



Title	視覚システムによる移動ロボットの廊下内位置認識
Author(s)	恩田, 邦夫; Onda, Kunio; 松島, 緑朗 他
Citation	北海道大學工学部研究報告, 142, 21-33
Issue Date	1988-10-31
Doc URL	https://hdl.handle.net/2115/42132
Type	departmental bulletin paper
File Information	142_21-34.pdf



視覚システムによる移動ロボットの廊下内位置認識

恩田 邦夫 松島 緑朗* 青木 由直

(昭和63年6月30日受理)

Determination of a Robot Position in a Corridor by a Visual System

Kunio ONDA, Rokurou MATSUSHIMA and Yoshinao AOKI

(Received June 30, 1988)

Abstract

This paper gives a method for determining a robot position in a straight corridor. In the method 3-dimensional image of the corridor is reconstructed from a perspective scene and the robot position is estimated by a pattern matching between the reconstructed image and an environmental map. For the reconstruction the method detects a vanishing point in the scene, determines a section of the corridor and calculates the distance to both sides of the wall surface. For the matching we used an arrangement of objects on the wall such as doors, windows and so on. These objects are expressed as compressed pattern data in the map for the fast matching and memory saving. And the reconstructed image is also compressed in the same manner as the map data. Experimental results shown in the paper gave satisfying precision for navigation of the robot.

1. ま え が き

移動ロボットの自律走行において重要となる機能の一つに自己の現在位置を認識する、いわゆる自己定位の機能がある。その実現のためにこれまで多くの方式や実験結果の報告が行われてきているが、特に近年は視覚システムを用い、自然環境を対象とした画像処理に基づく自己定位が盛んに試みられている^{(1)~(8)}。しかし従来の自己定位は、ロボットの移動開始位置または目標到達位置が既知の場合を想定し、これらの位置とロボットの現在位置との間の相対的位置関係、あるいは移動の開始前後の移動情報を得ることが目的であり、相対的自己定位といえるものである。

一方ロボットが環境内の任意の位置に置かれ、そこから出発の指令を受けた場合、ロボットはまず環境内での自分の位置を知らなくてはならない。このような絶対的自己定位の方法としては、経路上の白線⁽¹⁾やランドマーク⁽⁹⁾に位置情報を表す特別なコードを付加したり、電波灯台など⁽¹⁰⁾を利用するなど考えられるが、走行環境にはかなりの制約がある。また、相対的自己定位と同様に視覚システムによって周囲の環境認識を行い、この結果に基づき絶対的自己定位を行う方法も提案されているが⁽⁷⁾、実際の長距離の経路での実験は行っていない。

我々は先に移動ロボットの走行環境を建物の廊下内部に限定した上で、単眼視で撮像した廊下

の情景からその構造を3次元的に復元する試みを報告したが⁽¹¹⁾、静止、単眼という撮像条件下でも、廊下内に存在するドア、窓、配電盤などの配置をある程度正確に復元できることが分った。本報告では、この結果を基に廊下の環境地図内でのロボットの絶対的自己定位を行う方法を提案し、実験結果を示す。以下、この自己定位の方針を述べ、情景画像に対する処理および環境地図との照合方法を、実験結果を例示しながら説明する。

2. 自己定位の方針

絶対的自己定位の基本的考え方は、ロボットの現在の姿勢と認識した周囲の環境を基に、この環境に適合するような位置を環境地図内から検索するものである。ここで環境内の対象物やその特徴量、照合方法には種々の選び方がある。以下では、我々の対象とする物体および照合方法の方針を述べる。

2. 1 環境認識の対象

環境認識の対象の選択は、照合方法さらには処理時間とも関連している。誘導の際行われる相対的自己定位では、実時間処理の要求や、移動前後の変位が求められればよいことから環境内の特徴点^{(2)~(4)}や物体のエッジ^{(5),(6)}が利用されている。しかし絶対的自己定位の場合、対象物およびその特徴量は、そのものが環境内での位置を定めるに十分な情報量を持っていることが必要であり、エッジや特徴点ではその物体の識別や位置認識は行いにくい。

一方我々が先に行った廊下の環境復元⁽¹¹⁾では、静止、単眼という撮像条件下であっても、廊下内に存在するドア、窓、配電盤などの配置をある程度正確に復元できることが分った。このような物体は、その配置が環境内で固有成り位置不変のものであり、絶対的自己定位のための有力な情報となると考えられる。またこのような物体を利用しての自己定位は、

- (1) ロボットの走行のために環境に手を加えない、
- (2) ロボット内部の地図情報の変更がない（またはきわめて少ない）、
- (3) 物体の位置情報（設置箇所の高さや大きさなど）や色情報を制約条件としての物体の抽出や識別が容易になる可能性がある、

などの利点もある。しかし情景画像内の物体の識別までを行うことは、カメラの解像度や照明環境などの影響で実際にはそう簡単ではない。そこで本報告では、先に述べたような物体が壁や天井などの背景に対し明度が異なることに着目し、その明度情報による物体配置を環境照合の対象とすることにした。

2. 2 照合方法

環境照合では、ロボットが観測した情景と環境地図との間での照合が行われるが、当然ながら照合レベルでの双方のデータ表現は同一でなければならない。この場合、照合レベルをどこに設定するかによってロボットの得た情景画像や環境地図に対する前処理も異なってくる。

