

視覚サーボによるカメラ姿勢と 内部パラメータの同時決定法

木下 敬介 出口 光一郎
東京大学 工学部

あらかじめ設定した位置、姿勢(目標位置)で得られるであろう目標画像と、カメラから得られる画像を使ってカメラの運動にフィードバックをかける。これは視覚サーボと呼ばれ、目標位置にカメラを導くことができる。本研究では、これを利用して、従来のカメラキャリブレーションのように、1枚の画像からカメラパラメータを求めるのではなく、カメラを目標位置に移動させながら、目標画像を変化させて、焦点距離、画像中心といった内部パラメータをも同時に決定できることを示す。これは、従来からのカメラキャリブレーションのように1枚の画像からカメラパラメータを決定するのではなく、カメラの運動とカメラパラメータの推定を融合したものである。

Determination of Camera Pose and Intrinsic Parameters by Visual Servoing

Keisuke KINOSHITA and Koichiro DEGUCHI

Faculty of Engineering, University of Tokyo
Hongo 7-3-1, Bunkyo-ku, Tokyo, Japan

We can move the camera to a given goal position and orientation by controlling the velocity computed from the difference between the goal image and the currently obtained images. This is called visual feedback servoing technique.

In this paper, we propose a method to determine the intrinsic camera parameters as well as its pose simultaneously using the visual servoing technique. This method provides a new camera calibration scheme because we estimate the camera parameters by actively moving the camera. Our method is the integration of the camera movement control and the estimation of the camera parameters.

1 はじめに

能動視覚を実現するようなシステムでは、カメラを自由に動かすことができる。カメラは一般にロボットなどに搭載されているが、繰り返し精度は高いものの絶対移動精度は信頼の置けるものではない。このため、カメラ自身とカメラとロボットのキャリブレーションが必要になる。これらは、カメラキャリブレーション、Hand-Eye キャリブレーションなどと呼ばれ従来から研究されてきた。

カメラパラメータは大きく二つ - 外部パラメータと内部パラメータ - に分類できる。外部パラメータはカメラの位置、姿勢など我々が外部から変化させることができる量であり、内部パラメータは焦点距離、画像中心 (光軸と画像面との交点の位置)、レンズの収差など一般的に光学的な性質を表す量である。ところが、現実には両者とも定規で測るように知ることはできず、画像という観測値を使って間接的に推定するしかない。

本論文では、従来のように1枚の画像からその場所でのカメラパラメータを求めるのではなく、あらかじめ設定した位置 (目標位置) までカメラを移動させて外部パラメータと内部パラメータを同時に決定する。

カメラの移動には、視覚サーボ (Visual Servoing) の手法を使う。視覚サーボとは、あらかじめ画像上に設定した目標画像とカメラから得られた画像が一致するようにカメラの運動をフィードバック制御するものである。フィードバック量は画像の情報から計算することができる。これは [1] や [2] などに述べられている。対象をどこから見たいかという目標位置と、そこで得られるであろう目標画像を与えると、カメラを視覚サーボの手法を使って目標位置まで移動させることができる。つまり、移動後の外部パラメータを目標位置と等しいものにすることができる。

さらに、従来の研究では目標画像は一度与えたら収束するまで変化させなかったが、本論文では、この目標画像を内部パラメータの推定値に従って刻々変化させる。そして、この変化する目標画像と、カメラから得られる画像を一致させることで、外部パラメータは目標位置と等しくなり、内部パラメータはその推定値として同定することができる。

2 視覚サーボ

視覚サーボについて概説する。視覚サーボはロボティクスなどの分野で研究されてきた。ロボットにカメラを取り付け、画像から得られる情報をカメラの運動にフィードバックさせるものである。図1のように3次元空間に形状のわかっている対象 M が存在し、この対象に対して目標位置 B (これは、与える) にカメラを移動させることを考える。 M は既知であるので、位置 B で M がどのよう

