日本機械学会論文集(C編)
56巻521号(1990-1)

論文 No. 88-1659 B

モデルベースト画像計測システム*

大山英明*1, 舘 暲*1

Model-Based Image Measurement System

Eimei OYAMA and Susumu TACHI

A model-based image measurement system which defines a model with variables. measures data, defines measurement equations, solves these equations and estimates values of the variables is proposed. The feasibility of the system is verified by an experimental system that estimates the position, attitude, and inner parameters of an object from its image. Using the numerical method, this system can use measures not expressed by mathematical equations.

Key Words: Model-Based Measurement, Measurement Using Image, Nonlinear Least Square Method, Model Definition Using Variables

1. 緒 言

環境モデルの作成と利用は近年のロボティクスの重要な課題として研究されているが[1][2], 現時点では環境モデルを利用した実際のロボットシステムは少ない. 環境モデルの作成が容易でないことがその一因であると考えられる. 環境モデルの作成には従来のCADシステムを利用すればよいと言われている. CADデータの利用によって, 部屋や廊下の形状や, 置かれている物体の形状は得ることはできる. しかし移動できる物体の位置・姿勢は一般にわからない. そのような物体は観測によって位置・姿勢を推定しなければならない. 環境モデルの作成にはCADシステムのみでなく, 計測システムが必要であり, 両者の有機的な結合が必要である.

本論では, 人間にとってわかりやすい形式で, 計測対象のモデルを変数を使って定義し, 観測入力を得て観測方程式を導出し, 変数の値を推定するシステムを提案する. これは問題の定義から観測方程式を導出するCADシステムとその観測方程式を解く計測システムを統合したシステムである. その一例として1枚の画像と物体の形状からモデル, 人間の支援を受けて観測方程式を導き, 物体の位置・姿勢・可動部分の状態等を推定するシステムを試作し, その動作を確認する.

本システムは以下に述べる画像認識システムを画像計測処理の点からとらえなおし, かつ機能を拡張したものである.

フレーム形式の高級言語で物体のモデルを記述し, モデルをもとに画像を認識するシステムとしてはACRONYMがある[3]. ACRONYMは複雑な記号的な処理によって, 物体の位置・姿勢・形状を定義する内部パラメータのとる範囲を限定する機能を持つとされた. しかし画像の観測方程式の非線形性はACRONYMシステムの記号的な推論の能力を越えていたため, その位置・姿勢・内部パラメータ推定能力は高いものではなかった.

LoweはNewton法を利用することによって, 二次元画像から形状モデルを基に位置・姿勢を推定できるシステムを作成した[4]. Loweのシステムも画像認識システムであり, 画像認識の認識結果の検証に位置・姿勢推定法を用いている. Loweは物体の内部パラメータの推定にも, 位置・姿勢推定のための非線形解法を応用できることを述べている.

* 平成元年10月15日 第67期全国大会講演会において講演, 原稿受付 昭程63年12月6日.

*1 正員, 工業技術院機械技術研究所 (〒305 つくば市並木1-2).

本システムは，ACRONYMの形状モデル定義機能と，Loweのシステムの位置・姿勢推定機能を有機的に合わせ持つ画像計測システムであり，以下に述べるような機能の拡張を行っている．

LoweのシステムはACRONYMと異なり，モデルの記述にはあまり重点を置いていないので，推定すべき変数の指示の方法等，計測問題の定義法に問題が残っていた．本論のシステムは人間にとってわかりやすい高級言語で，形状モデルを定義できるようにしている．

Loweのシステムでは，システムがモデル上の頂点と画面上の頂点や直線との対応付けを自動的に行う．しかし一般に対応付けのような画像認識の処理は計算機にとっては不得意な処理であり，時間がかかり，信頼性は低い．本論では人間が対応付けを行うためのインタフェイスを用意し，対応付け処理を人間が行うことによって，より実用的なシステムを構成している．

Loweのシステムでは，観測方程式はモデル上の点と画像上の直線や頂点の対応に限られていた．本システムでは物体の画面上における輪郭線，輪郭線内部の面積，重心位置，輪郭線の長さ等も物体の位置・姿勢・内部パラメータの推定に利用できる．したがって例えば頂点を持たない曲面で構成される物体も，画面上の輪郭線上の点列を観測量として利用することによって，その位置・姿勢推定が可能になる．