我々と同様の対象物体を利用した自己位置認識には既に坪内ら^{(7),(8)}の報告があり、1.5次元的情報と呼ぶデータ表現による照合方法を提案している。ここでは、ドアや柱などのように実際は長方形である物体が、透視変形を受けた情景画像中では台形となることを利用し、色情報を基に抽出したこれらの物体に対し台形近似を施し、更に台形の形状情報をパラメータとして取り出してこれを実視野情報とする。一方、世界地図と呼ぶ環境地図に対してもロボットの位置および姿勢情報を基に透視変換を施し、実視野情報と同様のパラメータを取り出してこれを推定視野情報とする。環境照合はこの実視野情報と推定視野情報との間で行われる。彼らはこの方法を誘導中の相対的自己定位に利用しているが、絶対的自己定位にも適用できるとしている⁽⁷⁾。しかし絶対的

自己定位では、ロボットが得ることのできる実環境情報は1通りないしきわめて少ないのに対し、照合相手となる推定環境情報は最悪の場合環境地図の全域を対象にしなければならず、その照合の組み合わせは膨大な数になる。従って、照合のたびに環境地図に対し照合レベルへの歩みよりの処理を施すような方法は処理時間の面で得策ではなく、実環境情報に対する前処理に多少時間がかかっても、照合レベルを環境地図により近いレベルにした方がよいと思われる。

我々の提案する方法では、復元画像を格子状のパターンで表現し、更に明度値についても、背景の明るさを中心明度としてこれより明るいかによって3値で表現することによりデータ圧縮を図る。一方、環境地図も廊下面内の物体配置を同様のパターンで表現し、テンプレートマッチングによって環境照合を行う。このため環境地図に対しては変換などの処理を行わない。また坪内らの台形近似は、環境内の物体が長方形であることが前提になっているが、本方法では比較的大きな物体であれば長方形でない物体も環境照合の対象とすることができる他、ノイズなどで分断された物体に対する扱いも容易である。

本方法での処理の流れをまとめて図1に示す。以下では実験例を示しながら各処理を説明する。

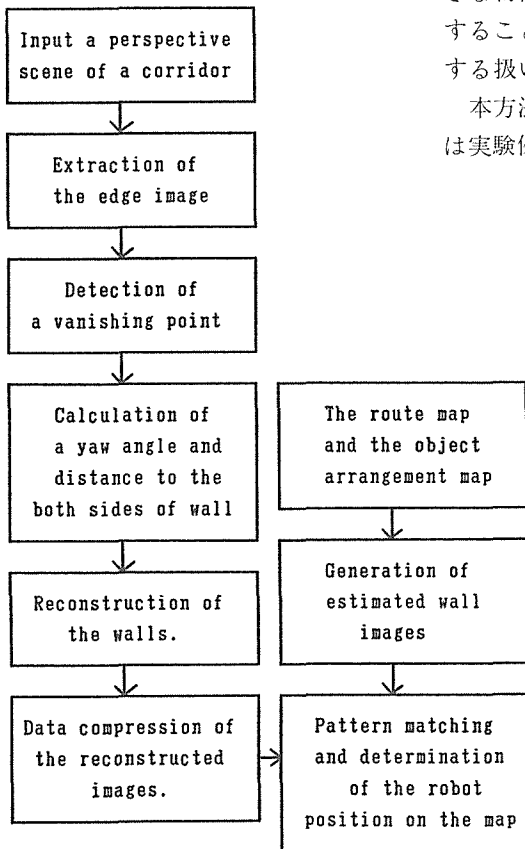


図1 処理の流れ

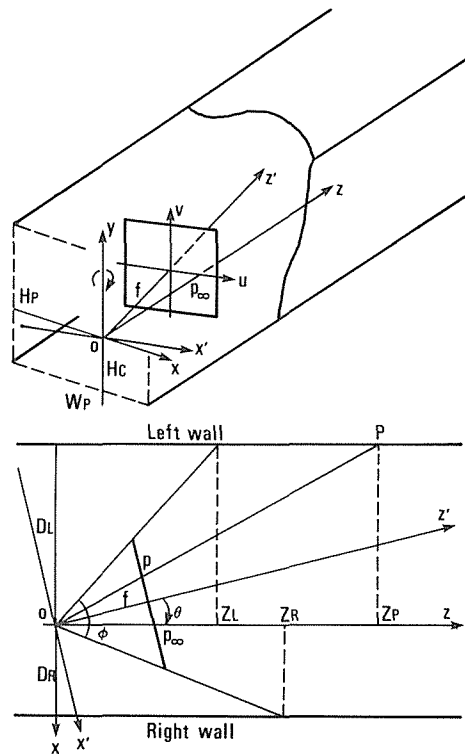


図2 廊下内ロボットの座標系

3. 情景画像に対する処理

3.1 廊下内環境の3次元的復元

静止状態で単眼視で得た廊下の情景画像から、壁や天井などの面情報の復元を行う。本処理は文献(1)での廊下を対象とした情景復元の方法を利用しており、再度その概略を述べる。

