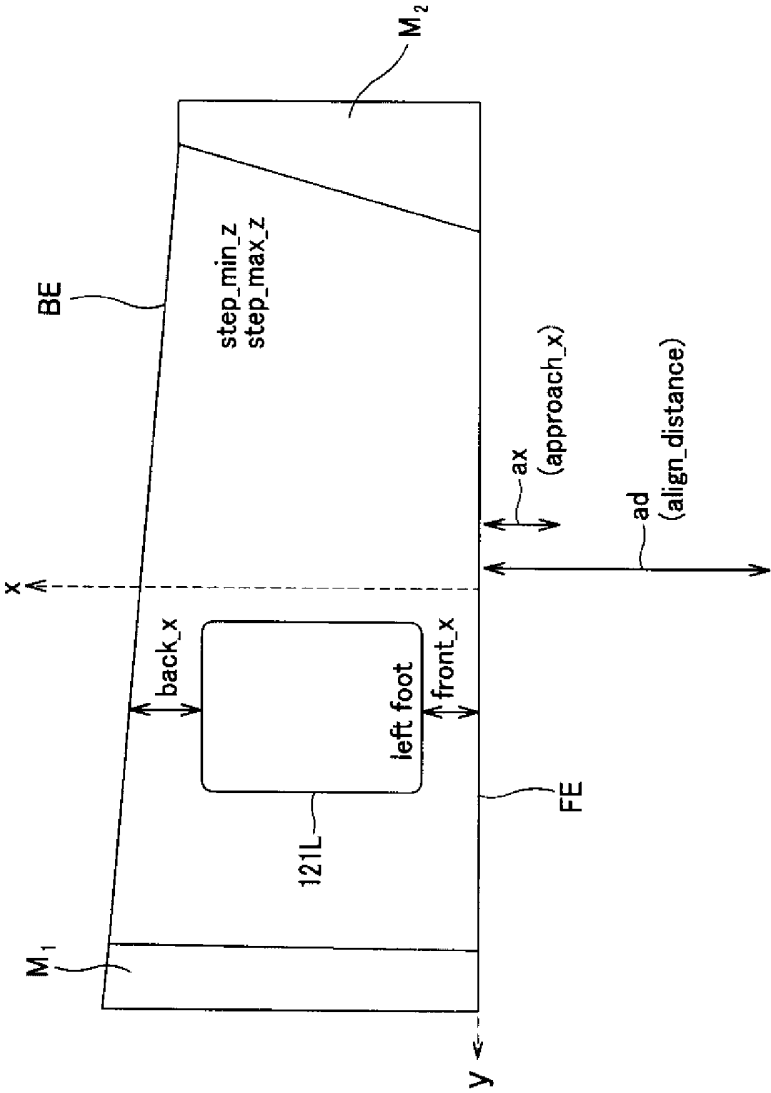


FIG.35A

	Ascend	Descend
step_min_z	15 mm	-50 mm
step_max_z	50 mm	-15 mm
align_distance	16 cm	5 cm
approach_x	-15 mm	5 mm
front_x_limit	-20 mm	5 mm
back_x_limit	-5 mm	10 mm
back_x_desired	-15 mm	5 mm

FIG.35B

[図36]

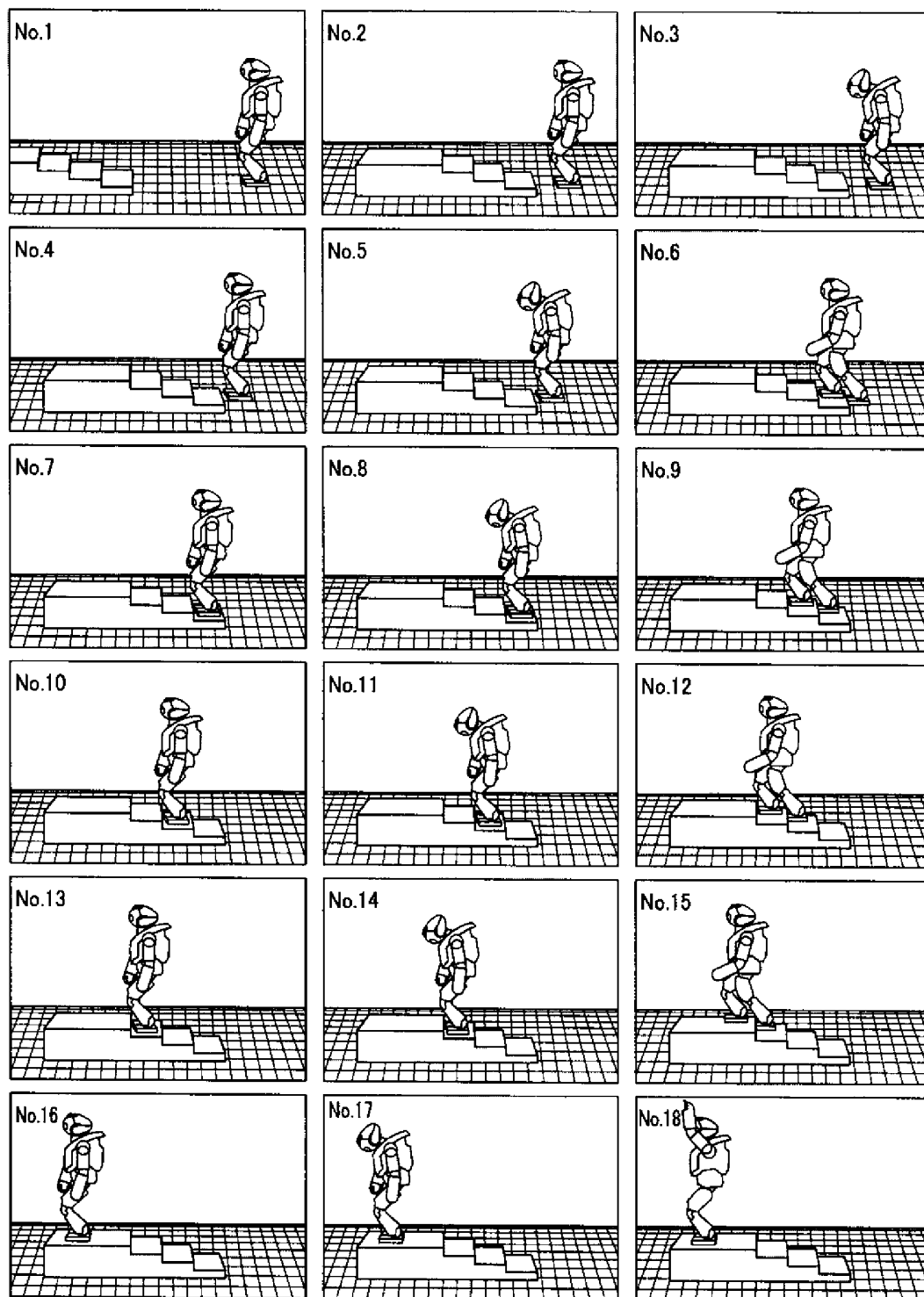
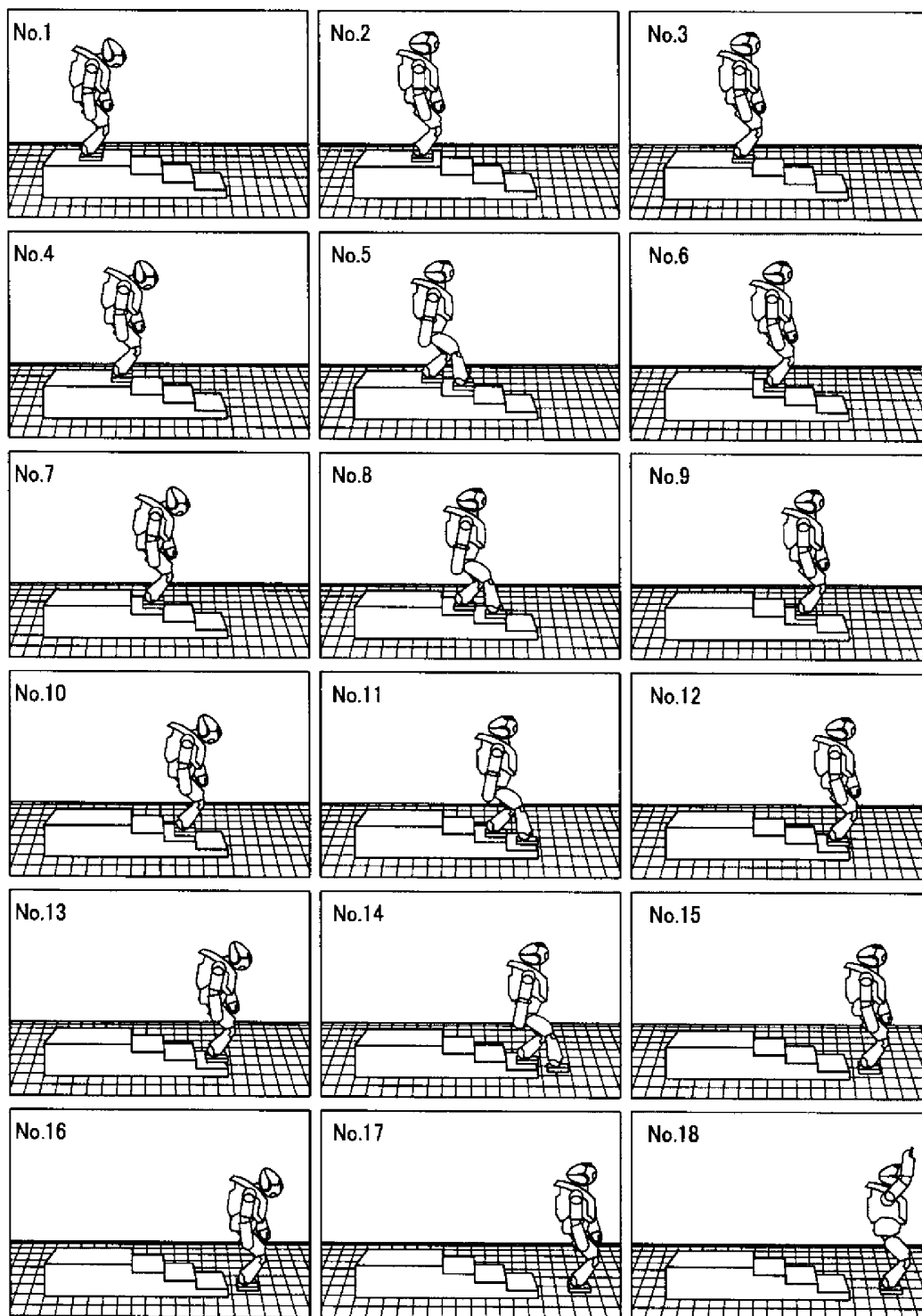