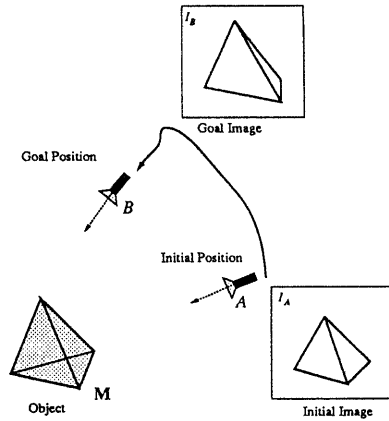


図1: 視覚サーボの図。カメラは初期位置から目標位置まで画像から情報のフィードバックで移動する。

に画像に投影されるかは容易に計算することができる。この画像 I_B を目標画像と呼ぶことにする。カメラは初期位置 A (これは未知) にあり M を撮影している。この画像 I_A からどのようにカメラを動かしたら I_B が得られるかを求め、それをカメラの運動にフィードバックする。カメラから得られる画像が I_B と一致したならば、カメラは B の位置に移動したということになる。問題は、どのようにカメラの運動にフィードバックを与えるかである。これを決定するのが次に説明する相互行列である。

2.1 プリミティブと相互行列

[1] では対象は点、直線、球などで構成されていると考える。このような点や直線、球などをプリミティブと呼び、カメラ座標系で

$$h(\mathbf{x}, \mathbf{p}) = 0$$

とあらわす。 \mathbf{x} はプリミティブ上の点、 \mathbf{p} はこのプリミティブの configuration と呼ばれるものである。 h はプリミティブの種類 (点、直線、etc.) をあらわし、 \mathbf{p} は実際のパラメータを表す。例えば、球面上の点 $\mathbf{x} = (x, y, z)^t$ は、

$$(x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2 - r^2 = 0$$

を満たす。このとき、 $\mathbf{p} = (x_0, y_0, z_0, r)^t$ である。この3次元空間のプリミティブをカメラで撮影すると、画像に2次元のプリミティブ

$$g(\mathbf{X}, \mathbf{P}) = 0$$

が得られる。 \mathbf{X} は画像のプリミティブ上の点、 \mathbf{P} はその configuration である。点は画像に点として投影され、直線は直線に、球は楕円に投影される。

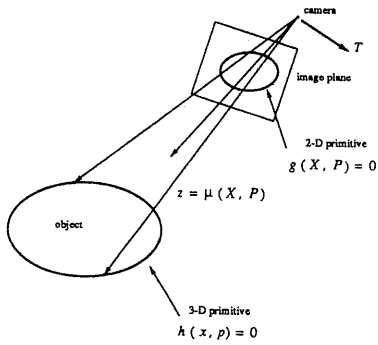


図2: プリミティブと configuration. 3次元空間のプリミティブ $h(x, p) = 0$ は $g(X, P) = 0$ に投影される。

このように、3次元空間のプリミティブが画像上のプリミティブに投影されているとき、画像上の configuration \mathbf{P} とカメラの運動 \mathbf{T} (カメラの平行運動と回転運動) の間には次の関係が成り立つことが知られている [1].

$$\dot{\mathbf{P}} = L^T(\mathbf{P}, \mathbf{p}) \mathbf{T} \quad (1)$$

この L^T のことを相互行列 (Interaction Matrix) と呼ぶ。 L^T は \mathbf{P} の \mathbf{T} に対するヤコビアンに相当している。ここで注意したいのが、 L^T は \mathbf{P} と \mathbf{p} に依存するという点である。 \mathbf{P} は画像上の configuration なので観測できるが、 \mathbf{p} はカメラ座標系での3次元プリミティブの configuration であり、測定することはできない。

さまざまなプリミティブに対する相互行列を具体的に求める方法も [1] に得られている。さらに、 L^T がフルランクでない時の扱いも述べられているが、本研究では、点プリミティブの L^T を考え、さらにフルランクである場合のみを考えることにする。

点プリミティブの相互行列は、オプティカルフローの導出式と一致し、

$$L_{of}^T = \begin{pmatrix} -\frac{f}{z} & 0 & \frac{x}{z} & \frac{xy}{f} & -(f + \frac{x^2}{f}) & y \\ 0 & -\frac{f}{z} & \frac{y}{z} & f + \frac{y^2}{f} & -\frac{xy}{f} & -x \end{pmatrix} \quad (2)$$

である。ただし、 $\mathbf{p} = (x, y, z)^T$ は3次元空間の点、 $\mathbf{P} = (X, Y)^T$ は画像への投影点である。つまり、

$$\begin{pmatrix} \dot{X} \\ \dot{Y} \end{pmatrix} = L_{of}^T \mathbf{T}$$

が成立している。

2.2 フィードバック

ここでの視覚サーボの目標は、(1)式という非線形なシステムに対して、目標位置での目標画像

(configuration で表現し) \mathbf{P}_d に画像から得られている \mathbf{P} を一致させるような \mathbf{T} を求めることである。しかし、先に述べたように、カメラが移動すると、それにつれて \mathbf{P}, \mathbf{p} が変化するので L^T が変化する。これは非線形なシステムであり解析的に一意に \mathbf{T} を求めることは容易ではない。

