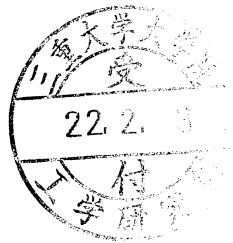


—修士学位論文—

簡易なオンラインキャリブレーションを付加した
画像特徴量推定によるビジュアルサーボ
の実機実装

Experimental Test of Visual Servo Using Simple
On-line Calibration for Estimated Image Features



平成 21 年度
三重大学大学院 工学研究科
博士前期課程 電気電子工学専攻
木下 弘輔

目次

第 1 章 緒言	1
第 2 章 ビジュアルサーボの概要と幾何学的関係	5
2.1 ビジュアルサーボの概要	5
2.1.1 位置ベース法	5
2.1.2 特徴ベース法	6
2.2 各座標系の関係	7
2.3 透視変換	7
2.4 ビジュアルサーボにおけるロボットの運動学	9
2.4.1 マニピュレータの運動学	9
2.4.2 アクティブカメラの運動学	9
第 3 章 画像処理遅れを補償した推定画像特徴量による 特徴ベースビジュアルサーボ	11
3.1 画像処理遅れを補償した画像特徴量の推定	11
3.1.1 特徴量の時間関係	11
3.1.2 マニピュレータの手先の特徴量	12
3.1.3 移動物体の特徴量	13
3.2 特徴ベースビジュアルサーボ	14
3.2.1 軌道コントローラ	14
3.2.2 トルク指令	14
第 4 章 簡易なオンラインキャリブレーションによる 推定誤差の低減	16
4.1 オンラインでの必要性	16
4.2 提案する簡易なオンラインキャリブレーション	17
4.2.1 モデルの簡易化	17
4.2.2 パラメータ更新則	18

第 5 章 実験システム	21
5.1 実験システムの構築	21
5.1.1 ハードウェア構成	21
5.1.2 各ハードウェアの役割	23
5.1.3 上位プログラムの概要	23
第 6 章 移動物体トラッキング	25
6.1 移動物体トラッキングの実機実験	25
6.1.1 実験条件	25
6.1.2 実験結果および考察	27
第 7 章 結言	31
付録	32
A 各ロボットのパラメータ	32
参考文献	33
謝辞	35
論文目録	36

記号表

w	作業座標系を表す添字
c	カメラ座標系を表す添字
m	マニピュレータ座標系を表す添字
l	画像座標系を表す添字
r	位置ベクトル
p	マニピュレータの手先の位置ベクトル
o	移動物体の位置ベクトル
I_n	$n \times n$ の単位行列
\hat{x}	x の推定値
$a \equiv b$	a を b と定義する

第1章

緒言

ロボットに求められる役割が変化してきたと言われるようになってすでに久しい[1]. 少品種大量生産から多品種変量生産への移行により, さらに高度な技能やフレキシブルな生産形態への対応が求められている. またさらに, 非製造業向けの社会支援や特殊環境下での作業が期待されるようになり, 従来の工場内などロボットに対して整備された環境ではなく, 未知環境やダイナミックに変化する環境での作業が求められている. 具体的には, 社会支援用途としてリハビリテーション支援などの福祉・医療分野や, 教育・アミューズメント分野などで注目されている[2]. また特殊環境下での用途としては, 原子力プラントや宇宙, 深海などの, 人間が行くことが出来ない危険が伴う環境での作業が注目されている.

従来の工場内などの整備された環境においては, ロボットは事前に教示された情報と自身の内界センサの情報のみで作業を行うことが出来た. また, 外界センサを持っていたとしても自律的な行動のためではなく, 遠隔操作などのための外界センサであることが多かった. しかし今, ロボットに期待されているのは整備されていない環境下での自律的な行動である. ロボットは外界センサにより周囲の環境の情報を取り込み, その情報を元に自律的に, かつ, 環境の変化に対して柔軟に行動することが求められる.

自身の情報を得るための内界センサに対して, 自身と環境との相対情報を得るための外界センサには, 6軸力覚センサや触覚センサなどの力センサや, レーザや超音波などを用いた距離センサなどがある. 外界センサの中でも, カメラなどの視覚センサは非接触で大量の情報を得ることが出来るため, 周囲の環境の情報を得るのに非常に有用である. この視覚センサを用いたロボットの制御法の一つに, ビジュアルサーボがある. 本研究ではこのビジュアルサーボについて研究を行っている.

視覚センサを用いたビジュアルフィードバック制御の中でも, 環境を認識してからおもむろに動き出す“見てから動く”静的なビジュアルフィードバックに対して, ビジュアルサーボはフィードバックループに直接視覚センサを組み込み,

周囲の状態を観測しながら反射的に反応する“見ながら動く”動的なビジュアルフィードバックである[3]。計測処理，軌道計画，アクチュエータ制御が全て並列して動作するという特徴を持つ。本研究では，ビジュアルサーボの中でもキャリブレーション誤差に対してロバストな特徴ベース法を用いる。

上述のような利点を持つビジュアルサーボであるが，問題点として，画像処理によるデータの時間遅れと画像処理サンプリング周期と制御周期が異なるという点が挙げられる。一般的なデバイスを用いた場合，画像処理サンプリング周期はロボットの制御周期よりはるかに長く，さらに制御入力に用いられる画像データには画像処理時間分の遅れが含まれているため，高速・高精度なロボットの制御を実現することは難しい。

これらの問題への対応の研究は大きく二つに分けられる[4]。一つは画像処理サンプリング周期の短い画像処理装置を開発する，ハードウェア的アプローチである。他方は対象やロボットの運動が持つ拘束条件を利用して画像処理による遅れを推定し，内部のサーボループに適切な指令を加えるソフトウェア的アプローチである。

前者については，石川ら[5]はロボット制御に対して十分なサンプリング周期である 1ms の高速なビジョンシステムを開発し，高性能なビジュアルサーボを実現している。この種のアプローチは画像処理の遅れという問題に対する本質的な解決法であり，将来的には高速な画像処理装置を用いたビジュアルサーボが主流になると考えられる。石川らの他に大明ら[6]も高速度カメラと FPGA を用いて高速な画像処理システムを開発しており，高速画像処理装置を用いた研究は広がりつつある。しかしこれらの装置は高コストであり普及しておらず，まだ一般的な画像処理装置とは言い難い。

一般的な画像処理装置を用いてソフトウェア的にアプローチする方法としては，対象とする物体の運動が既知の場合[7]や未知の場合[8][9][19]など幾つか研究されている。[8]では移動物体の運動を等速運動のノミナルモデルと仮定して画像座標系に外乱オブザーバを適用することで，センサノイズを低減し，かつ追従誤差を低減している。[9]ではモデルのパラメータを ARM（自己回帰モデル）を用いて推定し，物体の未来の位置を等速，または微少な加速度変化を持っていると仮定して予測している。また[19]では現在時刻からの 3 次元上での物体の軌道をカルマンフィルタによって予測している。

我々のグループでもこのアプローチで，前任者によって推定画像特徴量が提案されている[10][11]。[10]ではマニピュレータ手先の画像特徴量の遅れを，内部の関節角度情報とカメラとマニピュレータの幾何学的関係を使って推定することで遅れを補償している。[11]では未知の運動をする物体に対して，画像処理中は等速

度運動している，という仮定を用いて物体の特徴量を推定することで，対象物体の特徴量の遅れを補償している．

しかし，この推定画像特徴量は固定カメラを用いたシステムに適用された手法であり，アクティブカメラによるカメラ姿勢の変化は考慮されていなかった．

さらに，推定画像特徴量を含めた前述の遅れを推定する手法は，キャリブレーション誤差が推定に与える影響を考慮していない．推定画像特徴量の場合は，ロボットやカメラの幾何学的モデルである画像ヤコビ行列を利用するためキャリブレーション誤差が存在するとき，推定精度が低下し遅れを正しく補償できない．そこで，カメラの運動を考慮した推定補償を行い，さらにキャリブレーションを行うことで遅れ補償の精度を上げる必要がある．

