

単一画像による物体の位置計測

山田和幸 金谷健一

群馬大学 工学部 情報工学科

一枚の画像から、三次元形状が既知の物体の三次元的位置と向きを計算するアルゴリズムを提案する。このアルゴリズムは、物体モデルと物体画像との対応点のマッチングを必要としない。物体の三次元の位置と向きは、モデルを基準となる位置から物体像の位置まで移動させる“仮想的な三次元運動”を計算することによって決定する。物体の三次元運動はモデル画像と物体像のそれぞれについて定義した巨視的な“特徴量”の値から反復法によって計算する。また、いろいろな実際の問題を考察し、実際の画像を用いた例を示す。この技法は機械部品の位置ぎめが問題となる製造工程の産業用ロボットなどに広く応用できると思われる。

Determination of Object Position from a Single Image

Kazuyuki Yamada and Ken-ichi Kanatani

Department of Computer Science

Gunma University, Kiryu, Gunma 376, Japan

This paper presents an algorithm for computing, from a single image, the 3D position and orientation of an object whose 3D shape is known. This algorithm does not require "matching" of points of the object model with points of the observed object image. The 3D position and orientation of the object is determined by computing the hypothetical "3D motion" of the object from a reference position of the model image to the position of the observed object image. The motion of the object model is iteratively estimated from the values of "observables" globally defined over the model image and the observed object image. Various technical issues are discussed, and examples based on real images are shown. The technique can be applied to various manufacturing processes by industrial robots where the positioning control of machine parts is not very precise.

1. はじめに

本論文では、製造工程における機械部品のように形が既知で、その三次元形状がデータベースに蓄えられている物体の位置をロボットの視覚によって決定する問題を考える。位置決め精度が不十分な製造環境では、たえず機械部品の位置を補正する必要があるし、また場合によっては、機械部品はいつも同じ位置ではなく、ある範囲にランダムに置かれることもある。このような状況に柔軟に対処するには、ロボットのほうが自律的に環境に適応しなければならない。

物体の位置を決定するには、ある定まった基準位置からカメラに写っている位置までの相対的な「運動」を計算すればよい。そのためには普通、まずデータベースにある物体表面の記述と物体像との「マッチング」を行なう。そして、モデルが物体像と重なるような運動を直接計算する[11]か、あるいは物体像を次々と生成し、食い違い(たとえば生成した特徴点とそれに対応する画像上の点の距離の2乗和)がつかねに減少する位置と向きを探索し、生成したモデル画像と物体像とが十分に重なるまで反復する[9]。

しかし、モデル画像と物体像とをマッチさせることは容易ではない。これを解決する一つの方法は、可能性のある「部分マッチング」を列挙し、それぞれについて運動を計算することである。すると、ほとんどの解は偽の解であろうが、正しい部分マッチングが複数あれば正しい解は重複して現われるので、多数決によって正しい解を選ぶことができる[11]。この手法は「一般Hough変換」とよばれている。それにしても、選んだマッチングが正しいことを確かめなければならない。それには結局、マッチングは正しいものとして物体像を生成して、実際の画像と重なるかどうかを調べることになる。

本論文では、マッチングを用いずに運動を計算する方法を提案する。その基本的原理は、画像間のマッチングの代わりに、生成したモデル画像と実際の物体像のそれぞれについて定義した巨視的な「特微量」の値から三次元運動を計算するというものである。このような特微量を用いる考え方は、初めパターン認識に関連して提案された[1]。この考え方は、平面の位置や運動の決定に応用され[7, 2]、特微量として、テクスチャ[12]や面の輪郭の形状[5]、輪郭線に沿う線積分[6]などが用いられた。しかし、このような方法は「点の対応関係」は必要としないが、「面の対応関係」を必要とする。これは、特微量がある一つの平面領域についてのみ定義されるためである。さらに運動を決定するのに、画像面上の瞬間的な速度を表す「オブティカルフロー」の方程式をもとにしているため、二つの

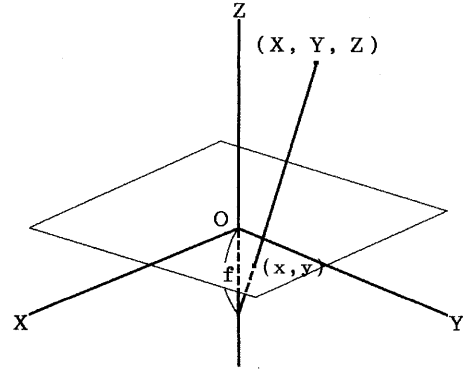


図1. 透視変換

画像間の変位が微小でなければならない。後に反復法の利用[8]や、特微量を定義するのに必ずしも平面領域を必要としない方法が示唆された[3]。本論文ではこの考えを拡張し、まったくマッチングを必要としない運動決定のアルゴリズムを述べる。物体は多面体である必要はなく、曲線や曲面を含んだ任意の形状をもっているてもよい。