そこで、 $\mathbf{P} \rightarrow \mathbf{P}_d$ という目的のために、フィードバックをかける。一般に対象は複数 (M 個とする) のプリミティブで構成されるので、各プリミティブおよび目標画像の configuration を縦に並べて、

$$\mathbf{P} = \begin{bmatrix} \mathbf{P}_1 \\ \vdots \\ \mathbf{P}_M \end{bmatrix}, \mathbf{P}_d = \begin{bmatrix} \mathbf{P}_{1d} \\ \vdots \\ \mathbf{P}_{Md} \end{bmatrix} \quad (3)$$

のように定義する。また、これに相当する相互行列として、

$$L^T = \begin{bmatrix} L_1^T \\ \vdots \\ L_M^T \end{bmatrix} \quad (4)$$

を考える。点プリミティブの場合、 L^T は $2M \times 6$ となる。

そして、 $\mathbf{e} = \mathbf{P} - \mathbf{P}_d$ が指数関数的に減少するようにフィードバックをかけることにする [3]。つまり、

$$\dot{\mathbf{e}} = -\lambda \mathbf{e} \quad (\lambda > 0) \quad (5)$$

を満たすようにする。(5)式は、

$$\begin{aligned} -\lambda \mathbf{e} &= \dot{\mathbf{e}} \\ &= \dot{\mathbf{P}} - \dot{\mathbf{P}}_d \\ &= \dot{\mathbf{P}} \\ &= L^T \mathbf{T} \end{aligned} \quad (6)$$

と変形できる。(6)式より、(5)式を満たすようなカメラの運動 \mathbf{T} の例として、

$$\mathbf{T} = -\lambda L^{T+} \mathbf{e} \quad (7)$$

が挙げられる。 L^{T+} は L^T の疑似逆行列である。ところが、 L^T や L^{T+} はカメラが運動すると変化し、これらの正確な値を求めることはできない。さらに、フィードバックをかけるごとに、いちいち L^{T+} を計算し直すのは大変である。そこで、 L^{T+} の近似値 L_d^{T+} を用いることにする。ただし、

$$L_d^T = L^T(\mathbf{P}_d, \mathbf{p}_d) \quad (8)$$

である。これは、目標位置での相互行列である。(7)式で、 L_d^{T+} を用いてフィードバックをかけると、 \mathbf{e} は(6)式より、

$$\dot{\mathbf{e}} = -\lambda L_d^T L_d^{T+} \mathbf{e} \quad (9)$$

のように振舞う。カメラの運動経路で $L^T L_d^{T+}$ が常に正定であるならば、 e は 0 に収束する。通常は、目標位置のまわりで $L^T L_d^{T+}$ は正定であると考えられるので、 e は近似的に指数関数的に 0 に収束するといえる。

ここで注目したいのは、このフィードバックをかける際に、対象の 3 次元形状 (正確には configuration) p を復元していないということである。もちろん、目標位置での相互行列を導き出す時は既知の対象 M の 3 次元形状と目標位置という 3 次元情報を使用するが、その後は e という画像から得られる量しか用いていない。これは、大きなメリットとなる。

3 カメラパラメータの同定

この視覚サーボの手法をカメラパラメータの決定に応用する。本研究ではカメラを自由に動かすことができる場合を想定している。このような場合、ある時点でのカメラの位置、姿勢を求めるのではなく、対象に対する相対的な位置関係を決定できれば良い。つまり、もし、位置がわからないのであれば、対象との相対的な位置のわかっている場所にカメラを移動させれば良いという考えにたつ。もちろん、1 枚の画像からカメラの位置や姿勢、焦点距離などを推定する方法を否定するものではない。ただ、カメラを自由に制御できるとい状況では、従来のカメラキャリブレーションの手法以外にも対象との相対的な位置関係を知る方法があるということを主張したい。

前節の議論から、目標画像にカメラから得られる画像を一致させることで、カメラの外部パラメータに関しては決定できることがわかる。本節では、このカメラを目標位置に運動させながら、同時にカメラの内部パラメータ (焦点距離、画像中心、画素サイズ) を同定できることを示す。