また本システムの位置・姿勢推定システムはよりロバストなアルゴリズムを採用している．

2. モデルベースト画像計測システム

2・1 モデルベースト計測システム 一般に計測は，推定すべきパラメータをx，観測値いをy，観測モデルを$h(y, x)$とすると$h(y, x)=0$という観測方程式から，xを推定する処理と考えることができる．従来の計算処理は，一般に人間が観測値yを得る観測システムを作成し，観測モデル$h(y, x)$を導き，観測方程式をなんらかの方法で解いて，xを推定するものであった．しかし観測モデル$h(y, x)$の導出が，容易でない場合も存在する．人間にとって解りやすい形式で，ユーザが問題を記述すると，システムが問題の記述から観測モデルを導出し，さらに観測方程式を解いて，観測値から未知パラメータを推定することができれば非常に有用である．このようなシステムをモデルベースト計測システム(Model Based Measurement System)と呼ぶことにする．

2・2 モデルベースト画像計測システム 物体のモデルの記述をM_o，物体の形状を定義するパラメータx_{in}から，三次元の形状モデルを合成する関数を$m(\ )$，三次元の形状モデルと物体の状態x_{st}から画面座標における画像の記述M_sに変換する関数を$t(\ )$，画像の記述M_sから特徴量Cを抽出する関数を$g_c(\ )$とする．観測量をyとし，観測モデルを$h(y, c)$として，画像の観測方程式はまとめると

$$h(y, g_c(t(x_{st}, m(x_{in}, M_o))))=0$$

となる．

画像による計測は，yからx_{st}やx_{in}を推定する処理である．$m(\ )$，$t(\ )$と言った関数はすべての物体について共通に使える．利用したい観測量がある場合，$g_c(\ )$，$h(\ )$といった関数を作成すれば，形状モデルによらず観測方程式の合成が可能である．あらかじめ十分な種類の観測量について$g_c(\ )$，$h(\ )$を定義しているので，通常は観測量の定義を行う必要はない．ただし現時点では，特徴量Cと観測量yとの対応付けを自動的に行うのは困難であり，本システムでは，人間が対応付けを行うようにしている．しかし本システムでも，人間が物体を画面から切り出せば，孤立図形を入力とする物体の状態推定の問題になり，時間はかかるが，自動的に推定することもできる．

$g_c(\ )$には，頂点の座標値や直線のパラメータ等を出力する関数のほかに，画像の記述から画像を生成して，その画像に処理を施して，孤立図形の重心や位置や面積など特徴量を抽出するといった複雑な処理を行う関数も用意してある．

3. 画像の観測方程式

物体上の点の画面上への変換関数$t(\ )$の具体的な式を示す．世界座標系と視点座標系が一致しているものとする．

(1) 物体固定座標系と視点座標系の関係

$\boldsymbol{x}_b$を物体固定座標系からみた物体上の点，$\boldsymbol{x}=(x, y, z)$を視点座標系における物体上の点の位置，$\boldsymbol{x}_{cg}$を物体の重心位置，C_{bi}を方向余弦行列，(ϕ, θ, ψ)を視点座標系に対する物体固定座標系の姿勢を表すオイラー角とすると

$$\boldsymbol{x}=C_{bi}\cdot\boldsymbol{x}_b+\boldsymbol{x}_{cg}$$

$$C_{bi}=F(\phi, \theta, \psi)$$

である．$F(\phi, \theta, \psi)$を(ϕ, θ, ψ)によって表すと

$$F(\phi,\theta,\psi)=\begin{bmatrix}\cos\phi\cos\psi & \cos\phi\sin\psi+\sin\phi\sin\theta\cos\psi & \sin\phi\sin\psi+\cos\phi\sin\theta\cos\psi\\ \cos\theta\sin\psi & \cos\phi\cos\psi+\sin\phi\sin\theta\sin\psi & \sin\phi\cos\psi+\cos\phi\sin\theta\sin\psi\\ -\sin\theta & \sin\phi\cos\theta & \cos\phi\cos\theta\end{bmatrix}$$