オフラインでの高精度なカメラキャリブレーションが幾つか提案されている[12][13]．しかしながら，これらのキャリブレーションは手間がかかることや，ビジュアルサーボでは制御中にカメラやロボットの間の位置姿勢関係が変化することを考えると，オンラインでのキャリブレーションが望ましい．画像ヤコビ行列などの撮像に関する幾何学的モデルは非線形で多くのパラメータを含んだ複雑なモデルであり，全てのパラメータを最小二乗法などでキャリブレーションしても各パラメータの正確な値を求めることは難しく，さらに計算量が膨大になってしまう．

そこで前任者である金原氏は，カメラの運動を考慮した推定法と，簡易なアルゴリズムで計算量の少ないオンラインキャリブレーションを提案している[14]．[14]では画像ヤコビ行列に含まれる各パラメータをキャリブレーションする代わりに，画像ヤコビ行列に定数倍の誤差が含まれると仮定して，この定数倍の誤差をキャリブレーションしている．このようにすることでキャリブレーションするパラメータの数を減らし，アルゴリズムを簡易にし，計算量を低減できる．さらにキャリブレーションするパラメータを特定しないため，誤差が含まれるパラメータの情報を必要としないという利点もある．

この手法を用いることで，一般的な画像処理装置を用いかつキャリブレーション誤差が存在する場合でも，高精度なビジュアルサーボが実現できる．[14]では平面2軸のマニピュレータと **pan**, **tilt** 角を有する2自由度のアクティカメラを用いてビジュアルサーボシステムを移動物体トラッキングのシミュレーションに適用し，提案手法の有効性を確認しているが，実機実験は実現されていない．そこで，本論文では実機実験を通して本手法の有効性を確認する．

本論文の構成は以下の通りである．第2章でビジュアルサーボの基本的な概念と幾何学的関係について説明する．続いて，第3章では従来提案されていた画像特徴量の推定法に加えて，新たにカメラの運動を考慮した推定法について述べ，

さらに制御系についても説明する．第 5 章で提案手法である簡易なオンラインキャリブレーションについて述べ，本手法を適用するために開発した実機システムについて説明する．第 6 章で移動物体トラッキングに適用し，提案手法の有効性を確認する．

第2章

ビジュアルサーボの概要と 幾何学的関係

本章ではビジュアルサーボの概要と本研究で用いる幾何学的関係について述べる。ビジュアルサーボの概要として、位置ベース法と特徴ベース法について説明する。それから本研究で設定する座標系について述べた後、マニピュレータとアクティブカメラ、それぞれの運動学について述べる。

2.1 ビジュアルサーボの概要

フィードバックループに視覚センサを組み込んだ制御系をビジュアルサーボ系という。ビジュアルサーボでは画像情報を常にフィードバックして制御しているため、運動する物体への追従などの、環境の変化に対して柔軟な対応が可能となる。ビジュアルサーボは目標値に3次元上の位置情報をとる位置ベース法と、画像平面上の特徴量をとる特徴ベース法とに大別できる[3]。

2.1.1 位置ベース法

図 2.1 に示す、位置ベース法はサーボ系への目標値入力の対象物体とロボットの相対的な位置・姿勢により与えられる。カメラから得られる画像に基づいて対象の3次元上の位置姿勢を計算（3次元再構成）しロボットを制御する。位置ベース法には次に挙げるような特徴がある。

- (i) ロボットの制御系は通常的位置制御系になるので、これまで提案されている多彩な位置制御を適用できる
- (ii) 実際の位置関係に基づいて制御するので、情報が正確に得られるならば、高精度なビジュアルサーボが可能

- (iii) 実際には2次元情報から3次元再構成するため、画像ノイズやキャリブレーション誤差の影響を受けやすい

ここでキャリブレーション誤差とは、撮像に関する幾何学的モデルと実際の値との誤差のことをいう。実際にビジュアルサーボを行う際、多くの場合無視できない誤差である。

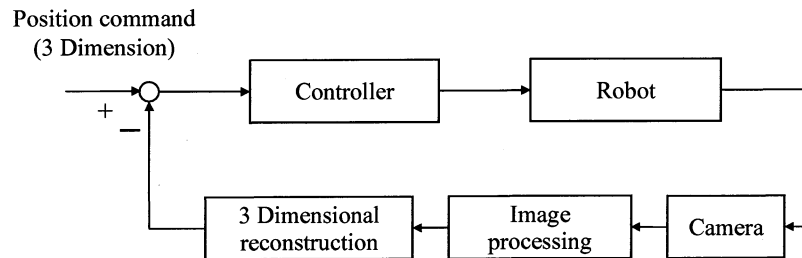


図2.1 位置ベース法

Fig. 2.1 Position based visual servo

2.1.2 特徴ベース法

図2.2に示す特徴ベース法は画像特徴量をフィードバックして制御する。ここで、画像特徴量とは、画像上のある領域の面積や重心位置、線分の長さ、傾きなどの、物体の像の特徴を表す量であり、画像の解釈を行わずに簡単な前処理だけで得られる量のことをいう[21]。サーボ系への目標値入力とは画像特徴量で与えられ、フィードバックされてきた特徴量との差でマニピュレータは駆動される。特徴ベース法には以下に挙げる特徴がある。

- (i) 画像の解釈を必要としないので処理が少なく高速に演算できる。
- (ii) 対象やロボットの3次元上の位置情報が不要なので計算量が少なく、また、対象物体やロボットの機構学的モデルの不正確さに影響されない。
- (iii) 画像平面の“見かけ”に基づいて制御するので、3次元上でのロボットの運動が所望の運動にならない場合がある。

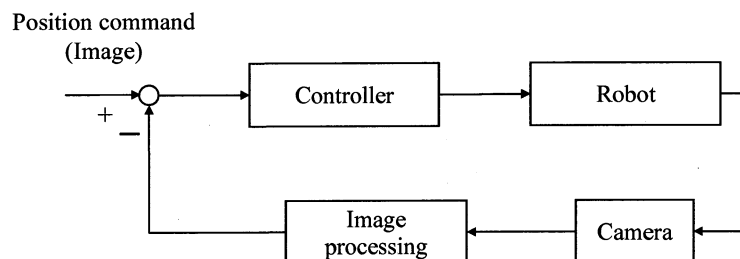


図2.2 特徴ベース法

Fig. 2.2 Feature based visual servo

本研究では特徴ベース法の(i)と(ii)の利点に注目し、特徴ベース法を用いる。また、本研究では画像特徴量として、画像上の領域の重心位置を用いる。

2.2 各座標系の関係

本研究では pan, tilt 角を有するアクティブカメラと平面 2 自由度のマニピュレータを用いる。図 2.3 に示すような 4 つの座標系を設定する。各座標系は以下の通りである。

- wS 作業座標系：基準となる 3 次元上の座標系。アクティブカメラの接地点を原点とする。
- cS カメラ座標系：カメラの撮像中心を原点とする 3 次元上の座標系。本論文では「奥行き」とは、この座標系における cY 方向の距離である。
- mS マニピュレータ座標系：マニピュレータの接地点を原点とする 3 次元上の座標系。
- iS 画像座標系：画像平面の中心を原点とする 2 次元の座標系。水平方向を X 軸、垂直方向を Z 軸とする。

2.3 透視変換

カメラでは前方の 3 次元の世界がレンズを通して 2 次元の画像平面へと投影され、像を結ぶ。このときの幾何学的関係を、本研究では最も基本的な透視変換[14]に基づいて考える。図 2.4 に透視変換のモデルを示す。画像平面は撮像中心から焦点距離 f だけ離れており、光軸 (cY 軸) は画像平面の中心の点を垂直に交わるとする。カメラ座標系の 3 次元上の点 ${}^c\mathbf{r} = [{}^cX, {}^cY, {}^cZ]^T$ がカメラの撮像により 2 次元の画像面上へ投影され、点 ${}^i\mathbf{r} = [{}^ix, {}^iz]^T$ が得られたとすると、相似三角形の関係から次式のように透視変換の式が得られる。

$$\begin{aligned} {}^ix &= f \frac{{}^cX}{{}^cY} \\ {}^iz &= f \frac{{}^cZ}{{}^cY} \end{aligned} \tag{2.1}$$