本節でもプリミティブとして点を用いて説明をするが、他のプリミティブを用いても同様の議論をすることができる。画像上の点プリミティブの configuration P は X と一致するので、以降、点プリミティブの configuration として X を用いる。

3.1 焦点距離

焦点距離がわからない場合、先の手法を適応する時に目標値 X_d を決定することができないという問題が生じる。焦点距離が変化すると、目標値も画像中心を中心として放射状に変化する (図 3)。焦点距離 f が真の値から微小量ずれている場合、 f の変化に対する目標値 X_d の変化 \dot{X}_d は、

$$\begin{aligned} \dot{X}_d &= \frac{\partial X_d}{\partial f} \dot{f} \\ &= [X_d/f \quad Y_d/f]^T \dot{f} \end{aligned} \quad (10)$$

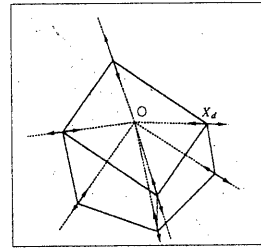


図 3: 焦点距離 f の変化による目標値 X_d の変化

とあらわせる。 $e = X - X_d$ とすると、

$$\begin{aligned} \dot{e} &= \dot{X} - \dot{X}_d \\ &= L_{of}^T T - \frac{\partial X_d}{\partial f} \dot{f} \end{aligned} \quad (11)$$

が得られる。複数の点を用いるので、(3) 式、(4) 式のように L_{of}^T は、 $2M \times 6$ の行列になる。 $\frac{\partial X_d}{\partial f}$ も各点について 2×1 の大きさがあるので、縦に並べて、 $2M \times 1$ の大きさになる。カメラの運動パラメータ T と f の変化率を一つのベクトル U として表現する。

$$U = \begin{bmatrix} T \\ \dot{f} \end{bmatrix} \quad (12)$$

さらに、先の相互行列に相当するものとして、

$$M^T = [L_{of}^T \quad -\frac{\partial X_d}{\partial f}] \quad (13)$$

を定義する。これは、 $2M \times 7$ の大きさになる。すると、(11) 式は、

$$\dot{e} = M^T U \quad (14)$$

とあらわすことができる。あとは (7) 式と同様に

$$U = -\lambda M_d^{T+} e \quad (15)$$

とフィードバックすればよい。ただし、 M_d^T は、焦点距離の初期値を使って求めた目標位置での M^T である。このフィードバックは、(15) 式で得られる T に基づいてカメラを運動させ、さらに、 \dot{f} に基づいて焦点距離の推定値 f を変化させて、この f を使って目標値 X_d を作り直すものである。これは、カメラを運動させると同時に、カメラのモデル (焦点距離) をも変化させていることになる。モデルの変化により目標画像も変化するが、その変化の仕方は対象の形状 (点の位置) に拘束されている。それを表しているのが $\frac{\partial X_d}{\partial f}$ であるといえる。十分時間が経つと、目標画像とカメラから得られる画像が一致し、外部パラメータは目標位置に、焦点距離は f で推定できることになる。

3.2 画像中心と画素サイズ

焦点距離と同様に、画像中心や画素サイズさえも同定できることを示す。

画像中心（光軸との交点）と画像の座標系の原点は一致するとは限らない。画像の座標系の原点を $(0, 0)^T$ とし、真の画像中心をこの座標系で $(U_0, V_0)^T$ であるとする。X や X_d は真の画像中心を原点とした座標系で表したものであるので、 X_d は、画像の座標系では、 $X_d + (U_0, V_0)^T$ と表せる。

また、画像の1画素の大きさもわからない場合が多い。一般に、CCD素子から得られる信号はいったんビデオ信号に変換され、カメラから出力される。そして、画像入力装置によって再び量子化され、画像データとして計算機内のメモリに取り込まれる。このとき、画像データの垂直方向(Y軸)の1画素の大きさは、CCD素子の垂直方向の大きさと一致し、これはCCD素子の仕様書に記載されている。しかし、水平方向(X軸)の1画素の大きさは、カメラのビデオ信号への変換と、画像入力装置のビデオ信号から画像データへの変換の双方のタイミングに依存し、CCD素子の水平方向の大きさと同じであるとはいえない。