（2） 透視変換（現点座標系と画面座標系の関係）

(X, Y) を (x, y, z) に対応する画面上の点とすると，その関係は以下のとおりである．

$X=k_0 y/x$

$Y=k_1 z/x$

4. システムの特徴

今回試作したモデルベースト画像計測システムには，以下のような特徴がある．

（1） 対象モデルの定義を立体図形記述のための高級言語を用いて行う．

（2） 変数を利用してモデルの定義ができ，可動部分を持つ物体を定義できる．人間や回転椅子のような形状可変な物体を定義し，その変数を推定できる．

（3） 観測方程式の解法に非線形最小二乗法を用い，それに必要なヤコビアンマトリックスの計算に数値解法を用いるので，可観測性があれば，観測量や推定すべき変数の種類によらず推定が可能である．

（4） 非線形最小二乗法の利用により，既知情報と観測量とを統合できる．

今回作成したモデルベースト画像計測システムの構成を図1に示す．本システムは，観測モデル処理システム，モデルデータベース，観測システム，観測方程式解法システム，対人インタフェイスから構成される．観測モデル処理システムは，モデルの定義から観測モデルを導出する．観測方程式の解法システムは，観測方程式を解いて，未知パラメータの値を推定する．システムには Lisp で書いたシステムと，Cで書いたシステムが存在する．Cで書いたものは柔軟性に乏しいが，高速である．

5. 推定の手順

推定の手順を以下に示す．図2はそのフローチャートである．計測の前提として，モデル記述言語で対象のモデルを作成しておくことが必要である．システムで既に用意している観測量以外の観測量を計測に用いる場合には，観測量を画像から得るプログラムとモデルの特徴量と観測量の関係を新たに登録する必要がある．

（1） 画像の入力を行う．

（2） 形状モデルと観測モデルを読み込む．

（3） 形状モデルの変数に適当な値を入れて画面表示を行う．

（4） 画像と形状モデルとの対応関係をユーザが指示することによって，観測方程式を得る．

（5） 観測方程式を解くことによって，推定すべき変数の値を得る．

（6） 変数の推定値をもとに形状モデルを入力画像に重ねて描く．ある誤差範囲内で画像とモデルが重なっていれば，推定成功である．そうでなければ失敗であり，再び推定過程に戻る．この判断はユーザが行う．

（7） 観測値の誤差の情報と観測方程式から，推定誤差を見積もる．

6. モデルの定義と観測モデル導出

6·1 形状モデルの定義　物体の形状定義を行うための簡単なモデラとして，以下のような機能を持つ，図形記述用の高級言語を作成した．

（1） 変数によって形状を決定できるパラメトリック機能

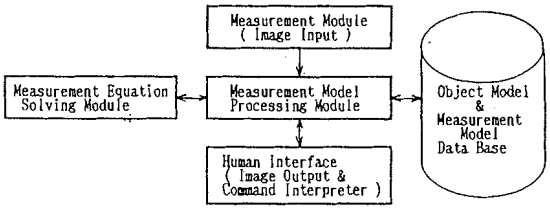

図 1　モデルベースト画像計測システムの構成

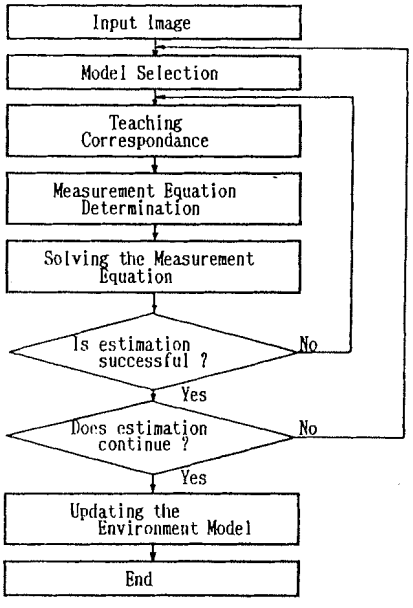

図 2　推定の手順

（2） モデルを階層的に定義する機能

（3） スイープ(sweeping)による形状定義機能

図3に形状モデルを記述する言語の仕様を示す．最もプリミティブなモデルを図4に示す．これは長方形をサーフェスモデルで表現したものである．このようなプリミティブを組合せてモデルの定義を行う．物体の可動部分はrotateやtransといった変換の引き数に変数を与えることで表現できる．図5に机のモデルを示す．