2. 速度場と物体の運動

空間中にXYZ座標系をとる。XY平面を画像面とし、点(0,0,-f)を透視変換の視点とする(図1)。図1から次の関係がなりたつ。

$$x = \frac{fX}{f+Z}, \quad y = \frac{fY}{f+Z}. \quad (2.1)$$

物体モデルはその3次元形状が既知であれば、計算によって空間中の任意の位置と向きに配置することができる。3次元剛体運動はある基準点での平行移動(A,B,C)と、その点まわりの回転R=(r_{ij})(直交行列)で表せる。ここで、基準点を(X₀, Y₀, Z₀)とすると、平行移動(A,B,C)と回転Rで指定される運動により、物体上の点(X,Y,Z)は次のような点(X',Y',Z')へ移動する。

$$\begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix} = \begin{bmatrix} X_0 + A \\ Y_0 + B \\ Z_0 + C \end{bmatrix} + \begin{bmatrix} R \end{bmatrix} \begin{bmatrix} X - X_0 \\ Y - Y_0 \\ Z - Z_0 \end{bmatrix}. \quad (2.2)$$

次のような新しい量zを定義する。

$$z = \frac{fZ}{f+Z}. \quad (2.3)$$

また次のようにおく。

$$x_0 = \frac{fX_0}{f+Z_0}, \quad y_0 = \frac{fY_0}{f+Z_0}, \quad z_0 = \frac{fZ_0}{f+Z_0}. \quad (2.4)$$

式(2.2)の運動を、画像座標(x,y)と式(2.3)で定義したzで表すと次のようになる。

$$x' = f \frac{x + (1-z/f)A + (r_{11}-1)\hat{x} + r_{12}\hat{y} + r_{13}\hat{z}}{f + (1-z/f)C + r_{31}\hat{x} + r_{32}\hat{y} + (r_{33}-1)\hat{z}},$$

$$\begin{aligned} y' &= f \frac{y + (1-z/f)B + r_{21}\hat{x} + (r_{22}-1)\hat{y} + r_{23}\hat{z}}{f + (1-z/f)C + r_{31}\hat{x} + r_{32}\hat{y} + (r_{33}-1)\hat{z}}, \\ z' &= f \frac{z + (1-z/f)C + r_{31}\hat{x} + r_{32}\hat{y} + (r_{33}-1)\hat{z}}{f + (1-z/f)C + r_{31}\hat{x} + r_{32}\hat{y} + (r_{33}-1)\hat{z}}. \end{aligned} \quad (2.5)$$

ただし、次のように定義した。

$$\begin{aligned} \hat{x} &= x - \gamma(z)x_0, \quad \hat{y} = y - \gamma(z)y_0, \quad \hat{z} = z - \gamma(z)z_0, \\ \gamma(z) &= \frac{f-z}{f-z_0}. \end{aligned} \quad (2.6)$$

3次元回転は、回転軸 (n_1, n_2, n_3) (単位ベクトル)とそのまわりの右ネジ方向の回転角 Ω で決まる。これに対応する回転の行列 $R=(r_{ij})$ は次のように与えられる。

$$r_{ij} = \delta_{ij} \cos \Omega + n_i n_j (1 - \cos \Omega) - n_{ij} \sin \Omega. \quad (2.7)$$

ただし、 δ_{ij} はKroneckerのデルタであり、 $n_1 = n_{23} = -n_{32}$, $n_2 = n_{31} = -n_{13}$, $n_3 = n_{12} = -n_{21}$ とおいた。回転角 Ω が非常に小さいとき、 $R=I+\Delta R$ と書ける。ただし

$$\Delta R = \begin{bmatrix} 0 & -\Omega_3 & \Omega_2 \\ \Omega_3 & 0 & -\Omega_1 \\ -\Omega_2 & \Omega_1 & 0 \end{bmatrix} + O(\Omega^2), \quad (2.8)$$

であり、 $\Omega_i = \Omega n_i$, $i=1, 2, 3$, とおいた。 $O(\Omega^2)$ は Ω に関して2次以上の項を表す。ここで運動の継続時間を Δt とし、 $(A/\Delta t, B/\Delta t, C/\Delta t)$ と $(\Omega_1/\Delta t, \Omega_2/\Delta t, \Omega_3/\Delta t)$ の極限 $\Delta t \rightarrow 0$ をとると、並進速度 (a, b, c) と回転速度 $(\omega_1, \omega_2, \omega_3)$ が得られる。