FIG.36

[図37]

**FIG.37**

[図38]

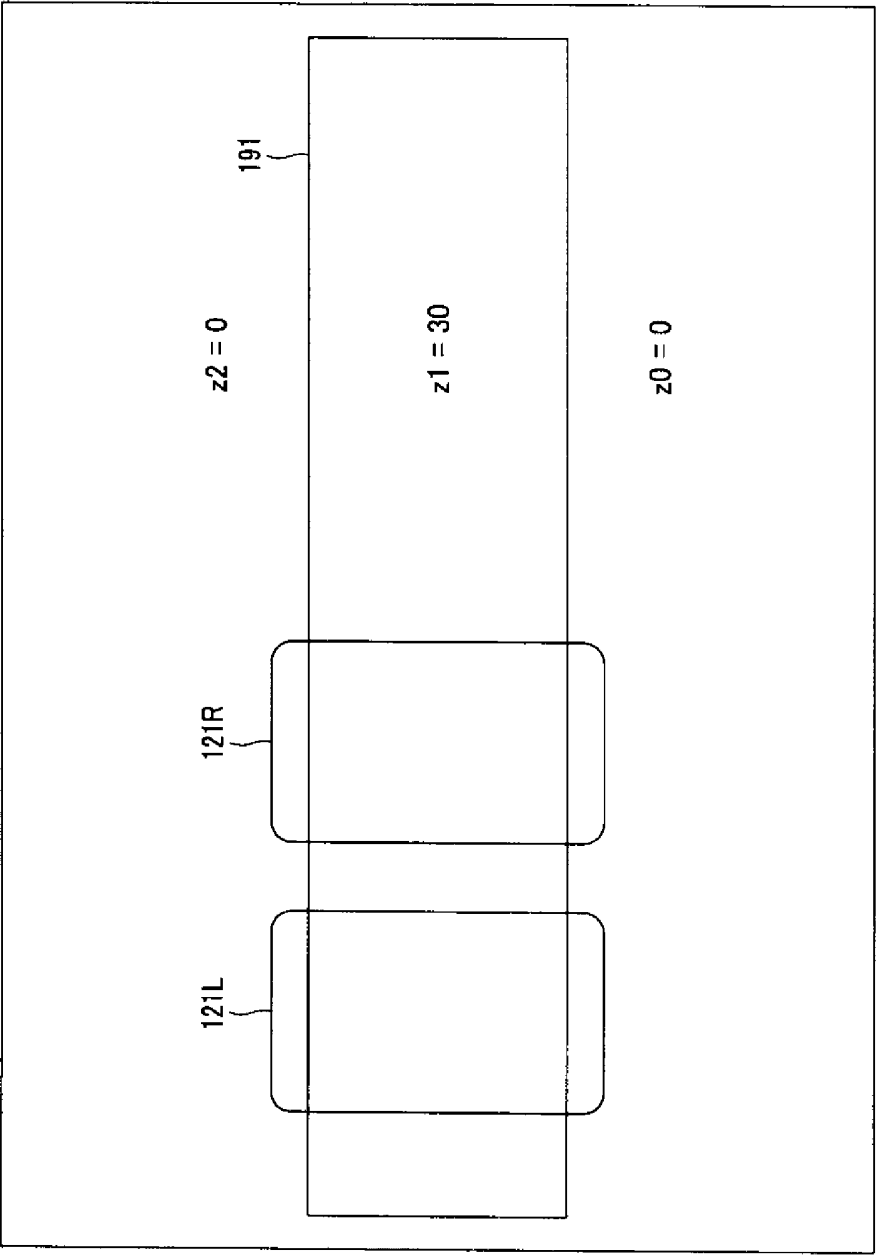


FIG.38

[図39]

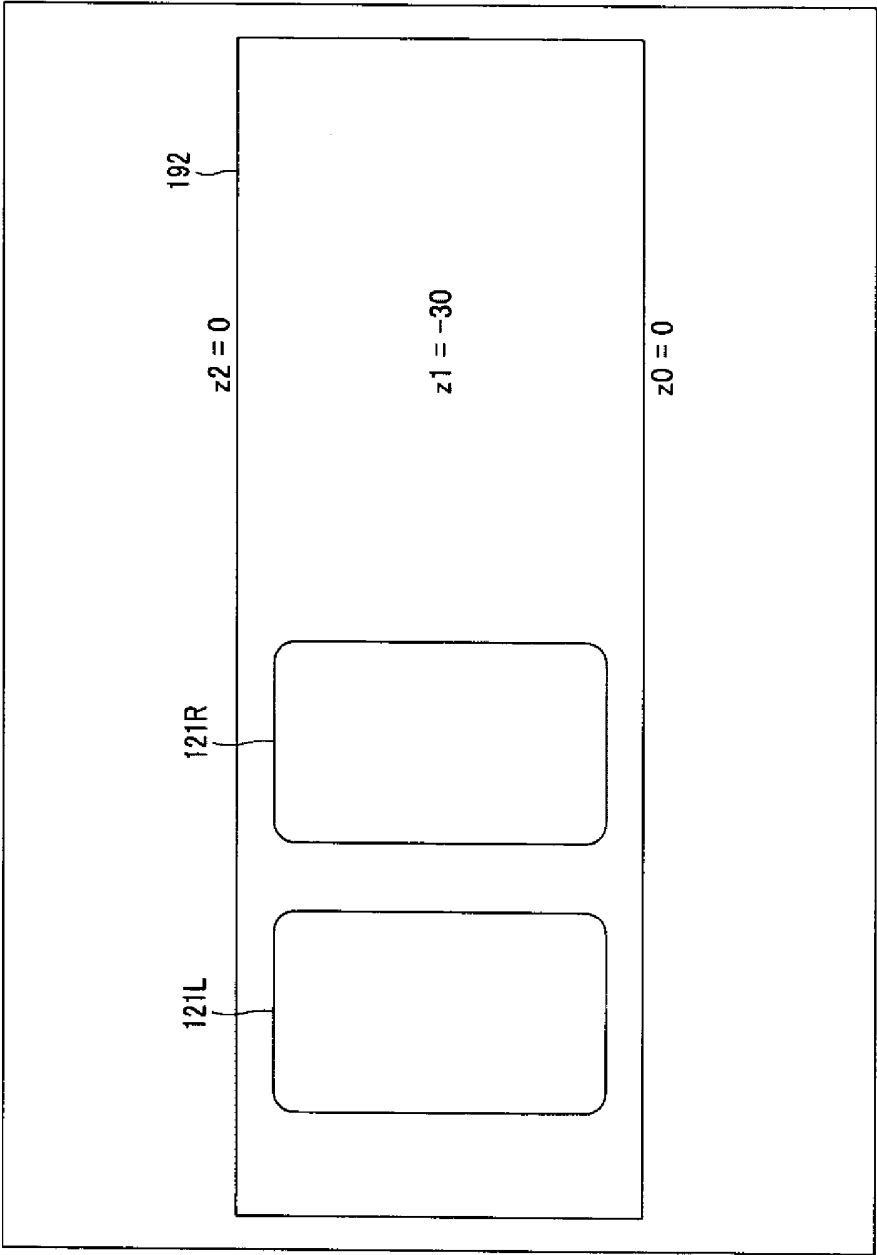


FIG.39

[図40]