6・2 観測モデル 観測モデルの定義を簡単な言語で簡潔に書くことは難しい．モデルの記述から特徴量を抽出する関数 $g_c(\ )$ および特徴量と観測量の関係 $h(\ )$ の記述にはLispの関数をそのまま用いる．観測モデルを出力する関数の定義は図6のように行う．get-vertex-from-model() という関数は $g_o(\ )$ のひとつである．図5の机の頂点の観測方程式は図7のようになる．代表的な特徴量の抽出関数や特徴量と観測量の関係式は既に定義されており，新たに作成する必要はあまりない．またC言語で書いたシステムでは，あらかじめ定義された観測量しか使用できない．

観測モデルを記述した関数の出力がそのまま観測モデルになっており，変数に値を代入し評価することによって，観測モデルの値を計算できる．C言語で書いたシステムでは，簡単なインタプリタで値の計算を行っている．

```
<model-definition>::=
        (<model-name> <argument-block>
        [<declaration-block>][<constraints-block>]
        <shape-description-list>)

<argument-block>::=()¦(<argument-list>)
<argument-list>::=<variable>
        ¦<variable><argument-list>
        ¦<argument-list> &key (<tag><variable>)
        ¦<argument-list> &optional (<variable> <default-value>)

<declaration-block>::= (declare <declaration-list>)
<declaration-list>::=<declaration>
        ¦<declaration> <declaration-list>
<declaration>::=(<variable> <form>)

<constraints-block>::=(constraints <constraints-list>)
<constraints-list>::=<constraint>
        ¦ <constraint><constraints-list>
<constraint>::=(<constraint-type> <argument-list-c>)
<constraint-type>::=  = ¦ > ¦ < ¦ >= ¦ <= ¦ normal  ¦ uniform

<argument-list-c>::= <form> ¦ <form> <argument-list-c>

<shape-description-list>::= <shape-description>
        ¦<shape-description> <shape-description-list>
<shape-descriprion>::=
        <surface-model>¦<function>¦<model>¦<control-sentence>

<surface-model>::=((vertexes <vertex-list>)
        (lines <line-list>) (planes <plane-list>))

<model>::=(<model-name> <argument-list>)
<function>::=(<functor> <argument-list-f>)
<functor>::= scale ¦ trans ¦ rotate ¦ perse  ¦ or
<argument-list-f>::=<vector>
        ¦<vector><argument-list-f>¦<shape-description-list>
        ¦<shape-descriprion-list> <argument-list-f>

<control-sentence>::=
  (for (<variable> <initial-value> <terminal-value>)
        <shape-descriprion-list>)
```

図 3 モデル記述言語（要旨）

```
(rectangular (x  y)
    (vertexes (1 (x y 0)) (2 ((- x) y 0))
              (3 ((- x) (- y) 0)) (4 (x (- y) 0)))
    (lines (1 (1 2))(2 (2 3))(3 (3 4))(4 (4 1)))
    (planes (1 (1 2 3 4))(2 (1 4 3 2))))
```

図 4 長方形のサーフェスモデル

7. 対 応 付 け

画像から得ることのできる観測量としては，いろいろなものが考えられるが，ここでは最も扱いやすい観測量として，画面上における対象物体の頂点や辺や輪郭線上の点列を利用する．図8に対応付けの時の画面を示す．変数に適当な値を代入して形状モデルを表示し，ユーザが観測量の種類を選択し，画面上の点や直

```
(desk (x y r h &optional (dx 0.03) (dy 0.03))
   (declare (x2 (- (/ x 2) dx)) (y2 (- (/ y 2) dy))
         (array leg ( leg1 leg2 leg3 leg4))
         (array legpos  (( x2 y2 0) ((- x2) y2 0)
                  ((- x2) (- y2) 0) (x2 (- y2) 0))))
   (board (trans (0 0 h) (rectangular x y)))
   (for (i 1 4)
      ((leg i)(trans (legpos i ) (cylinder r h)))))
```

図 5 机の記述

```
(defun vertex-on-model (Att Pos Model Vertex)
    (perse (trans (list ,@Pos) (rotate (list ,@Att)
      ,(get-vertex-from-model Model Vertex) ))) )

(defun vertex-on-image-vertex-on-model
   ( Vim Att Pos Model Vertex)
   `(subvv 2 ,(vertex-on-model Att Pos Model Vertex)
                  (list ,@Vim)))
```