対象とするロボットの走行環境は直線状の廊下とし、以下の前提条件を設ける。なお図2の座

標系を考える。

- (1) 廊下内には視野を遮るような大きな障害物はない。
- (2) 廊下の高さ (H_p) や幅 (W_p)、カメラの焦点距離 (f) は既知情報として与えられる。
- (3) カメラはロボット上に床と水平に固定され、床からの高さ (H_c) も既知とする。
- (4) カメラの回転成分は、 Y 軸まわりのヨー角 (θ) だけとする。

情景画像に対する処理は、始めに画面内から無限遠点を発見し、この位置情報からヨー角を求める。我々が先に提案した方法⁽¹²⁾で発見した画面内の無限遠点位置を $p_\infty(u_\infty, v_\infty)$ とすれば、先の拘束条件下では、

$$\theta = -\tan^{-1}\left(\frac{u_\infty}{f}\right) \quad (1)$$

の関係があり、これより θ を求めることができる。

次に廊下の壁、天井および床が交差してつくられる境界線を抽出し、廊下の断面を決定する。このとき既知情報の H_p/W_p 比および H_p/H_c 比を判定条件として利用する。

境界線が決るとカメラから壁までの距離が計算できる。すなわちカメラから左右の壁までの距離をそれぞれ D_R, D_L 、無限遠点を原点として第 i 象現の境界線の傾きを ψ_i とすると、

$$\begin{aligned} D_R &= (H_p - H_c) \cos\theta / \tan\psi_1 \\ &= -H_c \cos\theta / \tan\psi_1 \\ D_L &= (H_p - H_c) \cos\theta / \tan\psi_2 \\ &= -H_c \cos\theta / \tan\psi_3 \end{aligned} \quad (2)$$

として求まる。

更に、画面上の点 $p(u, v)$ が、実空間における壁、天井または床の面上の点 $P(x, y, z)$ に対応した点であるとみなすと、式(3)の逆透視変換によって u, v, θ, f から x, y, z の比例関係を求めることができ、 x, y, z の1つが分かれば残りの座標が計算できる。

$$\frac{x}{f \sin\theta + u \cos\theta} = \frac{y}{v} = \frac{z}{f \cos\theta - u \sin\theta} \quad (3)$$

ここで点 P が無限遠点を原点として第1象現と第2象現の境界線に挟まれる領域にあるとき、この点は天井面の点とみなし、 $y = H_p - H_c$ を(3)に代入して x, z 座標を得ることができ、また第2象現と第3象現の境界線に挟まれる領域にあるとき、この点は左壁面の点とみなし、 $x = D_L$ を式(3)に代入して y, z 座標を得ることができる。他の点についても同様に考えることができ、壁、天井および床の復元が行える。但し、両壁での復元可能な最近景距離 Z_L, Z_R は、カメラの画角 ϕ と D_L, D_R および θ で決定され、

$$\begin{aligned} Z_L &= D_L / \tan(\phi/2 - \theta) \\ Z_R &= D_R / \tan(\phi/2 + \theta) \end{aligned} \quad (4)$$

となる。従って $|\theta| > \phi/2$ のとき、片側の壁しか復元できない場合もあるが、 θ を知ることでロボットを正面向きに制御しなおすことは可能であり、本報告では θ は十分小さい値であるとする。

図3は $f = 12.5$ mm で撮像した廊下の情景画像の一例であり、画素数は 256×240 ドット、濃淡レベルは、256階調で得ている。この画像に、計算によって求めた無限遠点、廊下の境界線候補、廊下の断面を重ねて表示してある。またこのときの撮像条件、実際のカメラ位置、計算によって求められたカメラの向き、カメラと壁までの距離などを表1に示す。

更に図4はこの計算結果をもとに廊下の両壁を各々 Z_L, Z_R から 10 m 区間分復元したものであ

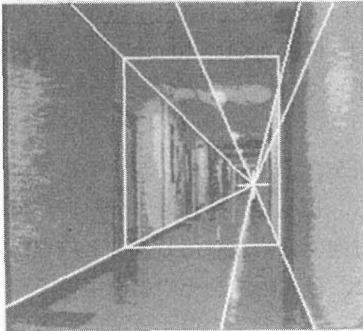


図3 廊下の情景画像の一例

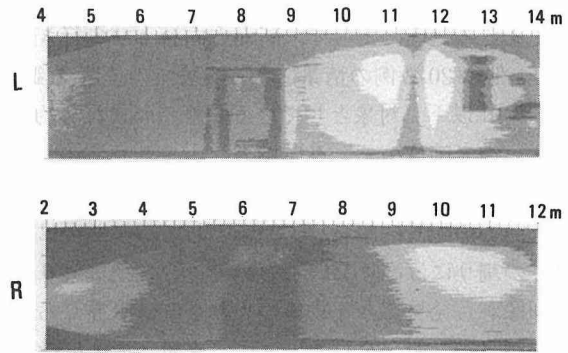


図4 両壁の復元画像

表1 情景についての諸データならびに情景解析から得た位置情報

	実測値	計算値
焦点距離 (f) [mm]	12.5	—
カメラの高さ (H _C) [cm]	93	—
廊下の高さ (H _P) [cm]	259	—
廊下の幅 (W _P) [cm]	246	—
右壁までの距離 (D _R) [cm]	40	40.8
左壁までの距離 (D _L) [cm]	206	205.2
ヨー角 (θ) [°]	-7.97	-7.81