ここでは、1画素の形状が正方形でない場合、画像上の座標はそのままに、X軸方向、Y軸方向の焦点距離が異なると考え、それぞれ f_x, f_y とする。画像データの画素サイズが $w_x \times w_y$ であるとする。 w_y はCCD素子の仕様書から求めることができ、Y軸方向を基準に考えると、実際のレンズの焦点距離は f_y であり、 w_x は、

$$w_x = (f_x/f_y)w_y \quad (16)$$

として求めることができる。つまり、 f_x, f_y を求めれば画素サイズを得ることができる。

このように、画像中心が未知で、X軸、Y軸の各方向で焦点距離が異なる場合は、焦点距離を求めた時と同様に、

$$V = \begin{bmatrix} T \\ f_x \\ f_y \\ U_0 \\ V_0 \end{bmatrix} \quad (17)$$

というベクトルと、

$$N^T = [L_{of}^T \quad -\frac{\partial X_d}{\partial f_x} \quad -\frac{\partial X_d}{\partial f_y} \quad -\frac{\partial X_d}{\partial U_0} \quad -\frac{\partial X_d}{\partial V_0}] \quad (18)$$

という相互行列を考える。ただし、

$$\frac{\partial X_d}{\partial U_0} = [1 \ 0]^T, \quad \frac{\partial X_d}{\partial V_0} = [0 \ 1]^T$$

$$\frac{\partial X_d}{\partial f_x} = [X_d/f_x \ 0]^T, \quad \frac{\partial X_d}{\partial f_y} = [0 \ X_d/f_y]^T$$

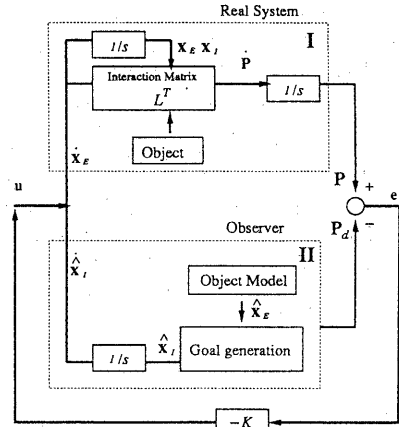


図4: システムのブロック線図

である。 $e = X - X_d$ を0に収束させるために、

$$V = -\lambda N_d^T e \quad (19)$$

とフィードバックする。目標値 X_d は、推定された f_x, f_y, U_0, V_0 に従って更新する。すると、最終的には f_x, f_y, U_0, V_0 の推定値が得られ、(16)式から1画素の大きさを決定することができ、画像中心の座標も得ることができる。

4 システムの記述

内部パラメータの同定も同時に行なうようなシステムは図4のようにオブザーバを構成していると考えられる。Iの部分が実際の制御対象であり、IIの部分がオブザーバである。状態量 x として、実際のカメラの外部パラメータ x_E と内部パラメータ x_I を考える。

$$x = [x_E \ x_I]$$

x_I は焦点距離や画像中心などでありカメラが運動しても変化しない。それに対し、 x_E はカメラの位置、姿勢をあらわし、カメラが運動することによって変化する。さらに、 x の推定値として、

$$\hat{x} = [\hat{x}_E \ \hat{x}_I]$$

を考える。 \hat{x}_I は内部パラメータの推定値であり、 \hat{x}_E は目標位置とする。 u はこのシステムでの制御量であり、 $u = [\hat{x}_E \ \hat{x}_I]$ である。

Iは、 u という入力のうち \hat{x}_E に対応してカメラが運動する。それにより x_E が変化し、 x_E, x_I に基づいて、つまり、実際の相互行列に基づいて、対象がどのように見えるかが決定されている。そして、出力として、画像上のプリミティブの configuration P が得られる。

II はオブザーバである。こちらは、 \mathbf{u} という入力のうち、 $\hat{\mathbf{x}}_I$ に基づいて、内部パラメータの推定値 $\hat{\mathbf{x}}_I$ を更新する。この $\hat{\mathbf{x}}_I$ と、目標位置 $\hat{\mathbf{x}}_E$ 、そして対象のモデルから、目標位置での画像上でのプリミティブの configuration \mathbf{P}_d を出力する。

画像上の観測値 \mathbf{P} と画像上での目標値 \mathbf{P}_d との差 \mathbf{e} を使い、フィードバックゲイン $-K$ を乗じた値を \mathbf{u} にフィードバックする。指数関数的に減少させるときは $K = \lambda M_d^{T+}$ あるいは $K = \lambda N_d^{T+}$ とすればよいことは既に述べた。