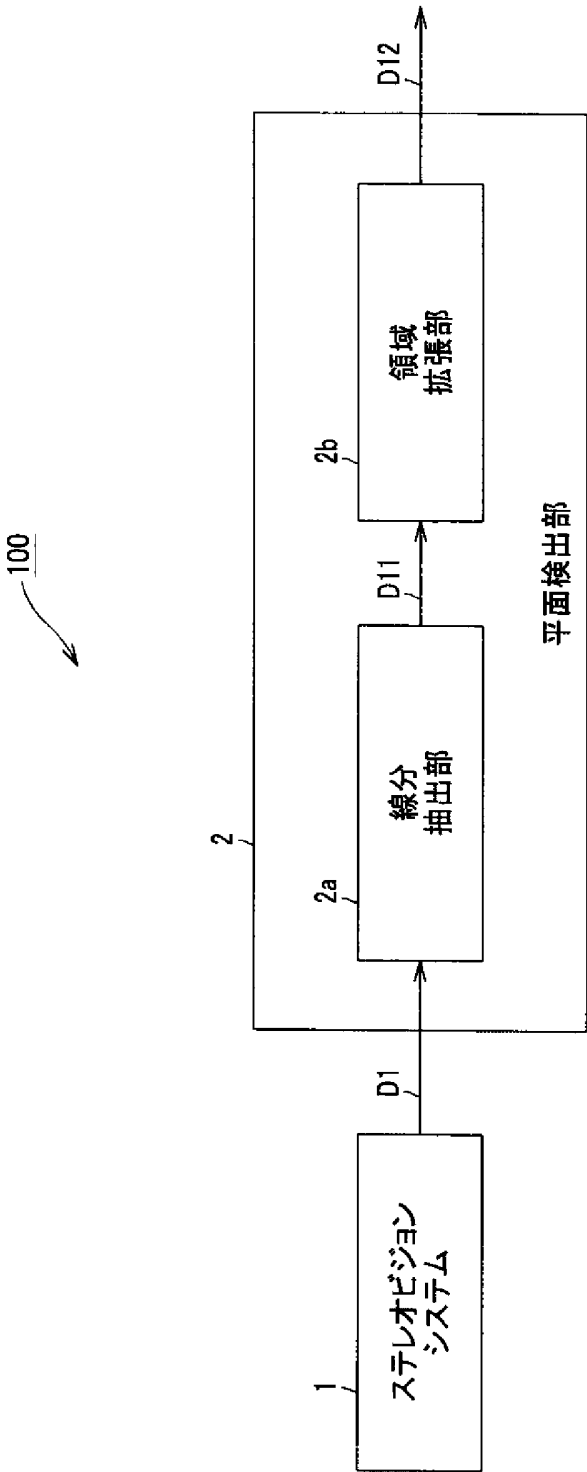


FIG.40

[図41]

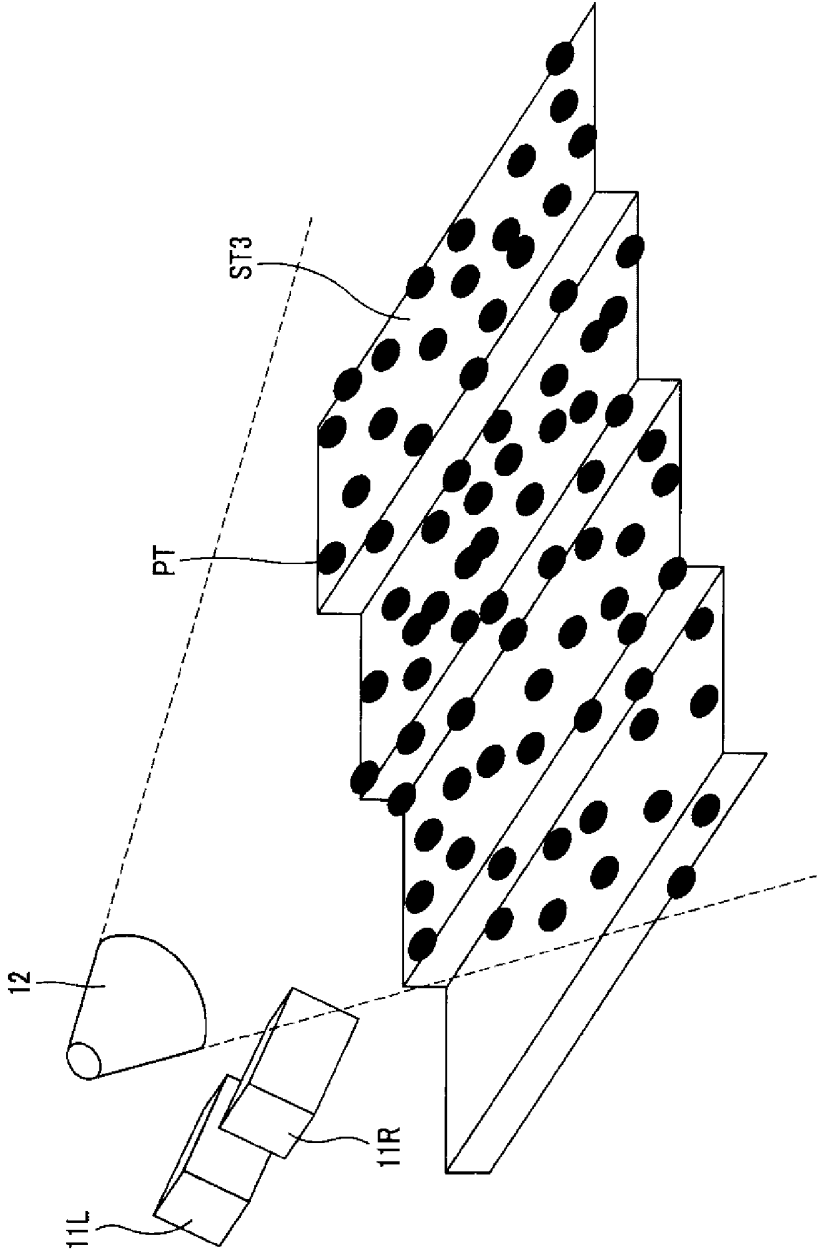


FIG.41

[図42]