り、復元画像は50ドット/mで表示してある。但し Z_L, Z_R はm単位で切上げしている。復元では、画面上の点Pの画素濃度を式(3)の逆透視変換によって点P上にマッピングしている。点Pは画像メモリ内では離散的な点であるため、復元画像には透視画面の画素との対応がつかない「すきま」ができる。壁の奥行き方向でのこの「すきま」間隔は復元距離のほぼ2乗に比例して広がるため、遠方ほど復元画像の分解能は劣化する。図4はこの「すきま」を上下・奥行き方向とも1次補間

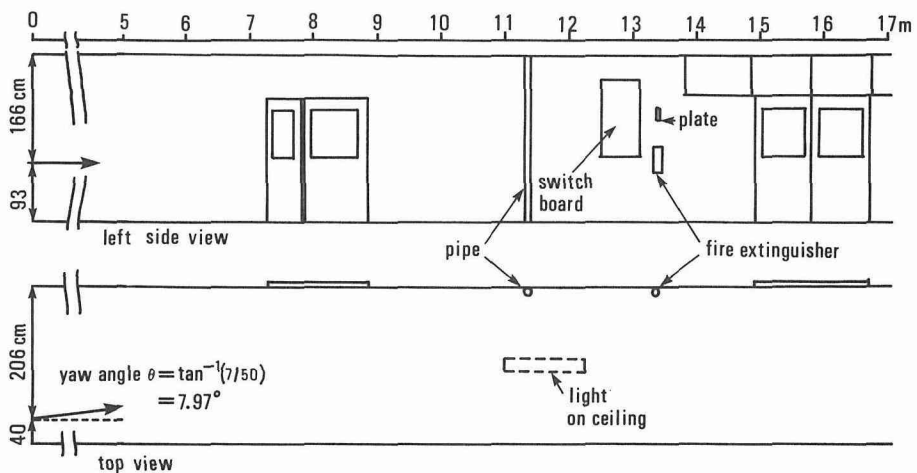


図5 実際の左壁

したものである。また図5はこの復元部分の実測図である。

ここまでの処理によって、断面方向および奥行き方向についての相対的自己定位が行えたことになり、他の20数例の情景による実験での位置認識の誤差は、断面方向で10 cm以下、奥行き方向で50 cm以下（対象としている距離のいずれも約5%以下）であった。

3. 2 復元画像のデータ圧縮

復元画像には、廊下内のドアや窓などの面物体の配置情報が含まれている。我々の提案する方法では、この配置情報をパターンとして環境照合に利用する。しかし、原画像の分解能が情景の複雑さの割りには高くないことや逆透視変換により奥行き方向の分解能が落ちることの理由で、復元画像から発見できる物体の大きさには限界がある。また、復元画像をそのまま用いて環境地図とのパターン照合を行うにはデータ量が膨大になりすぎる。そこでこの復元分解能を考慮し、適当なデータ圧縮を図ることとした。

ここでは復元画像を断面方向について8分割(約30 cm)、奥行き方向について20 cm間隔で分割し、更にこの格子状の領域内部の明度についても3値化している。このデータ圧縮により、例えば10 m区間の復元画像は100バイトのパターンデータとなり、後で述べる環境地図のデータ量も軽減できる。但し、この格子の大きさについての定量的な根拠はなく、特に断面方向を8分割としたのは、計算機内でのデータ長が8ビットを単位とすることによる。

図4で示した復元画像を例に、このデータ圧縮の手順を説明する。なおここでの復元画像は50画素/mとしたので、128(断面方向)×500(奥行き方向)画素、各画素256階調で得ている。

(1) 廊下の断面方向(壁の場合上下方向)は、奥行き方向に比べ距離が短く、後述する照明光の変化の度合いが大きくない。そこで断面方向のデータ圧縮を先に行う。廊下の断面方向を等間隔で8分割し、この区間(16画素に相当)の濃淡値の平均を求める。この結果図6のような明度値の分布 $F'(i, j)$ ($i=1, \dots, 8, j=1, \dots, 500$)を得る。

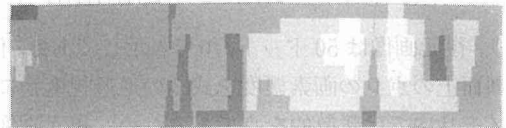


図6 断面方向のデータ圧縮

(2) ここで得られた明度分布は、照明光の分布が畳重されたものであり、物体の真の明度分布とならない。そこで本方法では、壁面に当たった照明光の分布の変化が物体からの反射光の変化に比べゆるやかであることに着目し、以下のように照明光の影響の除去を行う。先ず明度分布を長い窓(ここでは2 mの距離分に相当する窓)を使って平滑化し、照明光の分布 $E(i, j)$ とみなす(図7)。次に $F'(i, j)$ を次式のように差引き、これを真の物体分布 $F(i, j)$ とする(図8)。



図7 照明光の分布

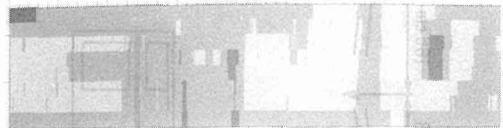


図8 照明光除去後の画像

$$F(i, j) = F'(i, j) - E(i, j) + 128 \quad (5)$$

(3) 濃淡画像の2値化方法としてよく知られるモード法を応用して、明度の3値化のためのしきい値 g_{-1} , g_{+1} を決定する。先ず図9のような $F(i, j)$ の明度ヒストグラムを作成する。