このシステムにフィードバックをかけることで、 $\mathbf{e} = 0$ が得られたならば、

$$\mathbf{x}_E = \hat{\mathbf{x}}_E, \quad \mathbf{x}_I = \hat{\mathbf{x}}_I$$

が成立していることになる。つまり、 \mathbf{x}_E は目標位置 $\hat{\mathbf{x}}_E$ と一致し、 \mathbf{x}_I は $\hat{\mathbf{x}}_I$ で推定できていることになる。もしズームレンズのように、焦点距離を外部から任意に制御できる場合、焦点距離は外部パラメータ \mathbf{x}_E として扱うことができる。外部から自由に变化させ、制御できるパラメータを \mathbf{x}_E に、推定しようとするパラメータを \mathbf{x}_I に設定すれば、同様の手法で、両者の値を同時に同定することができる。

従来のカメラキャリブレーションとの比較をすると、例えば、1枚の画像から反復法によってカメラパラメータを求めるものは、画像 \mathbf{P} は変化せず、すべてのパラメータを \mathbf{x}_I として扱って、 \mathbf{P}_d を \mathbf{P} に一致させるように \mathbf{x}_I を変化させていることになるであろう。

5 Hand-Eye キャリブレーション

現実にこの手法を実現しようとする、問題になるのは、ロボットの制御点の座標系とカメラ座標系が一致している保証がないという点であろう。もし、ロボットの制御点の座標系とカメラ座標系が一致していないと、望むカメラ運動をさせることができない。この両座標系間の関係を求めることをロボティクスでは Hand-Eye キャリブレーションと呼ぶ。しかし、次に述べるように、この Hand-Eye キャリブレーションは本手法を利用している限り必要はない。フィードバックをかけることで両座標系の不一致は無効化することができる。

一般に、ロボットの制御点の座標系とカメラの座標系はそれほど大きくずれているとは考えられない。すると、ロボットとカメラの座標系の関係は微小な変位と回転 $\Delta \mathbf{r} = (\Delta r_x, \Delta r_y, \Delta r_z, \Delta \omega_x, \Delta \omega_y, \Delta \omega_z)$ であらわせる (図 5)。ロボット座標系で $\dot{\mathbf{T}} = (t, \Omega)$ の速度で移動させると、カメラ座標系での速度 \mathbf{T} は、

$$\mathbf{T} = A \dot{\mathbf{T}} \quad (20)$$

とあらわすことができる。ただし、

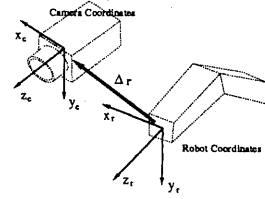


図 5: ロボットの制御点とカメラの座標系

$$A = \begin{pmatrix} R(\Delta \omega) & O \\ O & I_3 \end{pmatrix}$$

$$R(\Delta \omega) = \begin{pmatrix} 1 & \Delta \omega_z & -\Delta \omega_y \\ -\Delta \omega_z & 1 & \Delta \omega_x \\ \Delta \omega_y & \Delta \omega_x & 1 \end{pmatrix}$$

である。

このとき、

$$\dot{\mathbf{T}} = -\lambda L_d^{T+} \mathbf{e} \quad (21)$$

とフィードバックすると、(20) 式より実際のカメラ座標系では、

$$\mathbf{T} = -\lambda A^{-1} L_d^{T+} \mathbf{e} \quad (22)$$

という運動になる。 $\dot{\mathbf{e}} = L^T \mathbf{T}$ であるから、

$$\dot{\mathbf{e}} = -\lambda L^T A^{-1} L^{T+} \mathbf{e} \quad (23)$$

を得ることができる。 $L^T A^{-1} L^{T+}$ が常に正定である場合には \mathbf{e} は 0 に収束する。これは、Hand-Eye キャリブレーションをしなくても、つまり、 $\Delta \mathbf{r}$ を知ることなしに、目標値に収束させることができることを物語っている。同様に、内部パラメータが未知の場合も \mathbf{e} が 0 に収束することがいえる。

カメラを運動させる場合に、従来ではあらかじめ Hand-Eye キャリブレーションが必要であったが、視覚サーボ、つまり、画像情報をロボットの運動にフィードバックさせることで、このキャリブレーションが不要になった。位置や姿勢を求め、制御したいのはカメラの座標系であるので、 $\Delta \mathbf{r}$ を知る必要がないというのは利点となる。