式(2.5)を用いて $((x'-x)/\Delta t, (y'-y)/\Delta t)$ の極限 $\Delta t \rightarrow 0$ をとると、画像面上の点 (x, y) の速度 (\dot{x}, \dot{y}) は次の形になる。

$$\begin{aligned} \dot{x} &= u_1(x, y, z)a + u_2(x, y, z)b + u_3(x, y, z)c \\ &\quad + u_4(x, y, z)\omega_1 + u_5(x, y, z)\omega_2 + u_6(x, y, z)\omega_3, \\ \dot{y} &= v_1(x, y, z)a + v_2(x, y, z)b + v_3(x, y, z)c \\ &\quad + v_4(x, y, z)\omega_1 + v_5(x, y, z)\omega_2 + v_6(x, y, z)\omega_3. \end{aligned} \quad (2.9)$$

ただし、次のようにおいた $(\hat{x}, \hat{y}, \hat{z})$ の定義は式(2.6)。

$$\begin{aligned} u_1(x, y, z) &= 1 - \frac{z}{f}, \quad u_2(x, y, z) = 0, \\ u_3(x, y, z) &= -\frac{x}{f} \left(1 - \frac{z}{f}\right), \quad u_4(x, y, z) = \frac{x}{f} \hat{y}, \\ u_5(x, y, z) &= \hat{z} + \frac{x}{f} \hat{x}, \quad u_6(x, y, z) = -\hat{y}, \\ v_1(x, y, z) &= 0, \quad v_2(x, y, z) = 1 - \frac{z}{f}, \\ v_3(x, y, z) &= -\frac{y}{f} \left(1 - \frac{z}{f}\right), \quad v_4(x, y, z) = -\hat{z} - \frac{y}{f} \hat{y}, \\ v_5(x, y, z) &= \frac{y}{f} \hat{x}, \quad v_6(x, y, z) = \hat{x}. \end{aligned} \quad (2.10)$$

3. 特徴量からの運動の計算法

式(2.1)を用いると、計算によって空間中に置いた物体モデルの投影像が生成できる。物体には頂点のようなはっきりと識別できる点があるものとする。そのような点を「特徴点」とよぶことにする。モデル画像の n 個の特徴点の像を (x_i, y_i) , $i=1, \dots, n$ とする。これらの

特徴点の「特徴量」 J を次のように定義する。

$$J = \sum_{i=1}^n w(x_i, y_i). \quad (3.1)$$

ただし $w(x, y)$ は任意に与えた重み関数である。

画像面上に速度場 (\dot{x}, \dot{y}) があるとき、特徴量 J の時間微分 dJ/dt は次のようになる。

$$\frac{dJ}{dt} = \sum_{i=1}^n \left[\frac{\partial w}{\partial x}(x_i, y_i) \dot{x}_i + \frac{\partial w}{\partial y}(x_i, y_i) \dot{y}_i \right]. \quad (3.2)$$

式(2.9)を代入すると次のようになる。

$$\frac{dJ}{dt} = C_1 a + C_2 b + C_3 c + C_4 \omega_1 + C_5 \omega_2 + C_6 \omega_3. \quad (3.3)$$

ただし、次のようにおいた。

$$C_k = \sum_{i=1}^n \left[\frac{\partial w}{\partial x}(x_i, y_i) u_k(x_i, y_i, z_i) + \frac{\partial w}{\partial y}(x_i, y_i) v_k(x_i, y_i, z_i) \right]. \quad (3.4)$$

さて、実際にカメラで撮った画像が与えられたとする。もしこの画像と生成したモデル画像とが近ければ、物体モデルは微小な「運動」を行なったとみなすことができる。したがって、特徴量の時間微分 dJ/dt は二つの画像の特徴量の差分で近似できる。(二つの画像間の経過時間を単位時間とする。)そこで、6個以上の独立な重み関数 $w(x, y)$ を用いると、式(3.3)によって六つの未知数 $a, b, c, \omega_1, \omega_2, \omega_3$ に関する線形連立方程式が得られる。この方程式を解けば運動が計算できる。このとき、二つの画像の間の特徴点同士の対応を知る必要はない。

実際の画像がモデル画像と大きく食い違っていれば反復法を用いる。すなわち、上で述べた方法で運動を推定して、その通りにモデル画像を動かす、この過程をモデル画像が実際の画像に重なるまで反復する(付録参照)。ただし、二つの画像の一致しているかどうかの判定は特徴量の値を比較することで行う。