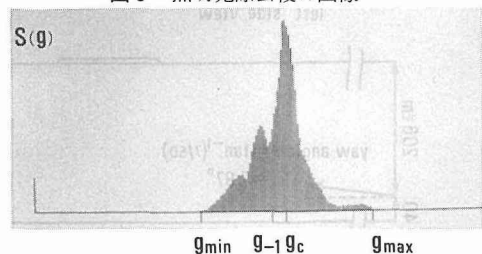


図9 明度ヒストグラム

ここで、明度を g 、その明度をもつ画素数を $S(g)$ とする。復元画像中では壁や正井などの背景が占める面積が最も大きい。従ってヒストグラムではこの背景部分に相当する凸峰ができる。そこでこのピーク値を示す明度を背景の中心明度 g_c とする。またもし大きな面積を占める暗い物体があれば、 g_c 以下のヒストグラムにも同様の凸峰ができる。この凸峰と先の背景の凸峰との谷間の最低値 g_{-1} を、背景と背景より暗い物体とを分離するしきい値とする。但し、凸峰があってもその大きさが t_0 以下の場合には凸峰とはみなさない。背景より明るい物体についてのしきい値 g_{+1} も、同様に決定する。実験では $t_0 = g_c * 0.1$ とした。

しかし、背景と物体とのコントラストが低い画像ではこの凸峰が明確に生じない場合もある。そこでヒストグラム $S(g) > 0$ でかつ g が最小および最大の明度を各々 g_{min} 、 g_{max} としたとき、 $|g_c - g_{min}| > t_1$ あるいは $|g_c - g_{max}| > t_2$ の場合は、凸峰がなくても背景より暗い(あるいは明るい)物体が存在しているものとみなし、次のようにしきい値を決定する。

$$\begin{aligned} g_{-1} &= (g_c + g_{min}) * 0.5 \\ g_{+1} &= (g_c + g_{max}) * 0.5 \end{aligned} \tag{6}$$

(4) $F(i, j)$ を(3)で得たしきい値 g_{-1} 、 g_{+1} により 3 値化する (図 10)。すなわち、

$$g = \begin{cases} 0 : g_{-1} \leq g \leq g_{+1} \text{ (背景)} \\ 1 : g < g_{-1} \text{ のとき (暗い)} \\ 2 : g > g_{+1} \text{ のとき (明るい)} \end{cases} \tag{7}$$

とし、また復元の際欠けた部分 (図 4 での左上の隅) については、未決定部分として $g = 3$ とする。

(5) 最後に $F(i, j)$ を奥行き方向について 20 cm (10 画素に相当) 毎にデータ圧縮を行う。10 画素の明度値の中で最も頻度の多い明度値をこの区間の明度値とする。

以上の処理により、最終結果である図 11 の復元パターンを得る。実際のこのパターンは $8 \times 50 \times 2$ ビットのデータである。

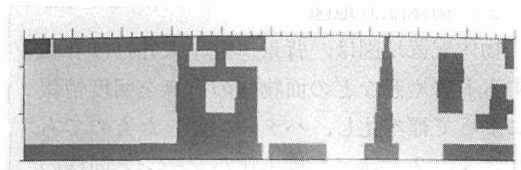


図 10 明度のデータ圧縮

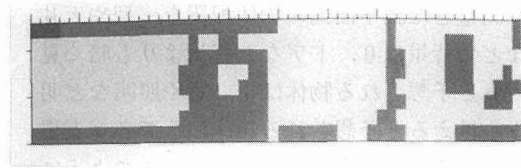


図 11 奥行き方向のデータ圧縮

4. 環境地図との照合

4. 1 環境地図の内部表現

自己定位のために、経路地図と物体配置地図と名付けた環境地図を用意した。

(1) 経路地図

経路地図は、廊下の結合関係、交差点や分岐点の絶対座標などを表現したもので、以下の制限がある。

- (1) 経路は直線状の廊下の結合により構成される。
- (2) 交差点では 4 本以下の廊下と直角に交わる。

経路はノードとリンクで表現され、ノードは交差点や廊下の端を意味し、リンクは廊下を意味する。

ノードは、ノードの型、ノードの座標、接続するリンク番号の情報を持つ。ノードの型には、十字路交差点、T字路交差点、曲がり角、中継点および端点の 5 種類があり、順にコード番号 1 ~ 5 で表す。中継点は廊下の幅や高さに変化する場所を示す。また接続するリンク番号の領域は

常に4個で、リンク間の相対的な接続関係も表現する。

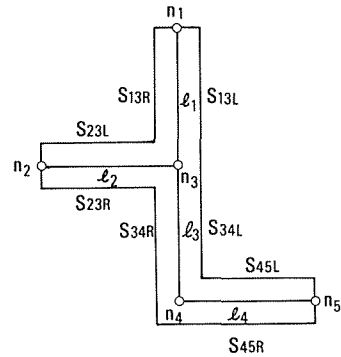
一方リンクは廊下の形状について、リンクの型、高さおよび幅、距離、接続するノード番号(始点および終点)、物体配置地図の格納ファイル名の情報を持つ、リンクの型は、ロボットの走行が可能な廊下か、走行が不可能または禁止されている廊下かをコード番号1, 2で表す。コード番号2の経路はロボットの走行には関係ないが、情景中には見える可能性があり、自己定位のための情報として利用する。接続するノード番号の始点、終点は物体配置地図の始点、終点にそのまま対応し、距離はそのデータ長に対応している。