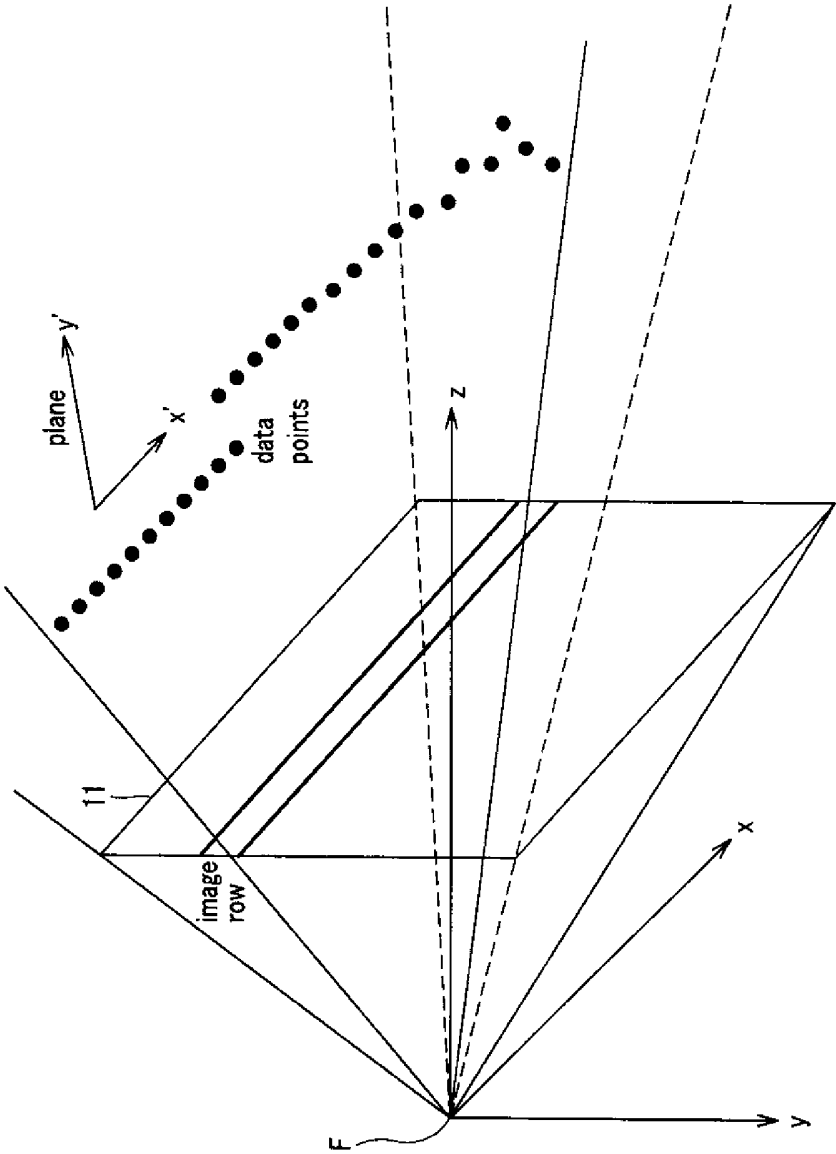


FIG.42

[図43]

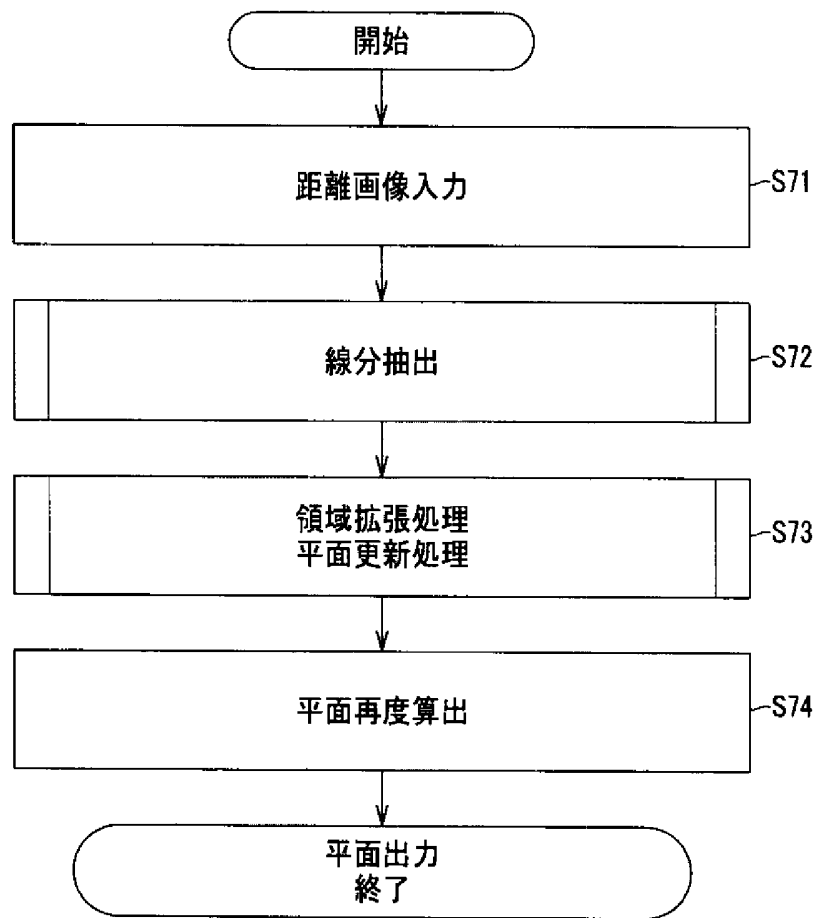


FIG. 43

[図44]

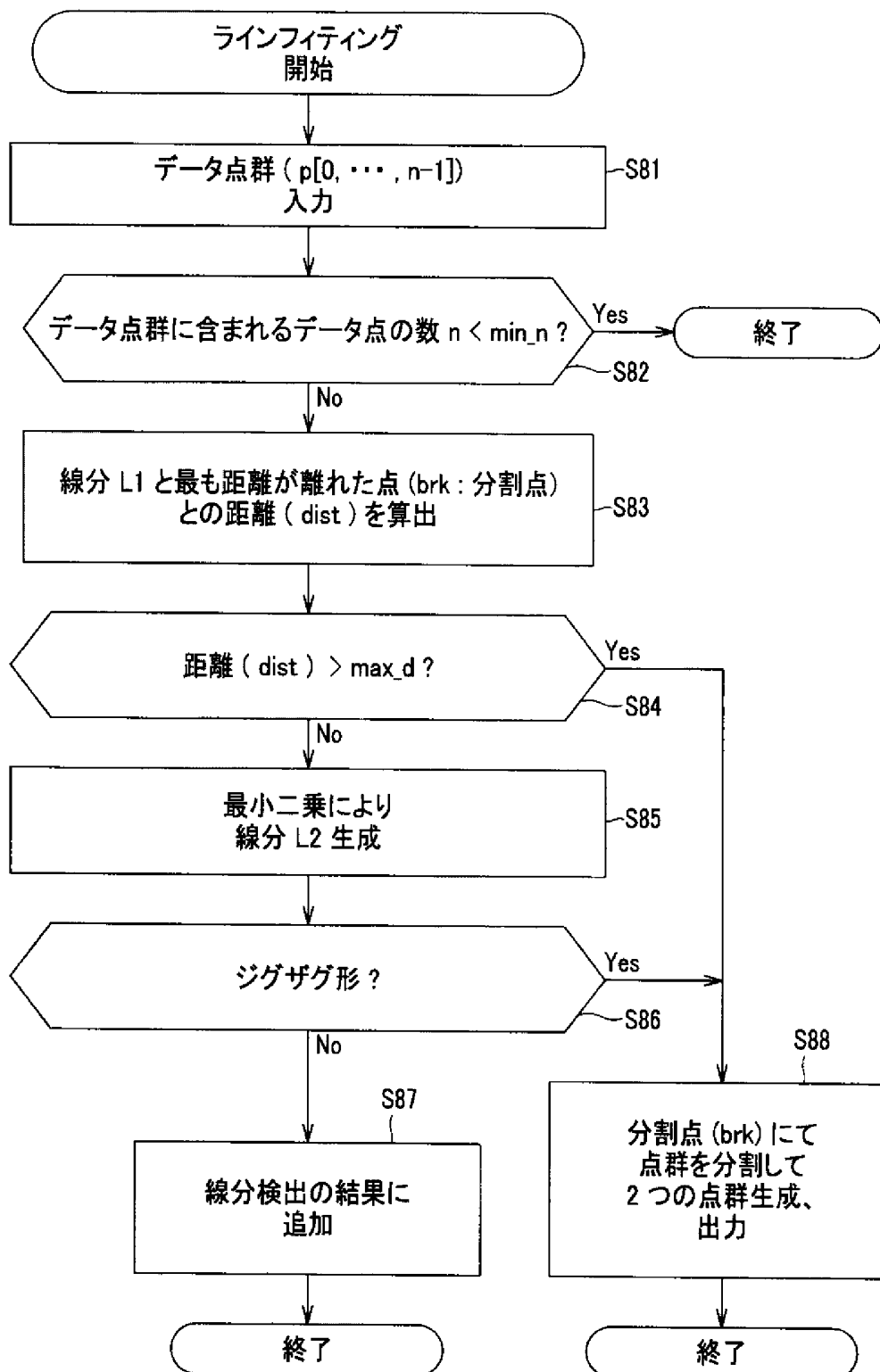


FIG.44

[図45]