4. エッジ像にもとづいた特徴量

物体モデルがエッジを持つものとする。モデル画像には m 個のエッジがあるとする。これは線分である必要はなく、任意の形状の曲線でもよい。その画像の特徴量 J を各エッジ L_i に沿った線積分の総和として、次のように定義する。

$$J = \sum_{i=1}^m \int_{L_i} w(x, y) ds. \quad (4.1)$$

ただし、 $w(x, y)$ は任意に与えた重み関数であり、 s は各エッジ像に沿う弧長である。

画像面上に速度場 (\dot{x}, \dot{y}) があるとき、特徴量 J は次のように変化する。

$$\frac{dJ}{dt} = \sum_{i=1}^m \int_{L_i} \left[\frac{\partial w}{\partial x} \dot{x}(s) + \frac{\partial w}{\partial y} \dot{y}(s) \right] ds$$

$$+w(x,y)\left(\frac{d\dot{x}}{ds}l_1(s)+\frac{d\dot{y}}{ds}l_2(s)\right)ds, \quad (4.2)$$

ただし、 $(l_1(s), l_2(s))$ はエッジ上の点 $(x(s), y(s))$ での弧長 s の増加方向への単位接線ベクトルである。式 (2.9) に代入すると次のようになる。

$$\frac{dJ}{dt}=C_1a+C_2b+C_3c+C_4\omega_1+C_5\omega_2+C_6\omega_3, \quad (4.3)$$

ただし、 $C_k, k=1, \dots, 6$, を次のようにおいた。

$$C_k=\sum_{i=1}^m \int_{L_i} \left[\frac{\partial w}{\partial x} u_k(x,y,z) + \frac{\partial w}{\partial y} v_k(x,y,z) \right. \\ \left. +w(x,y)\left(\frac{du_k}{ds}l_1(s)+\frac{dv_k}{ds}l_2(s)\right)\right] ds. \quad (4.4)$$

物体モデルの3次元形状や位置・向きが既知であるので、これらの $C_k, k=1, \dots, 6$, はすべて計算できる (付録参照)。したがって6個以上の独立な重み関数を用いて式 (4.1) の特徴量を計算すれば、前節で述べたようにしてエッジ同士のマッチングをまったく用いないで物体の3次元運動は計算できる。

5. 考察

本論文のアルゴリズムをまとめると次のようになる。最初に、データベースに蓄えてある3次元形状に関するデータを使って、物体のモデル画像を生成する。生成した画像と実際の画像とが一致したら、物体は正しく置かれていることになる。そうでないならば、その位置から画像上の位置までの「運動」を計算することで、物体の実際の位置と向きとが計算できる。とくに決められた位置がなければ、適当な位置にモデル画像を生成して、そこから出発すればよい。計算は物体像に対して巨視的に定義した「特徴量」を通じて行なうので、どの特徴点がどの点に、あるいはどのエッジがどのエッジに対応するかというマッチングを必要としない。

しかし、特徴点やエッジ同士の対応は必要でないが、モデル画像と実際の物体画像は全体として同じ特徴点やエッジをもたなければならない。この点を考慮すると、特徴量としてエッジを使う方が計算誤差の点で有利であろう。たとえばエッジ検出が完全でなくても、検出できなかったエッジや誤って検出したエッジの全長が短かければ、線積分の値に大きく影響しないと期待できるからである。

本方法のひとつの重大な問題は「隠れ」の可能性である。製造工程において位置ぎめの精度が高いとき、基準位置と物体位置とのずれは非常に小さいと期待できる。しかし、物体の向きのずれが大きいと、モデル画像のある部分は実際の画像では隠れてしまうし、モデル画像で隠れていた部分が実際の物体画像で現われることもある。この隠れの問題を解決する一つの方法は、物体の「代表的な見え方」をデータベースに用意して、適当な「類似の尺度」を導入してあてはまるもの

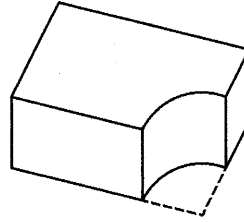


図2. 実験に用いた物体モデル

を見つけることである [4]。適当な基準がなければ、すべての候補を一つ一つ試して反復が収束するかを調べればよい。

本方法の反復の収束の基準は、特徴量の値が収束することである。実際、これは“画像自身のマッチング”を行なう代わりに、“特徴量のマッチング”を行なうことに相当する。しかし、特徴量が収束しても必ずしも画像が収束するとは限らない。したがって、画像の収束を保証する「収束の測度」があって、しかもその計算がマッチングを必要としないものがあることが望ましい。そのようなもので簡単なものは、モデル画像と実際の物体画像の「凸包」の共通部分の面積である。それには、まず特徴点 (またはエッジ) の集合の凸包を求める。これを計算するのに $O(n \log n)$ 回のステップを用いる効率的なアルゴリズムが存在する (n は特徴点の個数 [10])。凸包をモデル画像と実際の画像について別々に求めた後に、二つの凸包の共通部分の面積を求める。これに対しても $O(n+n')$ 回のステップを用いる効率的なアルゴリズムが存在する (n, n' はそれぞれの凸包の頂点の数 [10])。