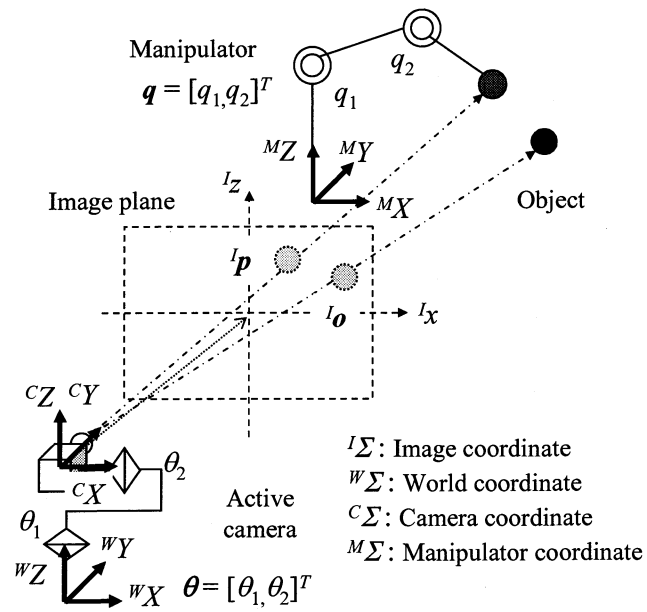


図2.3 アクティブカメラとマニピュレータを用いたビジュアルサーボシステム
Fig. 2.3 Visual servo system using active camera and manipulator

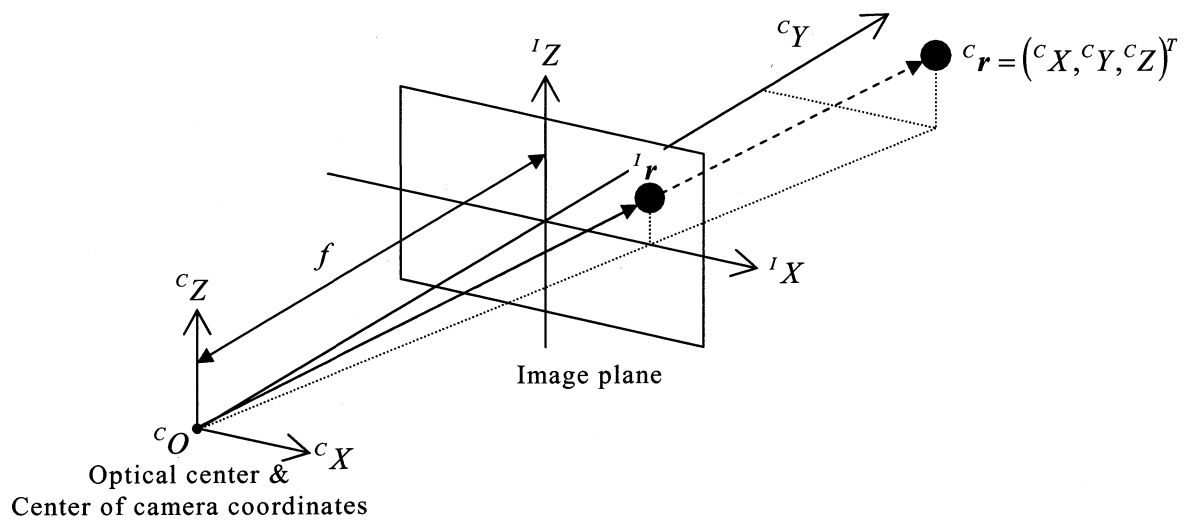


図2.4 透視変換
Fig. 2.4 Perspective transformation

2.4 ビジュアルサーボにおけるロボットの運動学

2.4.1 マニピュレータの運動学

図 2.3 における画像上のマニピュレータの手先位置 ${}^I p(q(t), \theta(t))$ は、マニピュレータの各関節の運動によって変化する。マニピュレータの関節速度（変位）と画像上の手先の速度（変位）との関係は次式のように表される。

$$\begin{aligned} {}^I \dot{p}(t) &= \frac{\partial {}^I p(t)}{\partial q^T(t)} \dot{q}(t) \\ &= {}^I J_M({}^I p, q) \dot{q}(t) \end{aligned} \quad (2.2)$$

ここで、関節速度から画像上の手先の速度までを関係付ける行列として、 ${}^I J_M({}^I p, q)$ をマニピュレータの画像ヤコビ行列と定義する。

2.4.2 アクティブカメラの運動学

画像上のマニピュレータの手先位置はマニピュレータ自体の運動によって変化するとともに、カメラの姿勢変化によっても見かけ上変化する。アクティブカメラの関節速度とマニピュレータ手先の速度の関係は次式のように表される。

$$\begin{aligned} {}^I \dot{p}(t) &= \frac{\partial {}^I p(t)}{\partial \theta^T(t)} \dot{\theta}(t) \\ &= {}^I J_C({}^I p, \theta) \dot{\theta}(t) \end{aligned} \quad (2.3)$$

同様に、カメラの関節速度から手先の見かけ上の速度までを関係付ける行列として、 ${}^I J_C({}^I p, \theta)$ をマニピュレータのアクティブカメラの関する画像ヤコビ行列と定義する。

また、カメラの姿勢が変化するとマニピュレータの手先と同様に移動物体の位置 ${}^I o(\theta(t))$ も見かけ上変化する。この関係は(2.3)式と同様に次式のように表される。

$$\begin{aligned} {}^I \dot{o}(t) &= \frac{\partial {}^I o(t)}{\partial \theta^T(t)} \dot{\theta}(t) \\ &= {}^I J_C({}^I o, \theta) \dot{\theta}(t) \end{aligned} \quad (2.4)$$

同じように ${}^I J_C({}^I \mathbf{o}, \boldsymbol{\theta})$ を移動物体に関するアクティブカメラの画像ヤコビ行列と定義する. なお, ${}^I J_C({}^I \mathbf{p}, \boldsymbol{\theta})$ と ${}^I J_C({}^I \mathbf{o}, \boldsymbol{\theta})$ はパラメータの値が ${}^I \mathbf{o}$ か ${}^I \mathbf{p}$ で異なるだけで, 形は同じ行列である.

本章で示したそれぞれのカメラのモデルや画像ヤコビ行列はカメラやマニピュレータの配置などで決定されるので既知の情報として扱える. しかし前述したように, 実際にはこれらの情報にはキャリブレーション誤差が含まれる.

第3章

画像処理遅れを補償した推定画像特徴量による特徴ベースビジュアルサーボ

本章では従来提案されている，画像特徴の推定法とビジュアルサーボによる制御法を説明する．まず特徴量の推定法をマニピュレータとアクティブカメラ，それぞれについて述べる．その後，特徴ベース法に基づくビジュアルサーボによる制御法について説明する．

3.1 画像処理遅れを補償した画像特徴量の推定

3.1.1 特徴量の時間関係

本研究では図 3.1 の時間関係のモデルに基づいて画像特徴量を推定する．図 3.1 において下添字 i は i 番目の画像処理周期を表す．また，制御周期は画像処理周期に比べて充分短い連続時間として見なしている．現在の時刻 t の時点で得られている画像データ ${}^I r(t)$ は，データが取り込まれて画像処理を経て得られるまでに実際の値から画像処理 1 周期分遅れている．さらにそこから次の処理が終わるまで 1 周期同じ値を出力し続けるため，結果として現在得られている画像特徴量は実際の値から 1～2 周期分遅れていることになる．

ここで，画像データに含まれる遅れ時間 ΔT を図 3.1 より以下のように定義する．

$$\Delta T \equiv t - t_i \quad (3.1)$$

制御周期に比べて画像処理周期が長いことは、撮像素子のメカニズムや、フレームのキャプチャや2値化などの画像処理に起因する問題である。遅れを含んだデータをそのままフィードバックすることは制御性能の劣化の原因となる。そこで、時間遅れの補償が必要となる。