6 実験

本手法の有効性を検証するためにシミュレーション実験と実画像による実験を行なった。

6.1 シミュレーション実験

焦点距離、画像中心、画素サイズが未知であり、カメラの座標系とロボットの座標系が一致しない場合でも、これらの内部パラメータを同定できる

ことを検証する。対象として図6のような1辺120の立方体の頂点を用い、目標位置は対象から800離れたところとする。画像中心の初期値は $(0, 0)^t$ とするが、真の画像中心は画像データの座標系で $(10, 10)^t$ である。また、Hand-Eyeの座標間の関係は、 $\Delta r = (10, 10, 10, 0.1, 0.1, 0.1)$ とした。焦点距離は $f_x = 18, f_y = 16$ とした。誤差とし

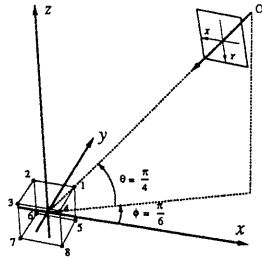


図6: シミュレーション実験での対象(立方体の頂点)と目標位置

ては、量子化誤差、マッチング誤差の両方を考慮して、分散 σ^2 ($\sigma = 0.5$ 画素) のガウス分布に従うノイズを付加した。収束条件はすべての画素で $\|P - P_d\|_2 < 1$ 画素 が成立することとした。また $\lambda = 0.1$ とした。このような設定で初期値(カメラの位置、姿勢、内部パラメータ)を3通りに変化させてシミュレーションを行なった。

図7に初期値2の場合の目標画像 P_d と実際の画像 P の軌跡を示す。それぞれ最終的には同一の画素に収束しているのがわかる。図8に焦点距離の推定値を示す。どの初期値からでも $f_x = 18, f_y = 16$ に収束している。ところが、図9のように画像中心はあまり精度良く求まらなかった。もちろん、誤差がない場合は正確な値に収束する。逆にいうと、画像中心の座標は、カメラの位置、姿勢、焦点距離の決定にあまり影響しないということでも

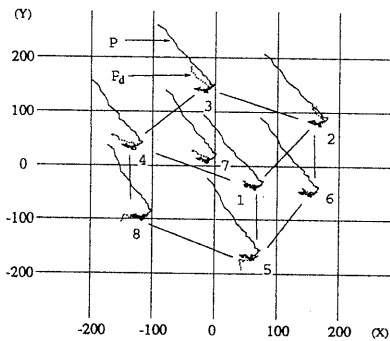


図7: 目標画像 P_d と実際の画像 P 。対象の輪郭は書き加えた。

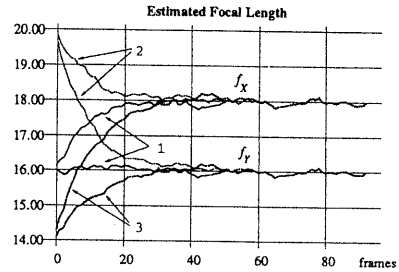


図8: 各初期値での推定焦点距離。それぞれ、18,16に収束している。

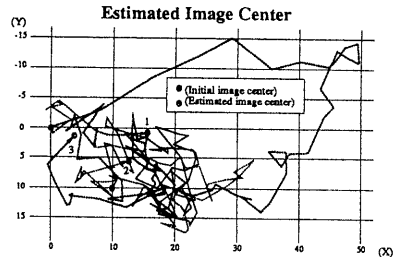


図9: 各初期値での画像中心の推定値

ある。

6.2 実画像による実験

立方体の対象に対して実画像による実験を行なった。初期位置での画像を図10に示す。対象の各頂点の目標値は□で示される。図11に収束地点での画像を示す。また、その時の推定された各軸の焦点距離を図12に示す。これより、この画像はほぼ正方形の画素を持っていることがわかる。また、画像中心はシミュレーションでもわかるとおり、それほど信頼おける値を示さなかった。

7 考察

このフィードバックでの収束条件を検討してみた。一般に画像への誤差として、量子化誤差やマッチング誤差の影響としてガウス性のノイズを想定している場合が多い。本研究のシミュレーション実験でもこのようなノイズモデルを採用した。しかし、実際の画像ではこのような解析が容易なノイズ分布であるとは限らない。確かに、誤差がある程度大きい場合はガウス性のノイズで近似できるかも知れないが、ノイズが1, 2画素より小さい場合(本手法での収束寸前のような状況)では先のような仮定は成立しない。カメラパラメータを最適に推定しようとするれば、画像上の誤差 ee^T のような量を最小にしないといけない。しかし、現