図12に経路の一例と、その内部データ表現を示す。経路地図は具体的にはC言語の構造体で実現している。

(2) 物体配置地図

物体配置地図は、背景(型や天井)に存在するドアや窓などの面物体の有無を明度情報によって標準化し、パターン化したものであり、3.2で述べた標準化パターンと同様の4値情報で表現されている。すなわち、実際に測量した廊下面内の物体配置を、型や天井などの背景は0、ドアなど背景よりも暗く見ると予想される物体は1、窓や照明など明るく見ると予想される物体は2で各々表現している。また廊下の分岐やホールのように見え方や明度が一意に決定できない部分は3としている。

この物体配置地図は、あるノードから隣のノードまでの区間についてのパターンが1つのファイルに保存される。例えば図13は工学部内の約100mの長さの経路であるが、この経路には中間地点に廊下の分岐があり、物体



node	type	coordinates	connect_link
n1	5	X2, Y2	1, 0, 0, 0
n2	5	X1, Y1	2, 0, 0, 0
n3	2	X2, Y2	1, 2, 3, 0
n4	3	X2, Y3	3, 0, 0, 4
n5	5	X3, Y3	4, 0, 0, 0

link	type	h	w	d	connect_node	fname_of_OAM
l1	1	H	W	D1	1, 3	S13
l2	1	H	W	D2	2, 3	S23
l3	1	H	W	D3	3, 4	S34
l4	1	H	W	D4	4, 5	S45

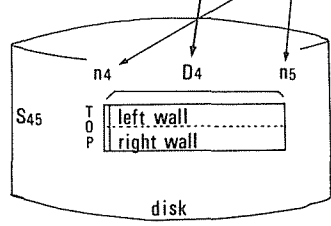


図12 (a)廊下の例と(b)環境地図の内部表現

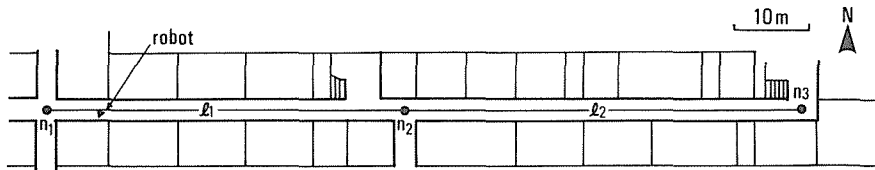


図13 実験を行った工学部内の廊下

配置地図はリンク1₁(n₁~n₂)と1₂(n₂~n₃)に分割されて登録される。図14はこの区間の壁の物体配置地図の一部であり、地図作成のための編集ツールを用いて作成したものである。この編集ツールは、編集画面上でマウスを用いて4値データを設定していくことができ、またドアや窓などの基本的な物体は「部品」としてパターンが用意されており、これらを出して配置してい

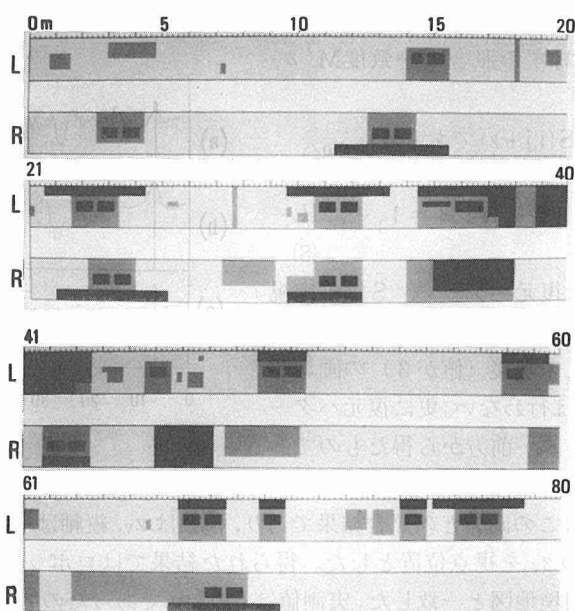


図 14 物体配置地図

けばよいようになっている。

4. 2 環境地図との照合

工学部の廊下の総延長は1階部分で約1000mであり、この物体配置地図に必要な総メモリ容量は壁、床、天井を合せて約40kバイトとなる。しかし現在のところ、工学部内の全経路についての環境地図は作成されていないため、本報告では図13に示した経路について行った照合実験の結果を示す。

環境地図との照合では、復元パターンと推定視野パターンとの間で照合を行う。この推定視野パターンの生成では以下の点を考慮する必要がある。

- (1) ロボットが図の経路内のどこかにおり、左壁の標本化パターンとして P_L が得られているとする。ここでロボットが 1_1 上にいると仮定して照合を行う場合、得られた情景が n_1 の方向を見たものなのか、 n_2 の方向を見たものなのかも分っていないので、結局 1_1 の物体配置地図 S_{12R} と S_{12L} の双方と照合を行う必要がある。但し、 S_{11R} はノード n_1 から n_j を見たときの右側の壁のパターンを表し、同様に S_{11L} は左側の壁のパターンを表す。ちなみに $S_{11R}=S_{11L}$ である。
- (2) ロボットが n_2 の方向を見ており、かつ n_2 に近い位置にいるとき、 P_L には 1_1 と接続する 1_2 の左壁も復元されている可能性がある。従って、ここでの照合では、 S_{12L} に S_{23L} を結合した地図との間でも照合を行ってみなければならない。