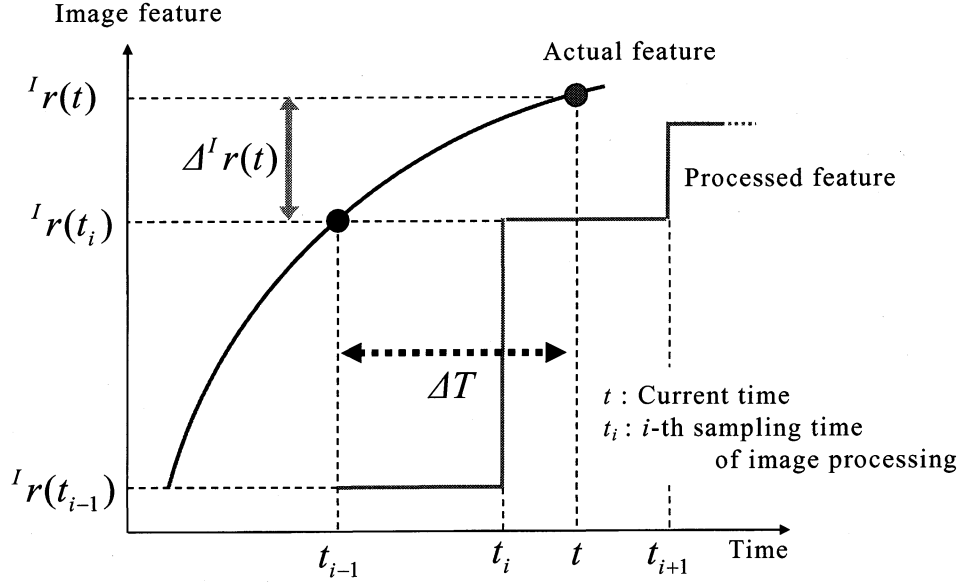


図3.1 画像特徴量の時間関係

Fig. 3.1 Time sequence of image feature

3.1.2 マニピュレータの手先の特徴量

マニピュレータの手先の画像特徴量は2つの幾何学的関係を用いて推定する。(2.2)式のマニピュレータの画像ヤコビ行列 ${}^I J_M({}^I p, q)$ と、(2.3)式のアクティブカメラのマニピュレータに関する画像ヤコビ行列 ${}^I J_C({}^I p, \theta)$ である。手先の画像特徴量は次式のように、画像上の手先の変位を計算することで推定する。

$$\begin{aligned} {}^I \hat{p}(t) &= {}^I p(t_i) + \Delta^I \hat{p}(t) \\ &= {}^I p(t_i) + \left\{ {}^I J_M({}^I p, q) \Delta q(t) + {}^I J_C({}^I p, \theta) \Delta \theta(t) \right\} \end{aligned} \quad (3.2)$$

ここで $\Delta q(t)$ 、 $\Delta \theta(t)$ はそれぞれ次式に示す ΔT の間の関節の変位である。

$$\Delta q(t) := q(t) - q(t_{i-1}) \quad (3.3)$$

$$\Delta \theta(t) := \theta(t) - \theta(t_{i-1}) \quad (3.4)$$

(3.2)式の右辺第2項がマニピュレータの運動による変位を表し、第3項がカメラの運動による見かけの変化を表している。

3.1.3 移動物体の特徴量

移動物体の画像特徴量は(2.4)式のアクティブカメラの移動物体に関する画像ヤコビ行列 ${}^I J_C({}^I \mathbf{o}, \boldsymbol{\theta})$ と、物体の画像上の運動より以下のように推定する。

$$\begin{aligned} {}^I \hat{\mathbf{o}}(t) &= {}^I \mathbf{o}(t_i) + \Delta {}^I \hat{\mathbf{o}}(t) \\ &= {}^I \mathbf{o}(t_i) + \left\{ {}^I J_C({}^I \mathbf{o}, \boldsymbol{\theta}) \Delta \boldsymbol{\theta}(t) + \Delta {}^I \mathbf{o}_{obj}(t) \right\} \end{aligned} \quad (3.5)$$

(3.5)式の右辺第2項がカメラの運動による見かけの変位で、第3項 $\Delta {}^I \mathbf{o}_{obj}$ は物体自体の運動による変位である。物体の運動モデルが既知であるならば、 $\Delta {}^I \mathbf{o}_{obj}$ はその既知のモデルに基づいて計算すればよい。本研究では物体の運動は未知として、画像処理1サンプリング期間中は等速度運動をしているという仮定を用いる。等速度運動の仮定を画像平面上の物体に適用すると、次式のようになる。

$$\Delta {}^I \mathbf{o}_{obj}(t) = {}^I \dot{\mathbf{o}}_{obj}(t_i) \Delta T \quad (3.6)$$

ここで、(3.6)式の ${}^I \dot{\mathbf{o}}_{obj}(t_i)$ は画像面上の物体の速度である。下式のように特徴量の速度求め、そこから見かけの速度を引くことで得られる。

$${}^I \dot{\mathbf{o}}(t_i) = \frac{{}^I \mathbf{o}(t_i) - {}^I \mathbf{o}(t_{i-1})}{t_i - t_{i-1}} \quad (3.7)$$

$${}^I \dot{\mathbf{o}}_{obj}(t_i) = {}^I \dot{\mathbf{o}}(t_i) - {}^I J_C({}^I \mathbf{o}, \boldsymbol{\theta}) \dot{\boldsymbol{\theta}}(t_{i-1}) \quad (3.8)$$

等速度運動という仮定は、1サンプリング間に限るという点を考慮すると、妥当な仮定であると考えられる。また、カメラと物体の相対的な奥行きが変化する場合など、3次元上で等速度運動する物体が必ずしも画像平面上でも等速度運動として投影されるとは限らない[11]ため、厳密には物体の運動は3次元上で推定するか、何らかの補正を付加するべきである。しかし本研究では簡単のため(3.6)式のように、画像平面上で等速度運動として推定する。

3.2 特徴ベースビジュアルサーボ

特徴ベース法によるアクティブカメラとマニピュレータの制御法について説明する．まず特徴ベース法に基づいて，画像座標系で構成した軌道コントローラによって加速度指令を生成する．そして外乱オブザーバを適用した関節座標系において，画像ヤコビ行列の逆行列を用いて加速度指令をトルク指令に変換する．

3.2.1 軌道コントローラ

アクティブカメラの軌道コントローラを次式に示す．

$${}^I\ddot{\mathbf{o}}^{ref}(t) = \mathbf{K}_{VC}({}^I\dot{\mathbf{r}}^{cmd}(t) - {}^I\hat{\mathbf{o}}(t)) + \mathbf{K}_{PC}({}^I\mathbf{r}^{cmd}(t) - {}^I\hat{\mathbf{o}}(t)) \quad (3.9)$$

上式は移動物体 ${}^I\mathbf{o}$ を画像上の目標位置 ${}^I\mathbf{r}^{cmd}$ に映るようにアクティブカメラを制御するコントローラである．フィードバックする信号は前節で示した時間遅れを補償した推定画像特徴量 ${}^I\hat{\mathbf{o}}$ を用いる． \mathbf{K}_{VC} ， \mathbf{K}_{PC} は速度と位置偏差に対するゲインである．

同様に，移動物体トラッキングのためのマニピュレータの軌道コントローラとして次式を用いる．

$${}^I\ddot{\mathbf{p}}^{ref}(t) = \mathbf{K}_{VM}({}^I\hat{\mathbf{o}}(t) - {}^I\hat{\mathbf{p}}(t)) + \mathbf{K}_{PM}({}^I\hat{\mathbf{o}}(t) - {}^I\hat{\mathbf{p}}(t)) \quad (3.10)$$

上式は手先を移動物体に追従するようにマニピュレータを制御するコントローラである．目標値として移動物体の推定値を用い，フィードバックする信号は同様に遅れを補償した手先の特徴量 ${}^I\hat{\mathbf{p}}$ を用いる．(3.9)式と同様に \mathbf{K}_{VM} ， \mathbf{K}_{PM} は速度，位置偏差に対するゲインである．