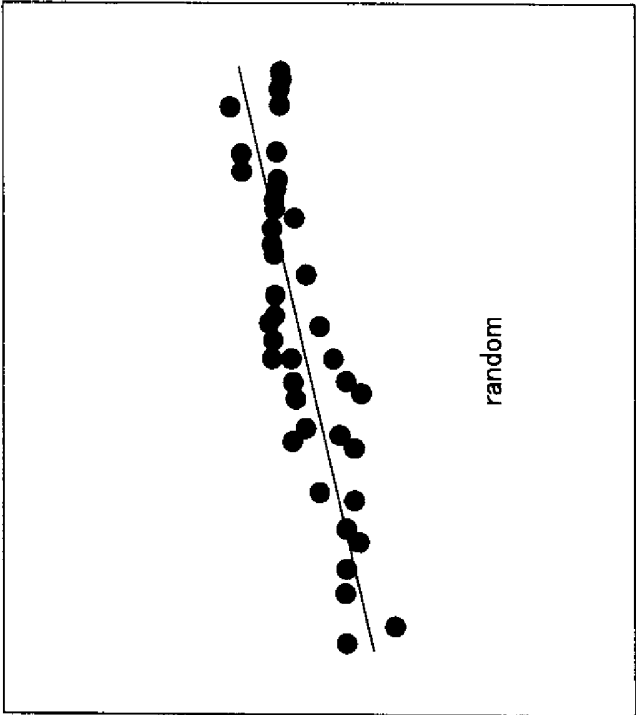


FIG.45B

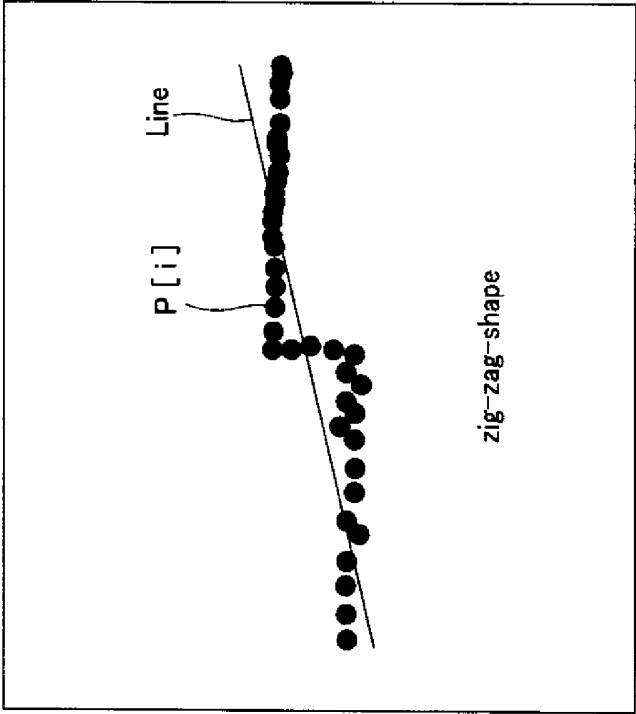


FIG.45A

[図46]

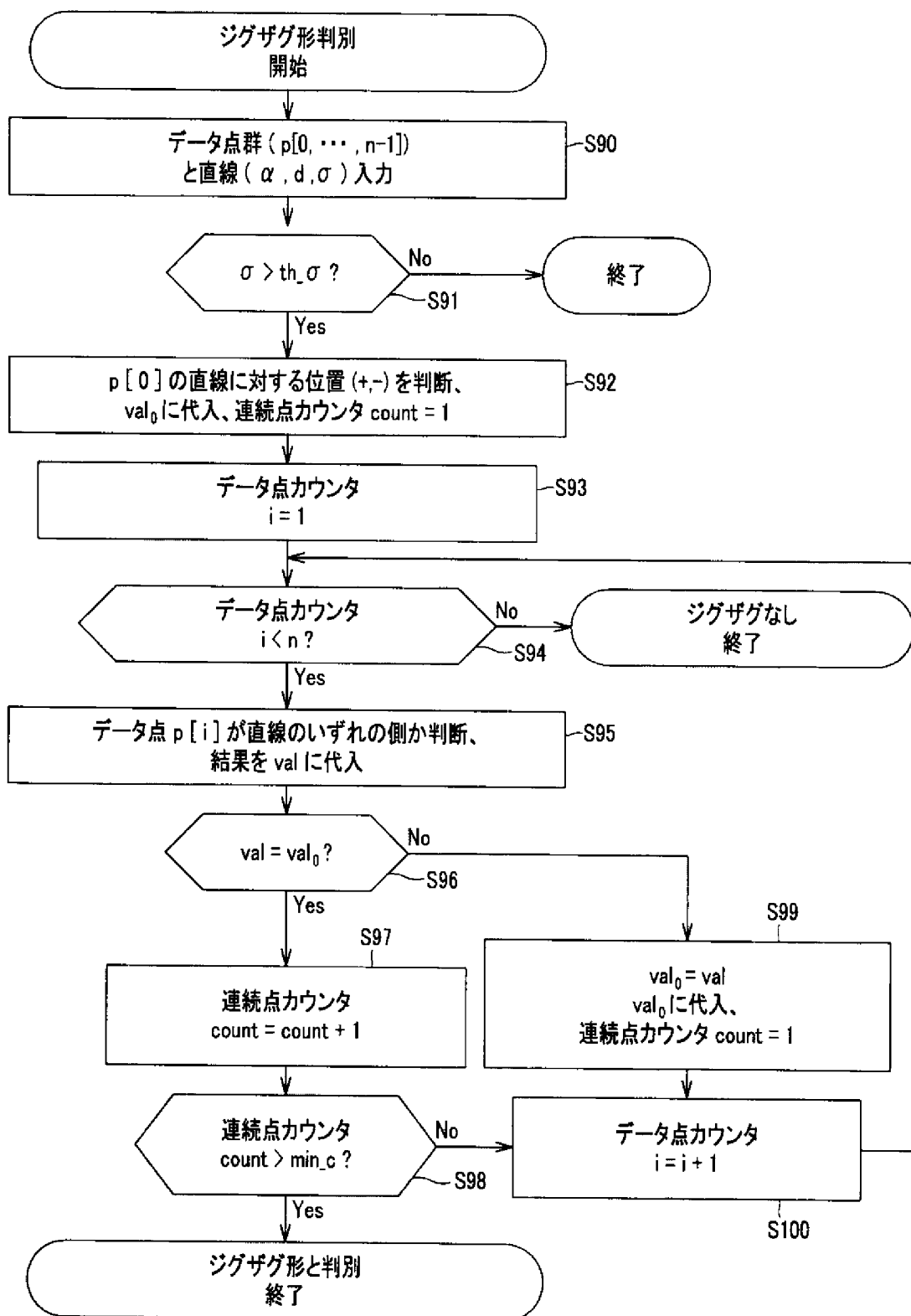


FIG.46

[図47]

Input: pts : vector of points, n : number of points, α, d, σ : parameters and std-dev of fitted line.

Output : true if curve contains a *zig-zag*- shape, false otherwise.

Sequence :

```

    if  $\sigma > thresh\sigma$  then
         $val0 = pts[0].x * \cos \alpha + pts[0].y * \sin \alpha + d$ 
         $count = 1$ 
        for  $i = 1$  to  $(n-1)$  do
             $val = pts[i].x * \cos \alpha + pts[i].y * \sin \alpha + d$ 
            if  $(val * val0 \leq 0)$  then
                 $val0 = val$ 
                 $count = 1$ 
            else
                 $count = count + 1$ 
                if  $(count \geq min-points-for-zig-zag-shape)$  then
                    return true
                endif
            endif
        endfor
    endif
    return false

```

FIG.47

[図48]