6. 実験

実験で用いた物体を図2に示す。未知の位置と向きに置いた物体の実際の画像を図3に示す。既知の位置と向きにモデルを表示し、モデル画像が実際の画像と一致するまで反復する過程を図4に示す。ただし、この実験では画像処理は省略して、画像から頂点の座標を直接読みとって入力データとした。第一ステップは“予備的な平行移動”である (付録参照)。同じ物体を前とは異なる位置と向きに置いた別の画像を図5に示す。それに対する反復の過程を図6に示す。特徴量として、図4では物体の頂点に関するものを、図6ではエッジに関するものを用いた。図3, 5と同じ画像で、反復を開始する位置の異なる例を図7, 8に示す。特徴量として、図7では物体の頂点に関するもの、図8ではエッジに関するものを用いた。

図4, 6の例では、自由度が6 (三つの平行移動パラメータと三つの回転パラメータ) である運動を求めてい

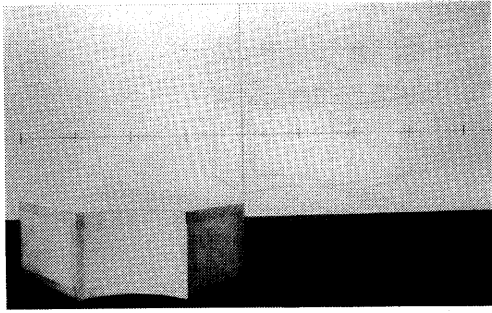


図3. 図2の物体の実際の画像

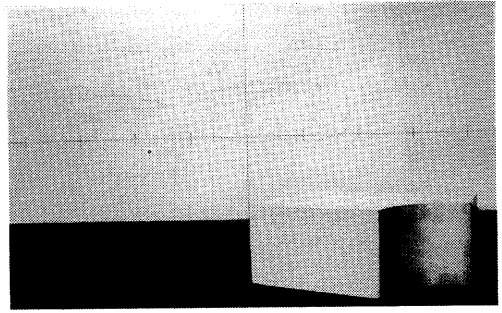


図5. 図3と異なる位置にある物体の実際の画像

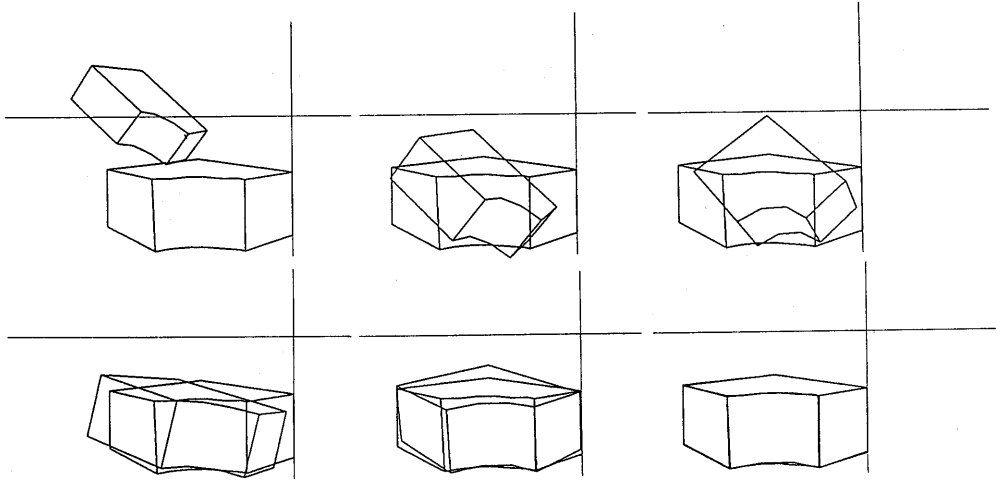


図4. ある位置から図3の物体の位置まで移動する過程