3.2.2 トルク指令

アクティブカメラやマニピュレータは本来，姿勢変化による慣性変動や遠心・コリオリ力や，ギアの摩擦などを含んだ非線形な動特性を持つが，各関節座標系に外乱オブザーバ[18]を適用することで，ノミナルな慣性を持った線形なシステムと見なせる．そこで，外乱オブザーバを適用したアクティブカメラとマニピュレータに対して，次式のようなトルク指令を生成する．

$$\mathbf{t}_C^{ref}(t) = \mathbf{J}_{NC} {}^I\mathbf{J}_C^{-1}(\boldsymbol{\theta}) \{ {}^I\ddot{\mathbf{o}}^{ref}(t) - {}^I\dot{\mathbf{J}}_C(\boldsymbol{\theta})\dot{\boldsymbol{\theta}}(t) \} \quad (3.11)$$

$$\mathbf{t}_M^{ref}(t) = \mathbf{J}_{NM}^I \mathbf{J}_M^{-1}(\mathbf{p}, \mathbf{q}) \left\{ {}^I \ddot{\mathbf{p}}^{ref}(t) - {}^I \dot{\mathbf{J}}_M(\mathbf{p}, \mathbf{q}) \dot{\mathbf{q}}(t) \right\} \quad (3.12)$$

\mathbf{J}_{NC} と \mathbf{J}_{NM} はマニピュレータとアクティブカメラのそれぞれのノミナル慣性行列である．上式では右辺第二項には画像ヤコビ行列の導関数の項が含まれるが，影響は小さいものとして本研究では以下のように無視したものを用いる．

$$\mathbf{t}_C^{ref}(t) \cong \mathbf{J}_{NC}^I \mathbf{J}_C^{-1}(\boldsymbol{\theta}) {}^I \ddot{\mathbf{o}}^{ref}(t) \quad (3.13)$$

$$\mathbf{t}_M^{ref}(t) \cong \mathbf{J}_{NM}^I \mathbf{J}_M^{-1}(\mathbf{p}, \mathbf{q}) {}^I \ddot{\mathbf{p}}^{ref}(t) \quad (3.14)$$

本章で説明した，特徴量の推定による遅れ補償を付加した特徴ベースビジュアルサーボのシステムは図 3.2 のようになる．

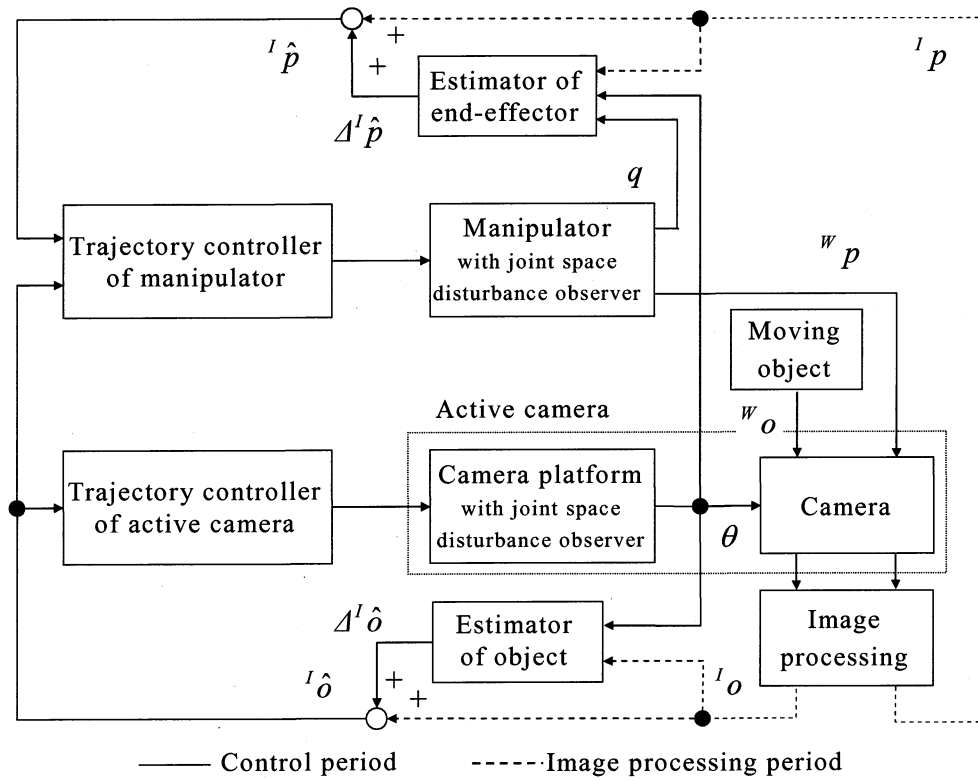


図3.2 画像特徴量の推定を用いたビジュアルサーボシステム

Fig. 3.2 Visual servo system using estimated image feature

第4章

簡易なオンライン キャリブレーションによる 推定誤差の低減

本章では提案手法であるキャリブレーション手法について述べる．オンラインでのキャリブレーションの必要性について述べた後，提案手法の具体的な説明をする．

4.1 オンラインでの必要性

前章で確認したように，キャリブレーション誤差が存在する場合，マニピュレータ手先の特徴量の時間遅れは正しく補償されない．これは画像ヤコビ行列に幾何学的関係に関するパラメータが含まれるからである．そのためキャリブレーションすることで，画像ヤコビ行列に含まれる誤差を低減する必要がある．

画像ヤコビ行列に含まれる幾何学的パラメータは大きく三つに分けられる．一つはカメラの焦点距離などカメラの内部パラメータである．もう一つはアクティブカメラやマニピュレータのリンク長などの運動学パラメータである．さらにもう一つは，カメラとマニピュレータの3次元上の相対的な位置姿勢である．前者二つのパラメータは，カメラやマニピュレータ固有のパラメータである．そのため従来提案されているようなオフラインでのキャリブレーション[12][13][16]を施せばよい．しかし後者の位置姿勢に関するパラメータはビジュアルサーボにおいてはロボットやカメラの相対的な位置関係は時々刻々と変化する．そのため本研究では，オンラインでその都度キャリブレーションしていくことが望ましい．

4.2 提案する簡易なオンラインキャリブレーション

4.2.1 モデルの簡易化

画像ヤコビ行列は非線形で多くのパラメータを含んだ複雑な関数である．例として，本研究で用いる，図 2.3 の平面 2 自由度のマニピュレータ及び，pan, tilt 角を有するアクティブカメラの場合のなかで，マニピュレータの画像ヤコビ行列 ${}^I J_M$ を次式に示す．

$$\begin{aligned}
 {}^I J_M({}^I p, q) &= \begin{bmatrix} {}^I J_{M11}({}^I p, q) & {}^I J_{M12}({}^I p, q) \\ {}^I J_{M21}({}^I p, q) & {}^I J_{M22}({}^I p, q) \end{bmatrix} \\
 &= \frac{1}{{}^c Y} \begin{bmatrix} f & -{}^I p_x & 0 \\ 0 & -{}^I p_z & f \end{bmatrix} \\
 &\quad \begin{bmatrix} \cos \theta_1 & \sin \theta_1 & 0 \\ -\sin \theta_1 \cos \theta_2 & \cos \theta_1 \cos \theta_2 & -\sin \theta_2 \\ -\sin \theta_1 \sin \theta_2 & \cos \theta_1 \sin \theta_2 & \cos \theta_2 \end{bmatrix} \\
 &\quad \begin{bmatrix} -L_b \cos q_1 - L_c \cos(q_1 + q_2) & -L_c \cos(q_1 + q_2) \\ 0 & 0 \\ -L_b \sin q_1 - L_c \sin(q_1 + q_2) & -L_c \sin(q_1 + q_2) \end{bmatrix}
 \end{aligned} \tag{4.1}$$

ここでは各パラメータの説明は省く（付録参照）．上式右辺の一つ目の行列はカメラの内部パラメータに関する部分である．二つ目の行列と奥行き ${}^c Y$ がカメラとマニピュレータの相対的な位置姿勢関係の部分である．さらに三つ目がマニピュレータの内部パラメータである．上式から分かるように，画像ヤコビ行列は多くのパラメータを含む非線形で複雑な関数である．そのため，最小二乗法などで各パラメータをそれぞれキャリブレーションしても正確な値を得ることは難しく，さらにパラメータの数の多さから計算量が増大してしまうおそれがある．