これらをまとめると、図における環境照合の組合せは、 $n_1 \rightarrow n_3$ の方向を向き、この区間内にいる場合とその逆の場合を考慮しなければならない。

推定視野パターンの生成では、このように2つ以上のリンクにまたがる物体配置地図の結合を行う。但し、この結合はリンクが対向している場合だけしか行わない。ロボットが端点や曲り角に向かっている場合、本方法では画像復元自体が行えないためである。従ってこのような場合の自己定位は本方法では扱えない。

図15は、ロボットが $n_1 \rightarrow n_3$ の方向を向いているものとして、図11の復元パターンと図14の

推定視野パターンとの照合を行った結果である。照合はテンプレートマッチングで行っており、奥行き方向の距離 z に対する照合の一致度 $M(z)$

は、

$$M(z) = \frac{(P(i,j)=S(i,j+z) \text{ である数})}{(\text{照合した数})} (\%)$$

$$i = 1, \dots, 8, j = 1, \dots, 50, z = 1, \dots, ML - 50 \quad (8)$$

としている。但し、 P は復元パターン、 S は推定視野パターン、 ML はその区間の距離である。また双方のパターンにはそれぞれ未定義（値が3）の画素があり、これに対する照合は行わない。更に復元パターンは視点から Z_L (または Z_R) 前方から得たものであるため、視点位置 z_v は z からこの Z_L (または Z_R) 分の距離を差し引いたところにある。

図15で(a)は左壁、(c)はこの両方との照合結果であり、横軸は z_v 、縦軸は一致度 M を表している。 M が最大値を持つときの z_v を視点位置とした。得られた結果ではロボットは n_1 から 680 cm の地点にいる場合が最も環境地図と一致した。実測値は 670 cm であったので、誤差は 10 cm であった。

この結果は、復元パターンおよび物体配置地図の標本化間隔から最低でも 20 cm の誤差は避けられないこと、実測した距離データにも 10 cm 程度の誤差があること、約 100 m の廊下を対象としたものでありロボットの自己定位を行うには十分な精度であることなどから、満足すべきものであるといえる。更に絶対的位置の決定は、この視点位置から、そのときのリンク番号、そのリンクの両端のノード番号、そのノードの座標を順に経路地図内でたどることにより行うことができる。

4. 3 他の情景についての実験結果

撮像位置を図13の経路内の n_1 から 5 m 間隔で 20 m 移動し、同様の位置認識を試みた。図16はこれらの情景、図17はこの一致度のグラフと推定された視点位置を表す。結果はほぼ正しい位置が得られている。しかし、図で一致度のほぼ等しい複数のピークが生じていることでも分かるように、対象とした廊下では、ドアなどの物体配置が同一形状でかつ繰り返し形で並んでいるため、片側の壁だけでの照合では誤った位置認識をした例もあった。対象とした廊下では両壁の物体配置に多少のずれがあったため、両壁の照合による総合的判定で良好な結果を得ることができたが、両壁が同一配置をしている場合や、異なるフロアで同一配置をしている場合などでは、本方法だけでは位置を唯一に定めにくい。

また図18のように情景内に廊下中央部のホールが見えているような場合では、復元画像が正しいものとならず位置認識が正しく行えなかった。本方法では、このように廊下内にホールや廊下の分岐がある場合の環境照合にはまだ問題が残されている。更に窓などのように撮像時刻や天候などによって明度が異なる物体の扱いや、窓からの光や影などのように照明光が急に変化する情景の扱いについても今後の課題である。

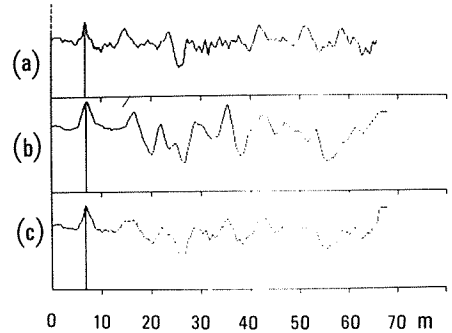
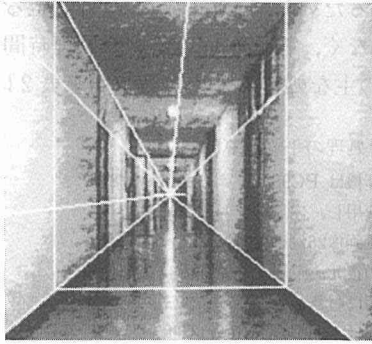


図15 照合結果



(a) $Z_v=50\text{m}$



(b) $Z_v=10\text{m}$



(c) $Z_v=15\text{m}$



(d) $Z_v=20\text{m}$

図 16 他の情景画像 (Z_v は n_1 から視点までの距離)