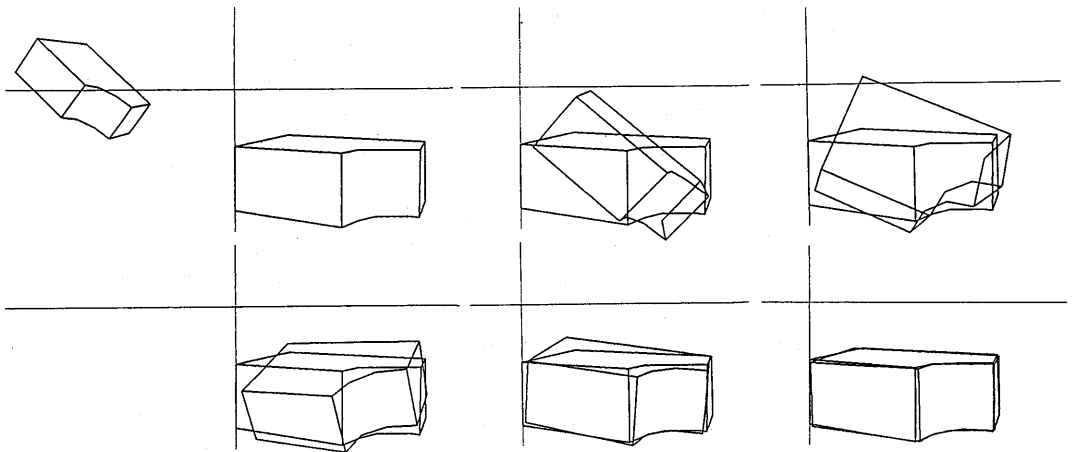


図6. ある位置から図5の物体の位置まで移動する過程

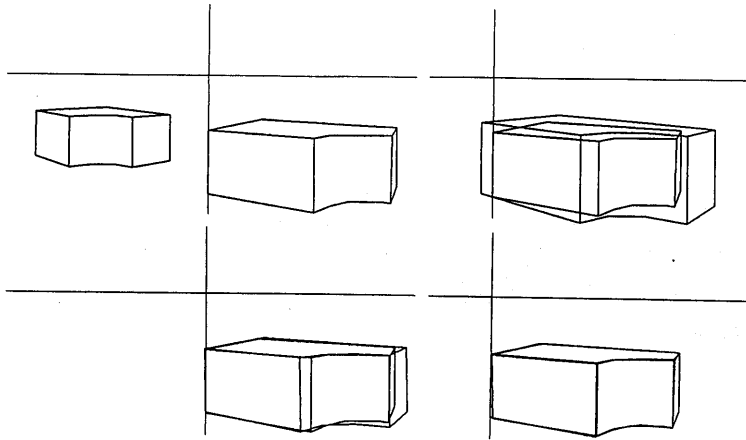


図7. 図4と異なる初期位置から移動する過程

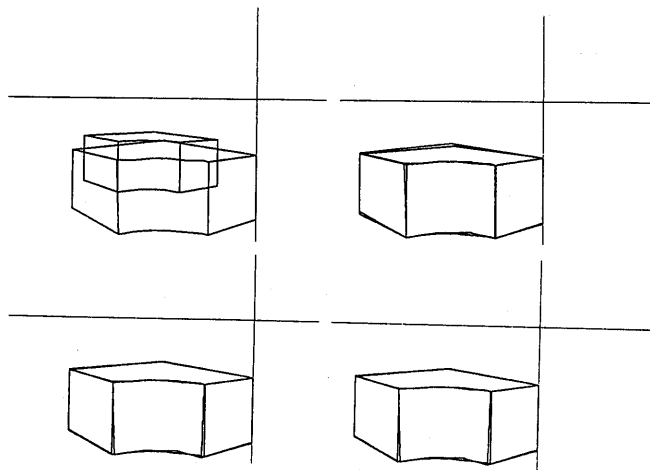


図8. 図6と異なる初期位置から移動する過程

る。これに対して図7, 8の例では, 水平な面の高さが近似的に既知で, その面の上に物体が置かれているという知識を用いて, 出発位置を与えたものである。その結果, ほぼ自由度3(水平面に沿っての平行移動と, 水平面に垂直な軸まわりの回転)のパラメータさえ求めればよい。ただし, この事実は計算過程では使っていない。(すなわち, 残りの三つのパラメータは零と仮定していない)。こうすると, 図7, 8からもわかるように, なめらかに収束する。実際の製造環境では水平な面上に置いた物体のずれが問題となることが多いので, この知識を利用して収束を速めることができる。

7. まとめ

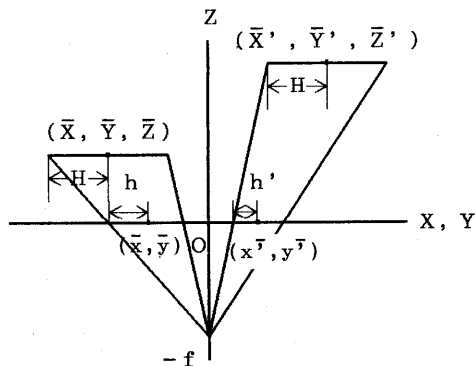
一枚の画像から, 三次元形状が既知の物体の三次元の位置と向きを計算するアルゴリズムを提案した。このアルゴリズムは, 物体のモデルと実際の画像との対