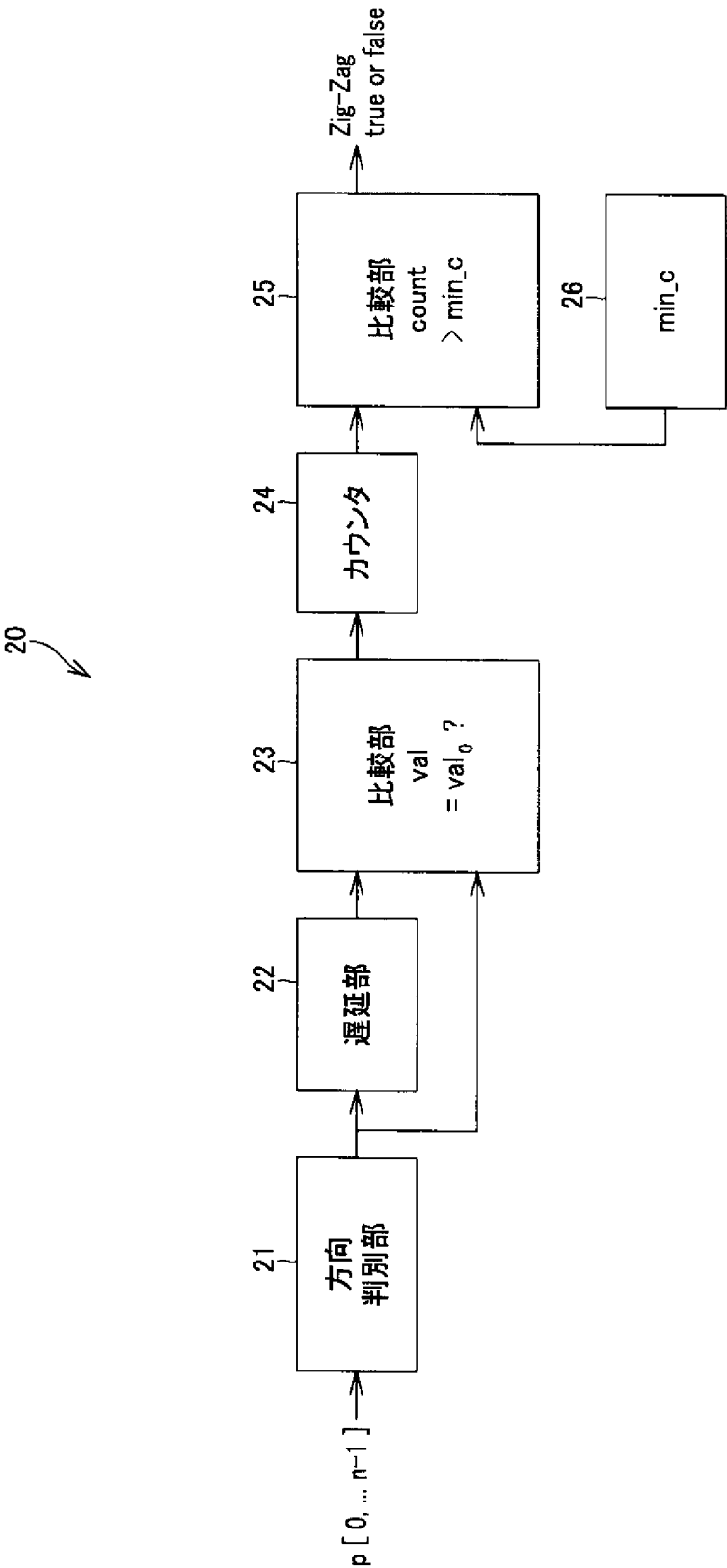


FIG.48

[図49]

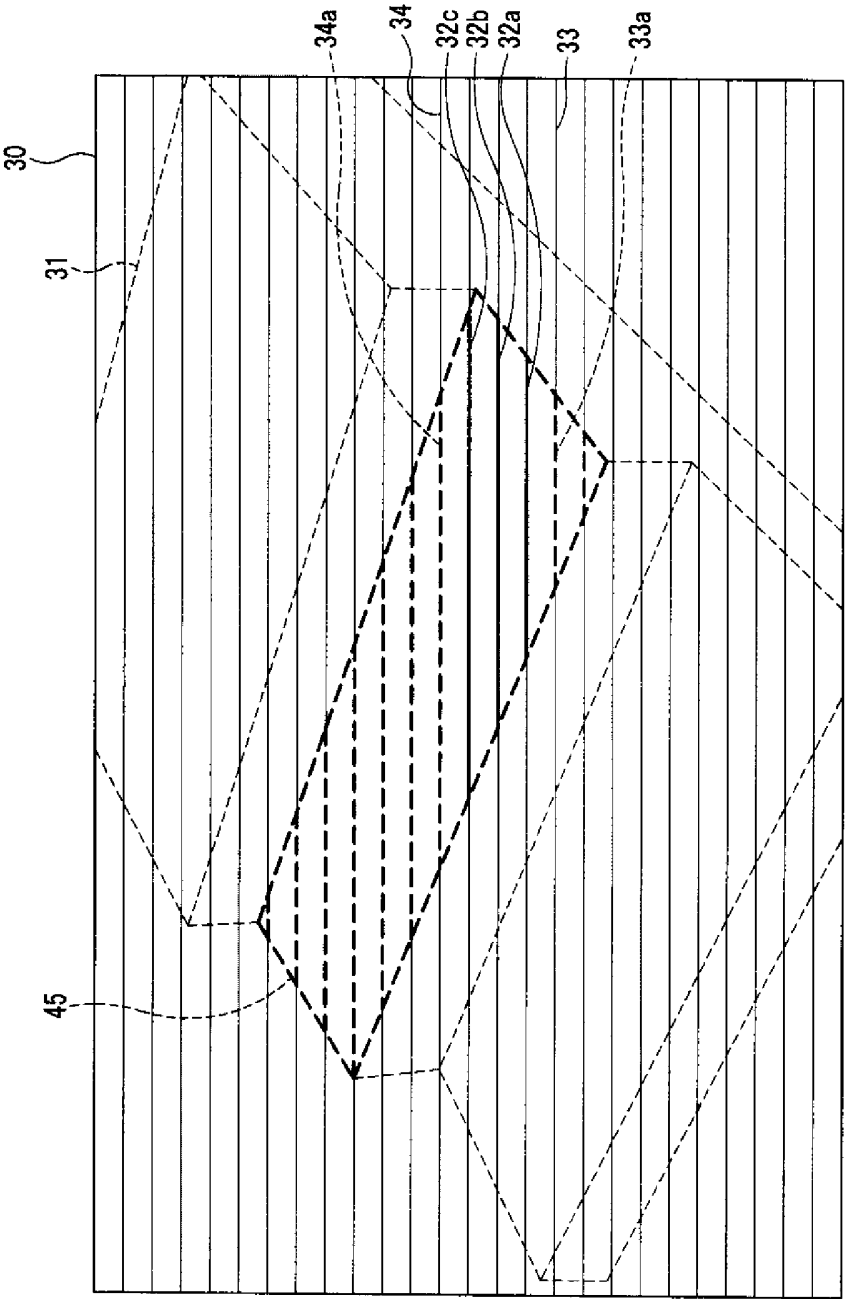


FIG.49

[図50]

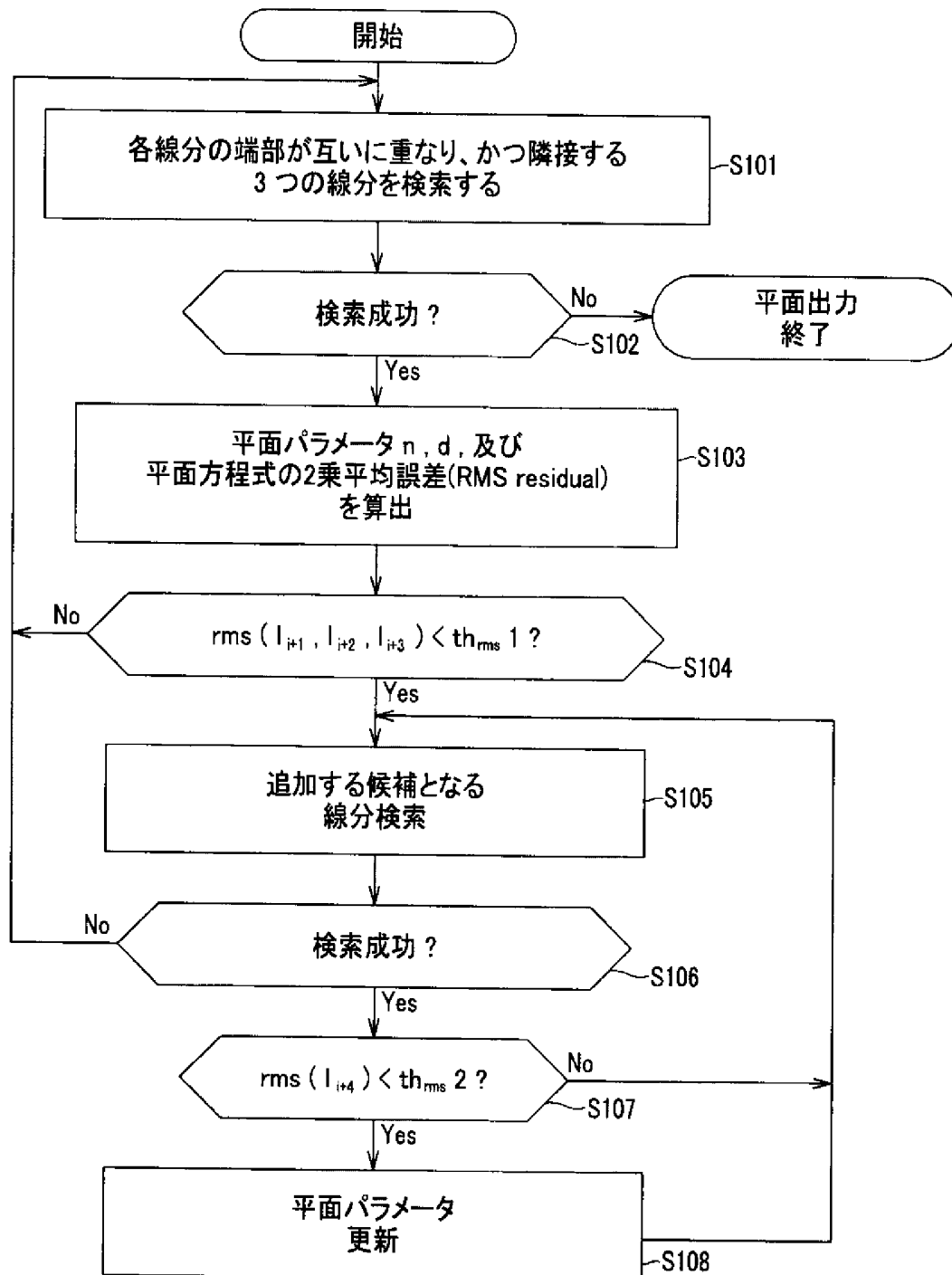


FIG.50

[図51]