そこで本研究ではキャリブレーション誤差を含んだモデルを簡易にするため，次式のように元のモデルを定数倍に変形させる誤差が含まれるとみなす．

$${}^I \hat{J}_M({}^I p, q) = \begin{bmatrix} \alpha_1 {}^I J_{M11}({}^I p, q) & \alpha_2 {}^I J_{M12}({}^I p, q) \\ \alpha_3 {}^I J_{M21}({}^I p, q) & \alpha_4 {}^I J_{M22}({}^I p, q) \end{bmatrix} \tag{4.2}$$

$${}^I\hat{\mathbf{J}}_C(\boldsymbol{\theta}) = \begin{bmatrix} \beta_1 {}^I J_{C11}(\boldsymbol{\theta}) & \beta_2 {}^I J_{C12}(\boldsymbol{\theta}) \\ \beta_3 {}^I J_{C21}(\boldsymbol{\theta}) & \beta_4 {}^I J_{C22}(\boldsymbol{\theta}) \end{bmatrix} \quad (4.3)$$

(4.1)式に含まれるような各パラメータをキャリブレーションする代わりに、(4.2)式の α 、(4.3)式の β をキャリブレーションすることで誤差モデルを簡易にすることが出来る。さらに誤差が含まれるパラメータを特定しないため、事前に誤差が含まれる情報を必要としないといった利点も生まれる。さらに誤差の情報を必要としないために、カメラのレンズの歪みのようなモデル化されていない誤差にも対応できると考えられる。

4.2.2 パラメータ更新則

まず、 i 番目の画像処理サンプリング周期における推定誤差 $\mathbf{e}(t_i)$ を次式のように定義する。

$$\begin{aligned} \mathbf{e}(t_i) &= {}^I\hat{\mathbf{p}}(t_{i-1}) - {}^I\mathbf{p}(t_i) \\ &= \Delta^I\hat{\mathbf{p}}(t_{i-1}) - \Delta^I\mathbf{p}(t_i) \end{aligned} \quad (4.4)$$

ここで、(3.2)式において(4.2)、(4.3)式を ${}^I\mathbf{J}_M, {}^I\mathbf{J}_C$ の替わりに用いると(3.2)式の $\Delta^I\hat{\mathbf{p}}(t)$ は以下のように変形できる。

$$\begin{aligned} \Delta^I\hat{\mathbf{p}}(t_i) &= \begin{bmatrix} \Delta^I\hat{p}_x(t_i) \\ \Delta^I\hat{p}_z(t_i) \end{bmatrix} = {}^I\hat{\mathbf{J}}_M({}^I\mathbf{p}, \mathbf{q})\Delta\mathbf{q}(t_{i-1}) + {}^I\hat{\mathbf{J}}_C({}^I\mathbf{p}, \boldsymbol{\theta})\Delta\boldsymbol{\theta}(t_{i-1}) \\ &= \begin{bmatrix} \alpha_1 {}^I J_{M11}({}^I\mathbf{p}, \mathbf{q})\Delta q_1(t_{i-1}) + \alpha_2 {}^I J_{M12}({}^I\mathbf{p}, \mathbf{q})\Delta q_2(t_{i-1}) + \beta_1 {}^I J_{C11}(\boldsymbol{\theta})\Delta\theta_1(t_{i-1}) + \beta_2 {}^I J_{C12}(\boldsymbol{\theta})\Delta\theta_2(t_{i-1}) \\ \alpha_3 {}^I J_{M21}({}^I\mathbf{p}, \mathbf{q})\Delta q_1(t_{i-1}) + \alpha_4 {}^I J_{M22}({}^I\mathbf{p}, \mathbf{q})\Delta q_2(t_{i-1}) + \beta_3 {}^I J_{C21}(\boldsymbol{\theta})\Delta\theta_1(t_{i-1}) + \beta_4 {}^I J_{C22}(\boldsymbol{\theta})\Delta\theta_2(t_{i-1}) \end{bmatrix} \\ &= \begin{bmatrix} \boldsymbol{\psi}_x^T(t_i)\boldsymbol{\zeta}_x(t_i) \\ \boldsymbol{\psi}_z^T(t_i)\boldsymbol{\zeta}_z(t_i) \end{bmatrix} \end{aligned} \quad (4.5)$$

ここで、 $\boldsymbol{\psi}$ 、 $\boldsymbol{\zeta}$ はそれぞれ、キャリブレーションパラメータベクトルとそれ以外の項を表し、以下のように表せる。

$$\boldsymbol{\psi}_x = [\alpha_1 \quad \alpha_2 \quad \beta_1 \quad \beta_2]^T \quad (4.6)$$

$$\boldsymbol{\psi}_z = [\alpha_3 \quad \alpha_4 \quad \beta_3 \quad \beta_4]^T \quad (4.7)$$

$$\zeta_x = \begin{bmatrix} {}^I J_{M11}({}^I \mathbf{p}, \mathbf{q}) \Delta q_1(t_{i-1}) \\ {}^I J_{M12}({}^I \mathbf{p}, \mathbf{q}) \Delta q_2(t_{i-1}) \\ {}^I J_{C11}(\boldsymbol{\theta}) \Delta \theta_1(t_{i-1}) \\ {}^I J_{C12}(\boldsymbol{\theta}) \Delta \theta_2(t_{i-1}) \end{bmatrix} \quad (4.8)$$

$$\zeta_z = \begin{bmatrix} {}^I J_{M21}({}^I \mathbf{p}, \mathbf{q}) \Delta q_1(t_{i-1}) \\ {}^I J_{M22}({}^I \mathbf{p}, \mathbf{q}) \Delta q_2(t_{i-1}) \\ {}^I J_{C21}(\boldsymbol{\theta}) \Delta \theta_1(t_{i-1}) \\ {}^I J_{C22}(\boldsymbol{\theta}) \Delta \theta_2(t_{i-1}) \end{bmatrix} \quad (4.9)$$

(4.5)式のモデルに基づいて、勾配法[17]を用いてオンラインで ψ_x , ψ_z を更新していくことでキャリブレーションを行う。推定誤差 e の二乗を評価関数として ${}^I x$ 方向, ${}^I z$ 方向それぞれについて, (4.4)式を各成分に分解して, 次式のように定義する。なお, (4.10), (4.11)式の $\Delta^I \hat{\mathbf{p}}(t_{i-1})$ は, (4.5)式の変形結果を用いる。

$$\begin{aligned} E_x(t_i) &= \frac{1}{2} (e_x(t_i))^2 \\ &= \frac{1}{2} (\Delta^I \hat{\mathbf{p}}_x(t_{i-1}) - \Delta^I \mathbf{p}_x(t_i))^2 \\ &= \frac{1}{2} (\boldsymbol{\psi}_x^T(t_{i-1}) \boldsymbol{\zeta}_x(t_{i-1}) - \Delta^I \mathbf{p}_x(t_i))^2 \end{aligned} \quad (4.10)$$

$$\begin{aligned} E_z(t_i) &= \frac{1}{2} (e_z(t_i))^2 \\ &= \frac{1}{2} (\Delta^I \hat{\mathbf{p}}_z(t_{i-1}) - \Delta^I \mathbf{p}_z(t_i))^2 \\ &= \frac{1}{2} (\boldsymbol{\psi}_z^T(t_{i-1}) \boldsymbol{\zeta}_z(t_{i-1}) - \Delta^I \mathbf{p}_z(t_i))^2 \end{aligned} \quad (4.11)$$

これらの評価関数が小さくなるように, 次式のように評価関数の勾配をとって画像処理周期毎に ψ_x , ψ_z を更新していく。

$$\Delta \psi_x(t_i) = -\mathbf{G} \frac{\partial E_x(t_i)}{\partial \psi_x(t_i)} = -\mathbf{G} \boldsymbol{\zeta}_x(t_i) e_x(t_i) \quad (4.12)$$

$$\Delta \psi_z(t_i) = -\mathbf{G} \frac{\partial E_z(t_i)}{\partial \psi_z(t_i)} = -\mathbf{G} \boldsymbol{\zeta}_z(t_i) e_z(t_i) \quad (4.13)$$

$$\psi_x(t_{i+1}) = \psi_x(t_i) + \Delta\psi_x(t_{i+1}) \quad (4.14)$$

$$\psi_z(t_{i+1}) = \psi_z(t_i) + \Delta\psi_z(t_{i+1}) \quad (4.15)$$

ここで \mathbf{G} は $\mathbf{G} \equiv \gamma \mathbf{I}_4$ ($\gamma > 0$) となるような対角定数ゲイン行列である. γ の値が更新の速度を決定する.