応点のマッチングを必要としない。物体の三次元の位置と向きは, 物体モデルを基準となる位置から物体像の位置まで移動させる“仮想的な三次元運動”を計算することによって決定する。その仮想的な運動はモデル画像と実際の画像のそれぞれの“特徴量”から反復法によって推定する。特徴量の選び方や物体の隠れ, 収束の問題などを考察し, 実際の画像を用いた例を示した。この技法は, 機械部品の位置ぎめが問題となる製造工程の産業用ロボットなどに広く応用できると思われる。

謝辞. 予備実験を行なったダイキン工業の内山昭之氏に感謝する。本研究の一部はカシオ科学振興財団, 矢崎科学技術振興記念財団, 稲盛財団, 文部省科学研究費一般研究C(No. 63550268)の助成金によった。

参考文献

- [1] 甘利俊一, バターン信号の特徴空間における正規化理論, 電気通信学会雑誌, 49-7 (1966), 1342 - 1350.
- [2] S. Amari and M. Maruyama, A theory on the determination of 3D motion and 3D structure from features, *Spatial Vision*, 2 (1987), 151 - 168.
- [3] T.-C. Chou and K. Kanatani, Recovering 3D rigid motions without correspondence, *Proc. IEEE 1st Int. Conf. Computer Vision*, London, June 1987, pp. 534 - 538.
- [4] K. Ikeuchi, Generating an interpretation tree from a CAD model for 3D-object recognition in bin-picking tasks, *Int. J. Comput. Vision*, 1 (1987), 145 - 165.
- [5] K. Kanatani, Tracing planar surface motion from a projection without knowing the correspondence, *Comput. Vision Graphics Image Process.*, 29 (1985), 1 - 12.
- [6] K. Kanatani, Detecting the motion of a planar surface by line and surface correspondence, *Comput. Vision Graphics Image Process.*, 29 (1985), 13 - 22.
- [7] 金谷健一, 対応点を用いない物体の運動認識の理論, 情報処理学会論文誌, 27-3 (1986), 373 - 382.
- [8] K. Kanatani and T.-C. Chou, Tracing finite motions without correspondence, *Proc. IEEE Int. Workshop on Industrial Appl. Machine Vision and Machine Intell.*, Tokyo, February 1987, pp. 118 - 123.
- [9] K. T. Gunnarsson and F. B. Prinz, CAD model-based localization of parts in manufacturing, *Computer*, August 1987, pp. 66 - 74.
- [10] F. P. Preparata and M. I. Shamos, *Computational Geometry*, Springer, New York, 1985.
- [11] T. M. Silberberg, D. Harwood and L. S. Davis, Three dimensional object recognition using oriented model points, *Techniques for 3-D Machine Perception* (ed. A. Rosenfeld), Elsevier, Amsterdam, 1986, pp. 271 - 320.
- [12] 山田・金谷, テクスチャによる面の三次元復元, 情報処理学会コンピュータビジョン研究会 49-5 (1987).



図A. 平面と仮定した物体モデルの平行移動

付録. 反復の手順

1. 予備的な平行移動

生成した物体像の頂点の重心を (\bar{x}, \bar{y}) とし, 実際の画像の頂点の重心を (\bar{x}', \bar{y}') とする. 物体像の「大きさ」を重心から頂点までの最大距離と定義する. 生成したモデルの物体像と実際の物体像の大きさを h, h' とし, 物体モデルの頂点の重心を $(\bar{x}, \bar{y}, \bar{z})$ とする. そして, 「見かけの大きさ」 H を

$$H = (1 + \frac{\bar{z}}{f})h. \tag{A.1}$$

とする. これは, 物体を頂点の重心 $(\bar{x}, \bar{y}, \bar{z})$ を通って画像面と平行な“平面”であるとみなしたときの大きさである (図A). モデルの物体像の大きさと実際の物体像の大きさを同じにし, それらの頂点の重心を一致させるには, 物体モデルの重心 $(\bar{x}, \bar{y}, \bar{z})$ を次の式で定める位置 $(\bar{x}', \bar{y}', \bar{z}')$ に平行移動すればよい.

$$\bar{z}' = (\frac{H}{h'} - 1)f, \bar{x}' = (\frac{\bar{z}'}{f} + 1)\bar{x}, \bar{y}' = (\frac{\bar{z}'}{f} + 1)\bar{y}. \tag{A.2}$$