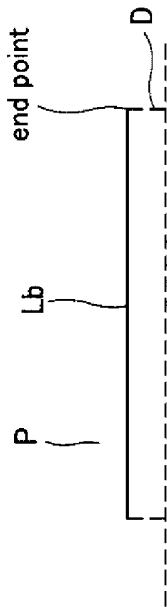


FIG.51B

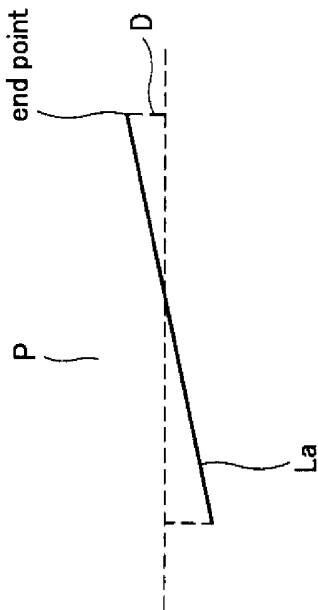


FIG.51A

[図52]

Algorithm *FindSeedRegion*Input: *lines[i]*: vector of lines for each image row (or column) *i*,*n*: number of image rows (or columns)

Output: set of lines (seed region) or empty set (no seed found).

Sequence:

```

for i = 0 to (n-3) do
  for l1 in lines[i] do
    for l2 in lines[i+1] do
      for l3 in lines[i+2] do
        if overlap(l1, l2) and overlap(l2, l3) then
          (n, d) = fitPlane(l1, l2, l3)
          if rms(l1, l2, l3) < threshlrms then
            seed = {l1, l2, l3}
            remove(l1, l2, l3)
            return seed
          endif
        endif
      endfor
    endfor
  endfor
endfor
return {}

```

FIG.52

[図53]

Algorithm *RegionGrowing*Input: *region* : set of lines as seed region,*lines[i]* : vector of lines for each image row (or column) *i*,*n* : number of image rows (or columns)

Sequence :

 $A = 0, b = 0$ for *l* in *region* do $(A,b) = add(A,b,l)$ endfor $(n,d) = solve(A,b)$ *open* = *region*while *not empty(open)* do $l_1 = select(open), open = open - \{l_1\}$ for *i* in *neighbor(index(l₁))* dofor *l₂* in *lines[i]* doif *overlap(l₁, l₂)* and $rms(l_2) < thresh2_{rms}$ then $region = region + \{l_2\},$ $(A,b) = add(A,b,l_2), (n,d) = solve(A,b)$ $open = open + \{l_2\},$ *remove(l₂)*

endif

endfor

endfor

endfor

plane = $\{n,d,A,b,region\}$ *planes* = *planes* + $\{plane\}$ **FIG.53**

[図54]

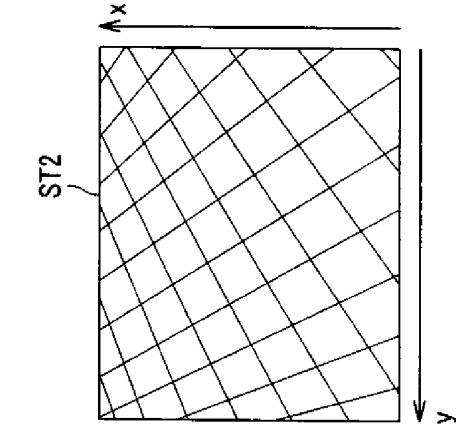


FIG. 54A

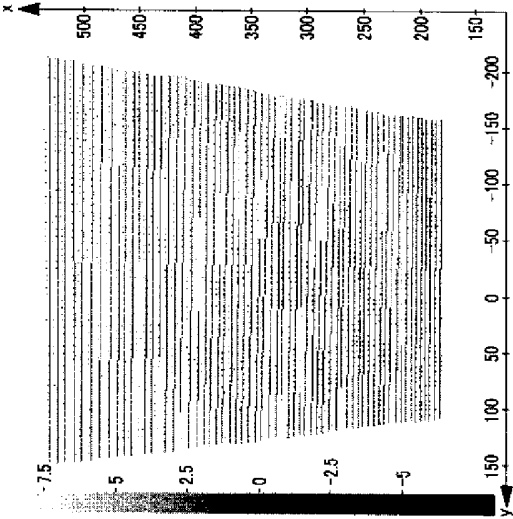


FIG. 54B

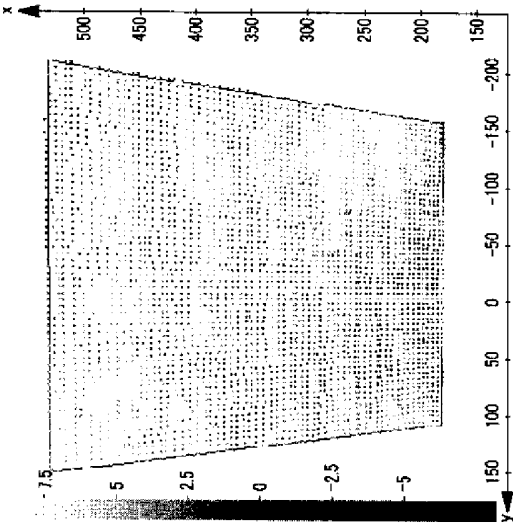


FIG. 54C

[図55]

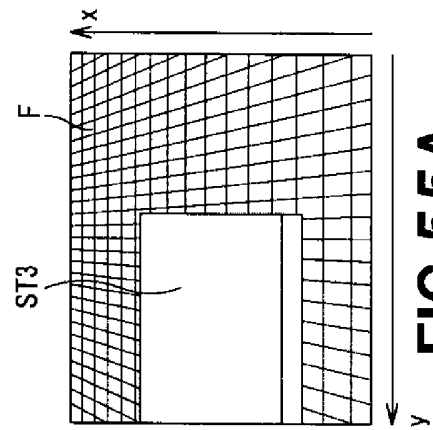


FIG. 55A

No	max_d	enable zig-zag	correct extraction (horizontal)	correct extraction (vertical)
1	30	no	0 / 10	0 / 10
2	25	no	0 / 10	0 / 10
3	20	no	10 / 10	0 / 10
4	15	no	10 / 10	3 / 10
5	10	no	10 / 10	10 / 10
6	30	yes	10 / 10	10 / 10

FIG. 55B

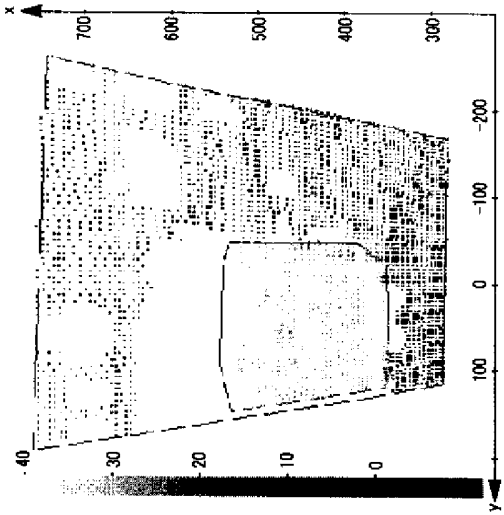


FIG. 55C

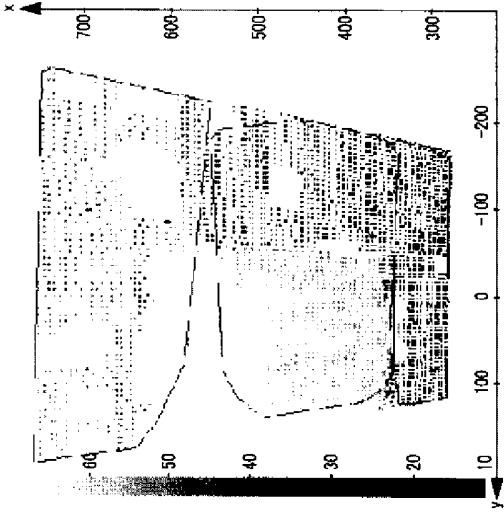


FIG. 55D

[図56]