以上のように, (4.12)~(4.15)式で更新した α, β を使った画像ヤコビ行列 ((4.2), (4.3)式) を画像特徴量の推定に用いることで, キャリブレーション誤差の影響による推定誤差を低減できる. 提案手法のオンラインキャリブレーションを付加したビジュアルサーボ系は図 4.1 のようになる.

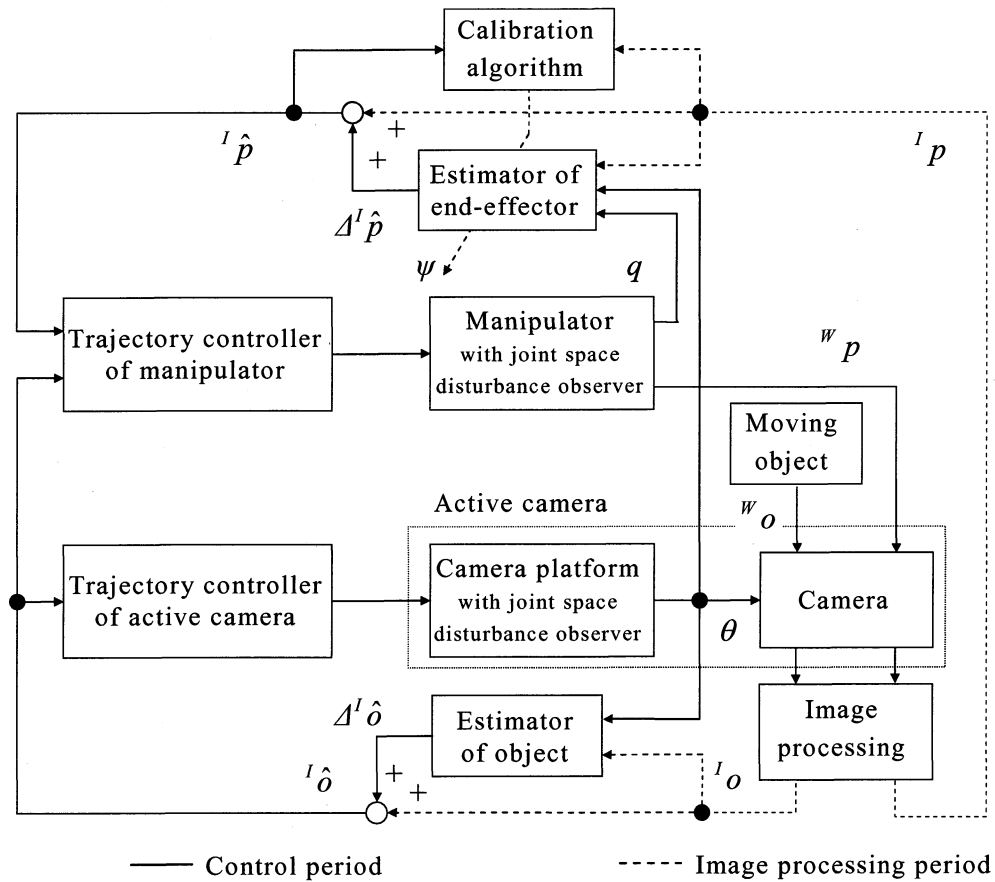


図4.1 簡易なオンラインキャリブレーションを付加したビジュアルサーボ系
Fig. 4.1 Block diagram of visual servo system with simple on-line calibration

第5章

実験システム

5.1 実験システムの構築

提案手法の有効性を確認するための実験システムを構築した。以下で詳細を述べる。

5.1.1 ハードウェア構成

本研究で使用する実験システムの構成を図 5.3 に示す。システムはアクティブカメラ（図 5.2）、4 軸マニピュレータ（図 5.3）及び制御用の PC からなる。

制御用 PC の OS はリアルタイム OS の一種である ART-Linux を用いた。制御プログラムは全て C 言語で実装した。

アクティブカメラはカラー CCD カメラ（Watec 製）と雲台部分の pan, tilt 角制御用の AC サーボモータ 2 つによって構成している。

マニピュレータは 4 自由度を持つ安川電機製、教育用ロボットを用い、そのうちの先端 2 関節を用いて平面二軸マニピュレータとしている。

制御用 PC の PCI バススロットに表 5.1 に示す PCI ボード (Interface 製) を挿入し、各ボードに演算負荷を分散する事で、市販の PC でのリアルタイム制御を実現している。