そこで, 反復法を開始する前に物体モデルを次のように平行移動させることにする.

$$A = \bar{x}' - \bar{x}, B = \bar{y}' - \bar{y}, C = \bar{z}' - \bar{z}. \tag{A.3}$$

2. 重み関数

実験では, 次の重み関数を使用した.

$$\begin{aligned} & 1/h, (x-\bar{x})/h, (y-\bar{y})/h, \\ & (x-\bar{x})^2/h^2, (x-\bar{x})(y-\bar{y})/h^2, (y-\bar{y})^2/h^2, \\ & (x-\bar{x})^3/h^3, (x-\bar{x})^2(y-\bar{y})/h^3, \\ & (x-\bar{x})(y-\bar{y})^2/h^3, (y-\bar{y})^3/h^3. \end{aligned} \tag{A.4}$$

これらに対応して10個の線形方程式が得られる. 未知数は6個であるので, これらを最小二乗法で解いた. 頂点の重心と大きさは新しいモデル画像を生成する度ごとと変わるので, 各ステップで重み関数を計算し直す.

3. 運動の推定

最初に式 (3.3), (4.3) の形の連立線形方程式を解いて速度成分 $a, b, c, \omega_1, \omega_2, \omega_3$ を推定した後, 平行移動を $A=a, B=b, C=c$, 回転を $\Omega=(\omega_1^2+\omega_2^2+\omega_3^2)^{1/2}$, $n_i=\omega_i/\Omega$, $i=1, 2, 3$, とする (経過時間を $\Delta t=1$ とみなしている). 次に式 (2.7) を用いて回転の行列 R を計算する. この推定した運動に基づいて, 新しいモデル画像を生成させ, (x_j, y_j) と z_j を式 (2.1), (2.3) によって計算する. 生成したモデル画像が実際の物体画像と十分に重なるならば, 推定値は正しいとする. (二つの画像が一致しているかの判定は特徴量の値を比較することで行なう.) そうでないならば, 同じことを反復する. すなわち, 生成したモデル画像と実際の画像の特徴量の値から, 再び平行移動 (A', B', C') と回転の行列 R' を推定して, それらを合成した平行移動 ($A+A', B+B', C+C'$) と回転の行列 $R'R$ を計算する. これを収束するまで反復する.

4. 線積分の近似

あるエッジ L について考える. これを N 分割した点 (x_j, y_j) , $J=0, 1, \dots, N$, をとる. ただし, (x_0, y_0) , (x_N, y_N) はエッジ L の端点である. 式 (4.1) の各エッジに沿う線積分は次のように近似できる.

$$\int_L w(x, y) \approx \frac{1}{2} \sum_{J=0}^{N-1} (w_J + w_{J+1}) \Delta s_J. \quad (A.5)$$

ただし次のようにおいた.

$$w_J = w(x_J, y_J), \quad \Delta s_J = ((x_{J+1} - x_J)^2 + (y_{J+1} - y_J)^2)^{1/2}. \quad (A.6)$$

式 (4.4) の係数 C_k を数値的に計算するには物体モデルのみを考えればよい. 物体モデルのエッジ L を N 分割した点 (X_J, Y_J, Z_J) , $J=0, 1, \dots, N$, をとる. ただし, (X_0, Y_0, Z_0) , (X_N, Y_N, Z_N) は端点である. 式 (2.1), (2.3) から画像座標 (x_j, y_j) と z_j が計算できる. 式 (4.3) のそれぞれの項の線積分は次のような有限和で近似できる.

$$\int_L \frac{\partial w}{\partial x} u_k(x, y, z) ds \approx \frac{1}{2} \sum_{J=0}^{N-1} \left(\frac{\partial w}{\partial x} \Big|_{X_J} u_{k,J} + \frac{\partial w}{\partial y} \Big|_{Y_J} v_{k,J+1} \right) \Delta s_J. \quad (A.7)$$

$$\int_L w(x, y) \frac{du_k}{ds} \Big|_{z=1} ds \approx \frac{1}{2} \sum_{J=0}^{N-1} (w_J + w_{J+1}) (u_{k,J+1} - u_{k,J}) \frac{x_{J+1} - x_J}{\Delta s_J}. \quad (A.8)$$

ただし, 次のようにおいた.

$$w_J = w(x_J, y_J), \quad \frac{\partial w}{\partial x} \Big|_{X_J} = \frac{\partial w}{\partial x}(x_J, y_J), \\ u_{k,J} = u_k(x_J, y_J, z_J), \quad \Delta s_J = ((x_{J+1} - x_J)^2 + (y_{J+1} - y_J)^2)^{1/2}. \quad (A.9)$$