図 6 観測モデルの定義

```
>(vertex-on-model-vertex-on-image  (list 10.0 -10.0)
       '(f t p) '(xg yg zg) '(desk x y r h)
       '(desk (board (vertex 1))))

(subvv 2 (list 10.0 -10.0)
     (perse
        (trans (list xg yg zg)
           (rotate (list f t p)
              (trans (list 0.0 0.0 h)
                 (list x y 0.0))))) )
```

図 7 観測モデル

線といった特徴要素とそれに対応するモデルの特徴要素をポインティングデバイスで指示することで得る．輪郭線は点列で指示する．

8. 観測方程式の解法システム

8·1 観測方程式の解法手順　観測方程式の解法は以下に示すような手順で行われる．図9にフローチャートを示す．

（1） 推定すべき変数，観測方程式，観測誤差，拘束条件等の情報を受け取る．

（2） 拘束条件をその種類に従って，観測方程式に付け加える．

（3） 観測モデルに含まれない変数が存在する場合など，明らかに可観測性がない場合のチェックを行う．

（4） ユーザが定義した解法で解けるかどうかを調べる．ユーザが定義した解法で解ける場合はその解法ルーチンを起動する．

（5） 非線形最小二乗法を起動し推定を行う．可観測性の最終的な判定は，ヤコビアンマトリックスの計算によって行う．

（6） 観測モデルのヤコビアンマトリックスと観測値の誤差の情報から推定精度を見積る．

図 8　対応付け時の画面

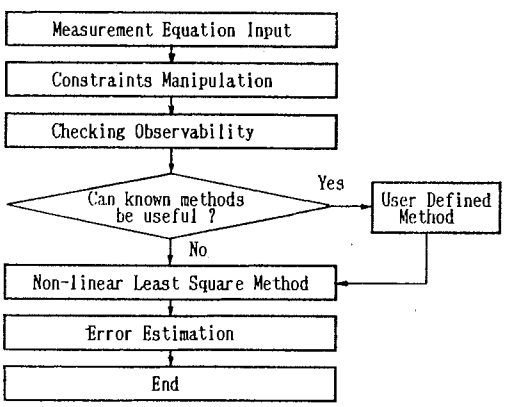

図 9　観測方程式の解法

8·2 非線形最小二乗法　1枚の画像からの計測において，形状既知の物体の場合，内部モデルと画像の3点以上の点について対応が付けば，三次元的な姿勢·位置について複雑な解析解が存在する[5]．しかし多くの場合解析的な解を求めることは困難である．また物体の画面上の面積とか輪郭線の長さ等を観測量とすると解析解を求めることは，ほとんど不可能になる．よって本システムでは非線形最小二乗法を主たる解法としている．

非線形最小二乗法は観測残差二乗和

$$s(x)=\sum_{i=1}^{n} h_i(y,x)^2/\sigma_i^2$$

を最小にする x を求める数値解法である．非線形最小二乗法のアルゴリズムとして Marquardt 法を基にした解法を用いる[6]．ヤコビアン行列，重み行列の定義は以下のとおりである．

$$H_{ij}=\frac{\partial h_i}{\partial x_j}$$

$$W_{ij}=\begin{cases}1/\sigma_i^2\ (i=j)\\ 0\ (i<>j)\end{cases}$$

基礎的な非線形最小二乗法である Gauss-Newton 法は

$$(H^TWH)\Delta x=-H^TWh(y,x)$$

で決定される Δx によって x の値を

$$x(k+1)=x(k)+\Delta x$$

と反復改良する．Marqurdt 法は

$$(H^TWH+\lambda I)\Delta x=-H^TWh(y,x)$$

から計算した Δx を用いる．Marqurdt 法は非線形性の程度によって λ の値を調節する．非線形性が大きい領域では λ を大きくすることによって，収束の安定化を計る．

非線形最小二乗法では，観測モデルのヤコビアンが必要である．現時点ではいくつかの簡単な場合を除いて，ヤコビアンは差分近似によって求めている．差分近似によるヤコビアンの計算は速度は遅いが，精度は解析的な計算法と変わらない[7]．