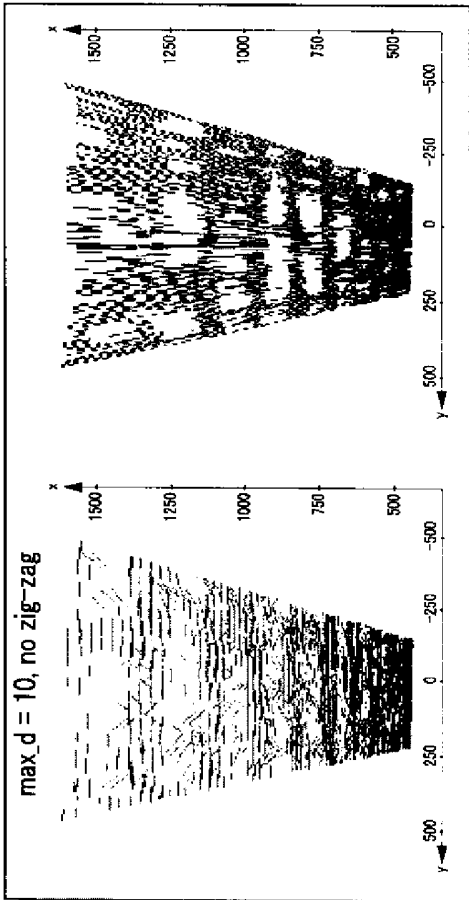


FIG. 56B

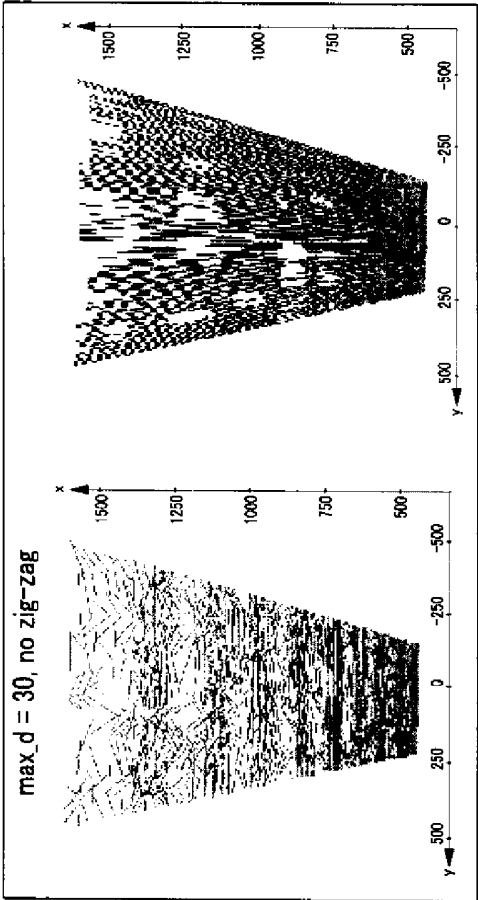


FIG. 56C

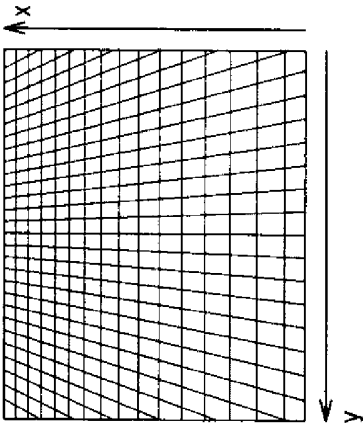


FIG. 56A

INTERNATIONAL SEARCH REPORT

International application No.

PCT/JP2005/004838

A. CLASSIFICATION OF SUBJECT MATTER

Int.Cl.⁷ B25J13/00, 5/00

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

Int.Cl.⁷ B25J1/00-21/02

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Jitsuyo Shinan Koho	1922-1996	Jitsuyo Shinan Toroku Koho	1996-2005
Kokai Jitsuyo Shinan Koho	1971-2005	Toroku Jitsuyo Shinan Koho	1994-2005

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

IEEE, JSTplus (JOIS), JST7580 (JOIS)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	Sei OKADA et al., "Jitsujikan Plane Segment Finder no Kenkyu", Dai 6 Kai Robotics Symposia Yokoshu, 18 March, 2001 (18.03.01), pages 51 to 56	1-4, 6, 7, 12, 17, 19, 21-24, 32-35, 37, 39, 40, 42, 44-47
Y		9-11, 16, 38
A		5, 8, 13-15, 18, 20, 25-31, 36, 41, 43
Y	JP 5-223549 A (Honda Motor Co., Ltd.), 31 August, 1993 (31.08.93), Par. Nos. [0009], [0020] to [0023]; Fig. 32 (Family: none)	9-11, 16, 38
A		1-6, 8, 12-37, 39-47

☐ Further documents are listed in the continuation of Box C.☐ See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search

30 June, 2005 (30.06.05)

Date of mailing of the international search report

19 July, 2005 (19.07.05)

Name and mailing address of the ISA/
Japanese Patent Office

Authorized officer

Facsimile No.

Telephone No.

A. 発明の属する分野の分類 (国際特許分類 (IPC))

Int.Cl.⁷ B25J13/00, 5/00

B. 調査を行った分野

調査を行った最小限資料 (国際特許分類 (IPC))

Int.Cl.⁷ B25J1/00-21/02

最小限資料以外の資料で調査を行った分野に含まれるもの

日本国実用新案公報	1922-1996年
日本国公開実用新案公報	1971-2005年
日本国実用新案登録公報	1996-2005年
日本国登録実用新案公報	1994-2005年

国際調査で使用した電子データベース (データベースの名称、調査に使用した用語)

IEEE, JSTPlus(JOIS), JST7580(JOIS)

C. 関連すると認められる文献

引用文献の カテゴリー*	引用文献名 及び一部の箇所が関連するときは、その関連する箇所の表示	関連する 請求の範囲の番号
X Y A	岡田慧、外4名、実時間ブレンセグメントファインダの研究, 第6回ロボティクスシンポジウム予稿集, 2001/03/18, p. 51-56	1-4, 6, 7, 12, 17, 19, 21-24, 32-35, 37, 39, 40, 42, 44-47 9-11, 16, 38 5, 8, 13-15, 18 , 20, 25-31, 36 , 41, 43

☒ C欄の続きにも文献が列挙されている。☐ パテントファミリーに関する別紙を参照。

* 引用文献のカテゴリー

「A」特に関連のある文献ではなく、一般的技術水準を示すもの

「E」国際出願日前の出願または特許であるが、国際出願日以後に公表されたもの

「L」優先権主張に疑義を提起する文献又は他の文献の発行日若しくは他の特別な理由を確立するために引用する文献 (理由を付す)

「O」口頭による開示、使用、展示等に言及する文献

「P」国際出願日前で、かつ優先権の主張の基礎となる出願

の日の後に公表された文献

「T」国際出願日又は優先日後に公表された文献であって出願と矛盾するものではなく、発明の原理又は理論の理解のために引用するもの

「X」特に関連のある文献であって、当該文献のみで発明の新規性又は進歩性がないと考えられるもの

「Y」特に関連のある文献であって、当該文献と他の1以上の文献との、当業者にとって自明である組合せによって進歩性がないと考えられるもの

「&」同一パテントファミリー文献

国際調査を完了した日

30.06.2005

国際調査報告の発送日

19.7.2005

国際調査機関の名称及びあて先

日本国特許庁 (ISA/J P)

郵便番号100-8915

東京都千代田区霞が関三丁目4番3号

特許庁審査官 (権限のある職員)

八木 誠

電話番号 03-3581-1101 内線 3324

3U

3504

C (続き) . 関連すると認められる文献		
引用文献の カテゴリー*	引用文献名 及び一部の箇所が関連するときは、その関連する箇所の表示	関連する 請求の範囲の番号
Y A	JP 5-223549 A (本田技研工業株式会社) 1993. 08. 31, 段落【0009】 , 【0020】 - 【0023】 , 図 32 (ファミリーなし)	9-11, 16, 38 1-6, 8, 12-37, 39-47