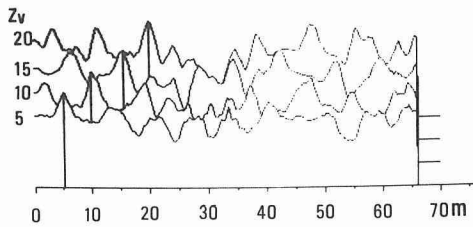


図 17 位置認識の結果



図 18 位置認識がうまくいかない情景の例

5. む す び

本報告では、ロボットが環境内の任意の位置に置かれた場合に必要となる絶対的自己定位の一方法を提案し、実験結果を示した。ここでは、静止、単眼視という条件下でも環境内での絶対的自己定位がある程度可能なことが示された。

本方法は、照合レベルを地図に近いところで行っているため、情景画像から照合に至る前処理に時間を要するが、環境地図に対しての処理はほとんどなく、自己定位のための処理時間は前処理に要する時間ではほぼ決定される。実験した 20 数例での主な処理の平均実行時間を表 2 に示す。

但し、ここでの時間はロボットが実際に自己定位を行う場合には必要のない、結果の画面表示などを省略してある。現在 16 ビットパソコンを使用して処理を行っており、照合までの前処理に約 2 分を要している。実時間化にはまだ程遠いが、エッジ抽出や Hough 変換についてはハードウェア化の試みもあり^{(13),(14)}、1 秒以下での処理が期待できる。またここでの照合は単純で、しきい値などを設ける必要がなく、一致度の判定基準が明確であるため、ハードウェア化も容易と思われる。更に地図がパターンであるため、その作成や変更が比較的簡単なことも利点である。

一方、静止した状態でカメラから得た画像からだけでは周囲の環境を決定づける情報量は十分ではなく、特に相似構造の廊下では、位置を唯一に定めることが困難である。そのため、廊下の分岐や個々の物体の識別、環境情報をより多く得るための方法の検討が必要である。また、本方法で利用した環境復元方法では、廊下内の壁面に凹凸があったり、正面に壁がある場合の復元画像が正しく得られないという問題も残されており、現在これらについて研究を進めている。

謝辞 環境地図の作成にあたり、当講座移動ロボット研究グループの渡並智君、笹尾和仁君らの協力を得た。ここに記して謝意を表します。

表 2 主な処理の平均計算時間

- *1) 計算機は PC9801UV (8MHz)+8087 を使用し、内蔵のクロックで計時した
- *2) 情景画像 26 例の処理時間の平均
- *3) 壁の復元、補間、データ圧縮は 1 面 10m 区間に付き
- *4) 照合は 1 面、地図 10m 区間に付き

処 理	時 間 (秒)
エッジの抽出	39.8
Hough 変換	38.7
無限遠点抽出	4.0
境界線の抽出	1.1
壁の復元	12.0
補間	20.2
データ圧縮	8.7
合 計	125.3
照 合	2.0

参 考 文 献

- (1) 石川, 他: "白線認識を利用した無人搬送車における視覚誘導の一方式", 電子通信学会論文誌(D), J 69-D, 4, pp. 580-590 (昭 61-04).
- (2) H. P. Moravec: "The Stanford Cart and the CMU Rover", Proc. of IEEE, 71, 7, pp. 872-884(1983).
- (3) 八木, 他: "動画像処理を用いた移動ロボットの環境認識", 電子通信学会論文誌(D), J 69-D, 6, pp. 967-974 (昭 61-06).
- (4) 南方, 他: "到達目標の指定と視覚フィードバックを用いた移動", 昭 62 電子情報通信学会全国大会論文集, 1566 (1987),
- (5) M. Yachida, et al.: "Model-guided Monitoring of a Building Environment by a Mobile Robot", Proc. 8th IJCAI, pp. 1125-1127(1983).
- (6) R. L. Madarasz, et al.: "The Design of an Autonomous Vehicle for the Disabled", IEEE J. of Robotics and Automation, RA-2, 3, pp. 117-126(1986).
- (7) 坪内, 池田: "地図を用いた移動ロボット用視覚(1X2)", 第 4 回日本ロボット学会学術講演会論文集, pp. 341-348 (昭 61-12).
- (8) 坪内, 池田: "地図とカラー画像を利用した屋内環境を移動するロボットのための視覚システム構成法", 電子情報通信学会論文誌(D), J70-D, 8, pp. 1685-1688 (昭 62-8).
- (9) 館, 他: "ランドマークとマップを用いる移動機械の誘導法", バイオメカニズム, 5, pp. 208-219(1980).

- (10) 津村：“ビークルオートメーションにおける位置決定方法”，システムと制御，25， 3， pp. 134-141(1981).
- (11) 恩田，他：“移動ロボットのための廊下の3次元環境認識”，電子情報通信学会論文誌(D)，J70-D，12， pp. 2510-2516 (昭62-12).
- (12) 松島，他：“Hough変換の性質を用いた無限遠点の抽出法”，電子情報通信学会論文誌(D)，J70-D，5， pp. 1051-1052 (昭62-10).
- (13) 木内，他：“剰余演算法を用いた動画像処理用FIRデジタルフィルタ”，電子通信学会論文誌(D)，J67-D，4， pp. 536-543 (昭59-4).
- (14) 花原，他：“実時間Hough変換プロセッサ”，昭61電子情報通信学会情報・システム部門全大92 (昭61-3).