8·3 推定精度　計測精度は，推定すべき変数，計測の対象，頂点や直線の選び方等によって異なる．個々の場合について，観測方程式と観測ノイズから推定できる．推定すべきパラメータを x，観測値を y，観測方程式を

$$h(y,x)=0$$

とし，H を $h(y,x)$ の x についてのヤコビアン，H_y を $h(y,x)$ の y についてのヤコビアン，観測誤差の共分散行列を $\sum_y$ とすれば，観測方程式の誤差共分散 $\sum_h$ は x が十分に真の解に近ければ

$\Sigma_h = H_y \Sigma_y H_y^T$

パラメータの誤差共分散行列は

$\Sigma_x = (H^T \Sigma_h^{-1} H)^{-1}$

8・4 局所最小解についての対策 非線形方程式の解法の大きな問題として，局所最小解の問題がある．数値解法では，必ずしも真の解に到達できるとは限らない．初期値を複数用意しておくことが必要である．変数の定義域がわかっている場合には，時間はかかるが，乱数によって，次々に初期値を発生させて，調べることができる．

8・5 拘束条件の扱い 拘束条件としては，式の値の確率分布，等式及び不等式拘束条件等がある．式の値の確率分布のうちこのシステムで扱えるのは，正規分布と一様分布だけである．式の値の確率分布が正規分布の場合，観測ベクトルにそれを加える．式の値が一様分布の場合や等式及び不等式拘束条件はペナルティ法で扱う．

9. 実験およびシミュレーション結果

いくつかの例について，推定の結果を示す．本システムを用いて推定を行い，入力画像と形状モデルを重ね合わせて描いたものである．Cで書いたMicro-VAX II上のシステムのものである．

図10に既知形状物体の姿勢・位置推定の例として，机の状態推定の結果を示す．机は床の上にあり，姿勢の1自由度，位置の2自由度を推定する．6頂点を対応付けた．数値計算によって求めた場合，計算時間は4秒であった．姿勢の推定誤差は2.4度であった．位置誤差は未計測である．

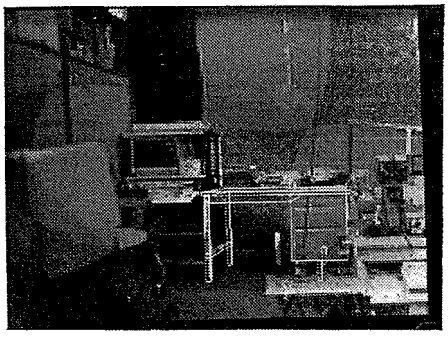

図 10 机の位置・姿勢推定

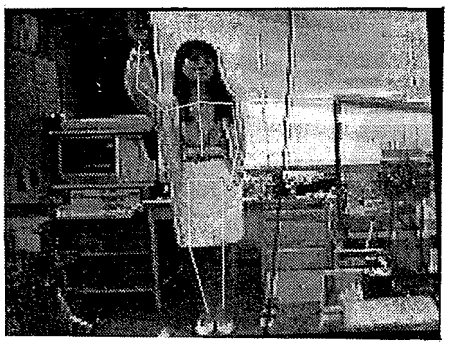

図 11 人間の関節角の推定

図11は関節角を変数とする関節と関節の距離で定義された人間のモデルを用いて，関節の位置を観測量として，関節角を推定し，モデルと画像を重ね合わせたものである．関節の角度を表す19変数と胴体の姿勢・位置の6変数の計25変数を推定した．推定にかかった時間は40秒であった．ただしこの場合，局所最小解となる複数の別解が存在する．1枚の画像から真の解であるかどうかを判別するのは難しい．

図12は，道路の路肩を模擬した曲線（図中の破線）上の点列を観測入力とし，曲線の形状モデル（図中の実線）とを対応付けることにより，視点の二次元平面上の位置の推定を行うシミュレーションである．観測方程式は入力した点列とモデル曲線との最短距離および，画面の縦横方向について極値をとる点の座標や，極値等を用いている．これらの観測量は数式では表現できないので，曲線を画面上に変換した後，観測量の抽出を行うプログラムを観測モデルの計算に用いている．姿勢誤差は0.18度，視線方向の位置誤差は68.0 cm，視線方向に垂直な方向の誤差は，2.9 cmであり，誤差の見積もりは順に1σで，0.03度，86 cm，12 cmであった．一般に画像による計測では，視線方向の誤差は大きい．曲線や曲面上の点を観測量として利用す