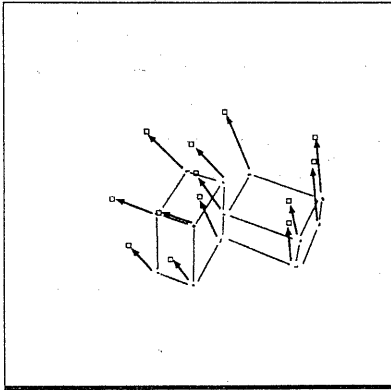


図 10: 初期位置での画像 (2 値化し、対象の輪郭は書き入れた)。□は目標画像 P_d である。

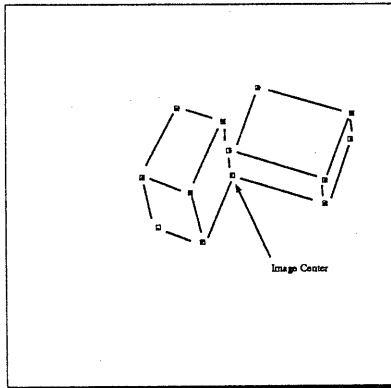


図 11: 収束位置での画像 (2 値化し、対象の輪郭は書き入れた)。□は目標画像 P_d である。

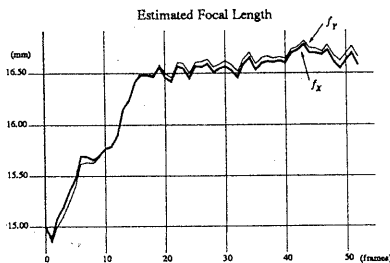


図 12: 推定された焦点距離。

実はノイズモデルが完全でないために、このような戦略が意味を持たなくなる恐れがあることを認識しておく必要がある。他の方面では最適制御や非線形フィードバックをかけた研究が行なわれている [2][4][5]。これらは、カメラの運動の動的な特性を改善するものである。つまり、ある意味で最適な軌跡で目標位置までカメラを移動させることができる。しかし、本手法で用いた、指数関数的に e を減少させる方法は、そのいずれでもない。フィードバックのかけ方については、ノイズモデルとの関係を含めて、まだ検討の余地が残されているように思われる。

8 おわりに

視覚サーボの手法を利用してカメラの外部パラメータと内部パラメータを同時に決定する方法を提案した。従来のカメラパラメータ同定法のように、1枚の画像から対象との相対位置関係を求めるのではなく、あらかじめ設定しておいた位置にカメラを移動させることで、対象との位置関係を知るものである。さらに、画像上の目標画像を内部パラメータの推定値に基づいて更新していく。最終的に目標画像とカメラから得られる画像が一致すると、外部、内部パラメータとも決定できることになる。このとき、ロボットの制御では必須であった Hand-Eye キャリブレーションが不要となるという利点もある。さらに、対象や目標位置を適切に選ぶことで、最も精度良くカメラパラメータを求めることができるようになる可能性がある。

本研究では、プリミティブとして点を用いたが、直線などを採用すると、パラメータの推定精度が上がることを期待される。

参考文献

- [1] Bernard Espiau, François Chaumette, and Patrick Rives. A new approach to visual servoing in robotics. *IEEE Trans. Robotics and Automation*, Vol. 8, No. 3, pp. 313-326, 1992.
- [2] 橋本浩一, 小谷嵩二, 木村英紀. 視覚にもとづく非線形フィードバック制御. 計測自動制御学会学術講演会, pp. 343-344, 1993.
- [3] Claude Samson, Michel Le Borgne, and Bernard Espiau. *Robot Control The Task Function Approach*. Oxford University Press, Walton Street, Oxford, 1991.
- [4] K. Hashimoto, T. Kimoto, T. Ebine, and H. Kimura. Manipulator control with image-based visual servo. In *IEEE International Conference on Robotics and Automation*, pp. 2267-2272, 1991.
- [5] K. Hashimoto, T. Kimoto, T. Ebine, and H. Kimura. Image-based dynamic visual servo for a hand-eye manipulator. *Proceedings of the International Symposium MTNS-91*, pp. 609-614. Mita Press, 1991.