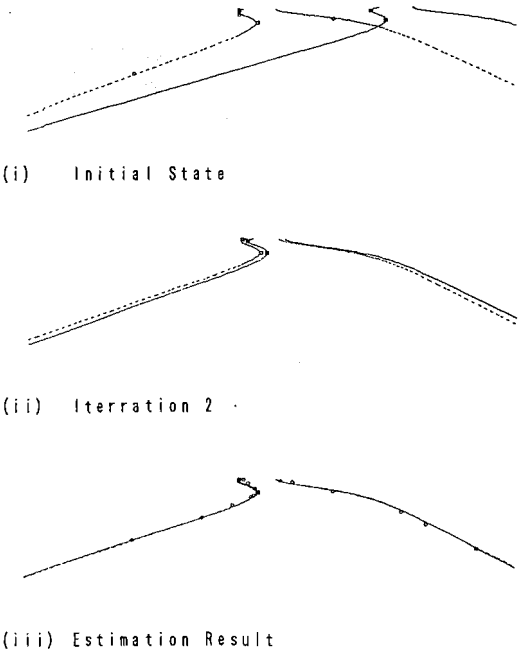

図 12 曲線上の点による位置・姿勢推定

る場合は，ノイズに非常に弱い．また頂点や直線を観測量とする場合と異なり，収束領域が狭く，局所最小解に陥りやすい．

10. 結　　論

10・1 結果　変数を使って対象をモデル化し，観測モデルを導き，観測値と照合することによって，変数の値を推定するシステムを提案・試作し，その動作を確認した．

10・2 問題点　本システムでは，観測方程式の計算はインタプリタとして行っている．そのため計算に時間がかかる．C言語やFORTRANのサブルーチンとして観測方程式を出力するようにしたほうが，計算時間の点では有利になる．しかし柔軟性の点で問題がある．

現時点でシステムは，ヤコビアンマトリックスの計算を差分近似で行っているが，計算時間が大きい．点の位置や直線の式は代数的に計算できるので，観測モデルを数式微分すれば，ヤコビアンを解析的に得ることができ，高速化が可能である．

形状モデルの定義のために，高級言語を使うことは，必ずしも良い選択とは言えない．立体図形を図形のままで，定義できれば，そのほうがユーザには使いやすい．しかしその場合にはかなり大きいマンマシンインタフェイスプログラムが必要となる．現時点では図形による入力インタフェイスはない．

本システムは1枚の画像から，物体の位置・姿勢・可動部分の状態等を推定できるが，実用的な環境モデル作成のためには，距離センサによる計測が必要になるだろう．

10・3 今後の展望　画像認識の性能を上げるために，形状モデルの利用が論じられている．個々の物体について，形状モデルを作成するのは困難である．少ないモデルの定義で，多くの物体対応する方法としては，変数によるモデル定義の導入とモデルの階層化がある．また可動部分を持つ物体のモデル定義には変数の導入が不可欠である．本システムは，画像認識の検証システムとしても利用できる．

文　　献

(1) 白井，モデルベースト・ビジョン，日本ロボット学会誌，**2**-6(1984)，89-94.

(2) 白井・井上，知能ロボットの展望—モデルベーストロボティクス—，日本ロボット学会誌，**5**-6(1987)，462-469.

(3) Brooks, R. A., Symbolic reasoning among 3-D models and 2-D images," *Artificial Intelligence*, **17**(1981), 285-348.

(4) Lowe, D. G., "*Perceptual Organization and Visual Recognition*, (1985), Kulwer Academic Pub.

(5) Fischler, M. A. and Bolles, R. C.," Random sample consensus: A paradigm for model fitting with applications to image analysis and automated car tography, Communications of the ACM, **24**(1981), 381-395.

(6) 中川・小柳，最小二乗法による実験データ解析，(1982)，東京大学出版会.

(7) Hiebert, K. I., "An Evaluation of Mathematical Software That Solves Nonlinear Least Squares Problems, *ACM Transactions on Mathematical Software*, **7**-1(1981), 1-16.