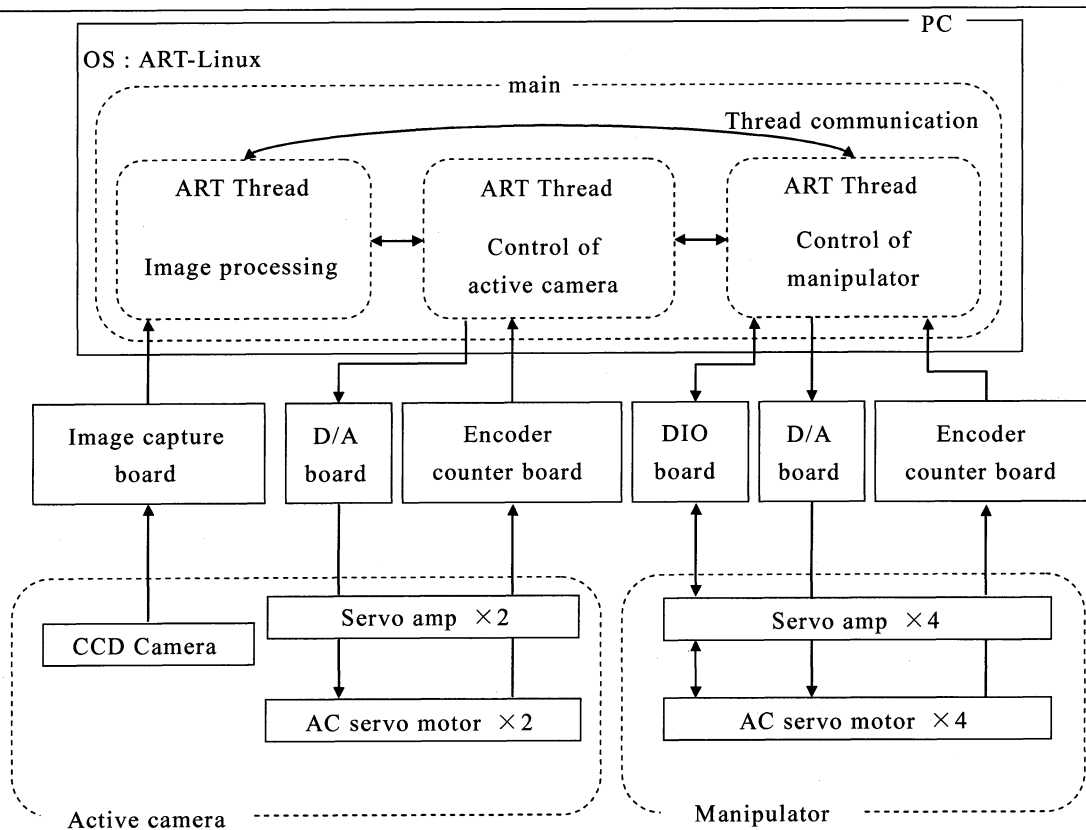


図5.3 ビジュアルサーボシステムのハードウェア構成

Fig. 5.1 Hardware setup of visual servo system

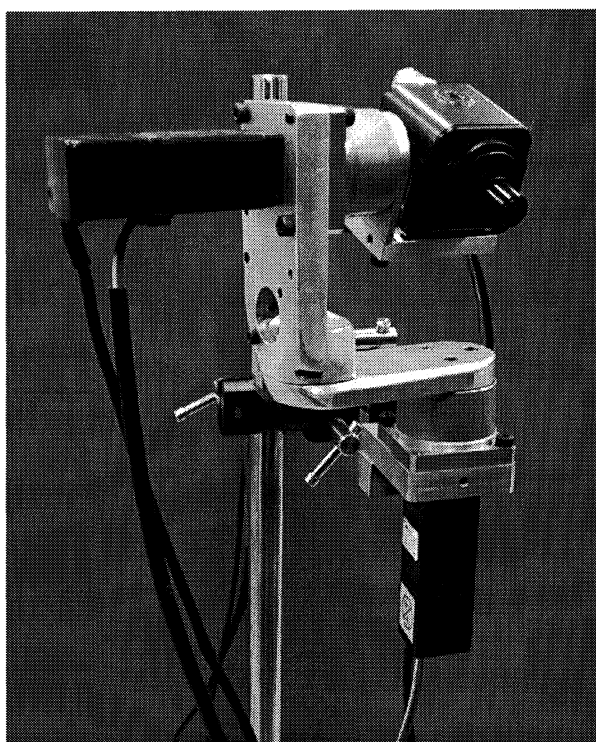


図5.2 アクティブカメラ

Fig. 5.2 Active camera

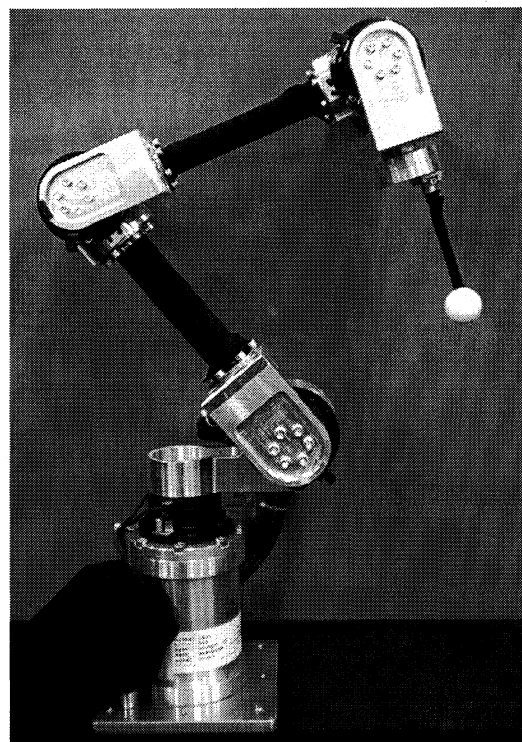


図5.3 マニピュレータ

Fig. 5.3 Manipulator

Table 5.1 Information of hardware

PC	OS	ART-Linux 2.4.22
	CPU clock	2.8[GHz]
	Main memory	512[MB]
Control of active camera	Servo amp×2	SGMM-A3CS
	Servo motor×2	SGMM-A3C312
	D/A board	PCI-3521
	Encoder counter board	PCI-6201
Control of manipulator	Servo amp×4	SGMAS-A5A2A2E
	Servo motor×4	SGDS-A5F01A
	D/A board	PCI-3310
	Encoder counter board	PCI-6205C
	DIO board	PCI-2752C
Image processing	CCD Camera	WAT-221S
	Image capture board	PCI-5532

5.1.2 各ハードウェアの役割

各ハードウェアの役割について述べる.

画像処理ボードはカメラからの情報をキャプチャし, 2 値化し, 重心位置計算まで行って, 処理終了時に上位プログラムへ割り込み通知で情報更新している.

D/A ボードは, 上位プログラムで計算されたモータへのトルク指令 (デジタル値) をアナログ電圧に変換し, サーボアンプへ出力する.

エンコーダカウンタボードは各モータのロータリーエンコーダのパルスを読み込み, 上位プログラムで位置・速度を計算できるようにしている.

DIO ボードはマニピュレータ保護回路の制御及び, マニピュレータ関節モータの絶対値エンコーダとの通信を行う.

カメラはカラーCCD カメラであり, 撮像方式が NTSC 方式であるので, 撮像周期は 33ms である.

5.1.3 上位プログラムの概要

本研究では PC の OS として ART-Linux を使用している. その理由として以下のものが挙げられる.

- ・ 安価である (無償で提供されている)
- ・ リアルタイム制御が保証されている

- ・ RT-Linux に比べ構造的に安全性が高い

次に、ART-Linux で用いる C 言語によるプログラムの構成について説明する．プログラムには比較的処理の軽いスレッドと呼ばれる処理を用いている．このスレッドとはプログラムの実行単位であり，複数のスレッドが CPU を交互に占有することによって見かけ上同時実行が可能となっている．さらに各スレッドの処理周期を ART-Linux によって一定の時間に規定することで，周期の違う複数のスレッドで，同時，並列なリアルタイム制御を実現している[20]．

各スレッドについて述べる（図 4.1 参照）．画像処理スレッドは，画像から抽出する特徴点である重心位置情報の更新を，画像処理ボードからの割り込みによって行うスレッドである．ここで，画像処理ボードによる特徴点抽出処理は 1 周期 30～40ms（3 回の処理が 100ms）で行われるが，1 周期に対して 1 つの特徴点しか抽出できないので，本手法では 2 つの画像特徴点（マニピュレータ手先，物体）を交互に取得する（図 5.4 参照）．そのため，特徴点抽出対象の切り替え設定の時間も含めて，画像処理スレッドの周期を 50ms とする．従って，各特徴点の更新周期は 100ms である．

アクティブカメラの制御スレッドは，画像処理スレッドで更新される特徴点情報をもとに，“手先位置の推定”，“物体位置の推定”，を行い，“アクティブカメラの軌道コントローラ”，“pan, tilt モータの制御”，“キャリブレーションアルゴリズム”，“マニピュレータの軌道コントローラ”までの処理を行っている．本スレッドの制御周期は 1ms である．

最後にマニピュレータの制御スレッドは，アクティブカメラの制御スレッドによって計算されるマニピュレータ関節トルク指令を受けて，“マニピュレータ関節モータの制御”のみを行っている．制御周期は同じく 1ms である．

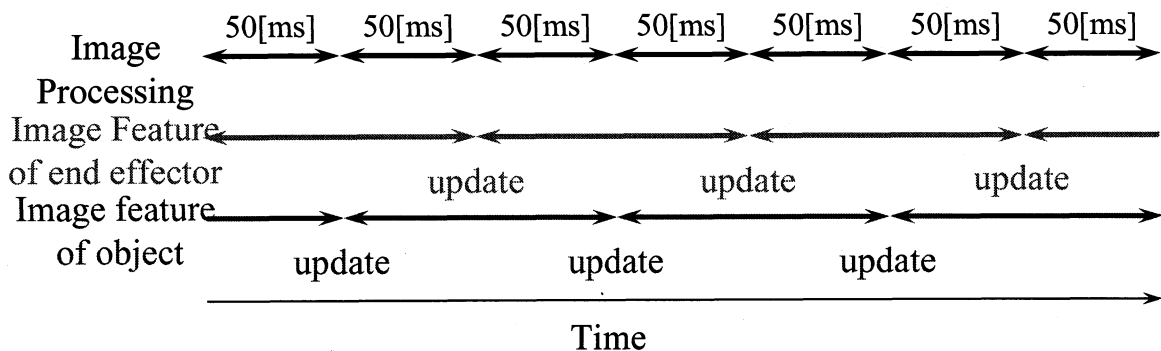


図5.4 画像特徴点の更新
Fig. 5.4 Update of image points

第6章

移動物体トラッキング

6.1 移動物体トラッキングの実機実験

提案手法の有効性を確認するため、移動物体トラッキングの実験を行った。

6.1.1 実験条件

実験条件を図 6.1 に、実験パラメータを表 6.1 に示す。マニピュレータとアクティブカメラは(3.13)式と(3.14)式の軌道コントローラを用いた。アクティブカメラを画面の中心に移動物体が映るように制御し、マニピュレータは、その手先が画面上で移動物体に追従するように制御した。ただし、手先と移動物体が画面内で完全に重なると見分けがつかなくなるため、手先位置は ${}^I Z$ 方向に-150pixel のオフセットをのせて追従させている。

また、マニピュレータの動作平面を手前に 30[deg] ねじり ${}^I J_M({}^I p, q)$ に誤差を与えた。この条件で提案手法のキャリブレーションを付加した場合と付加しなかった場合の二通り行った。

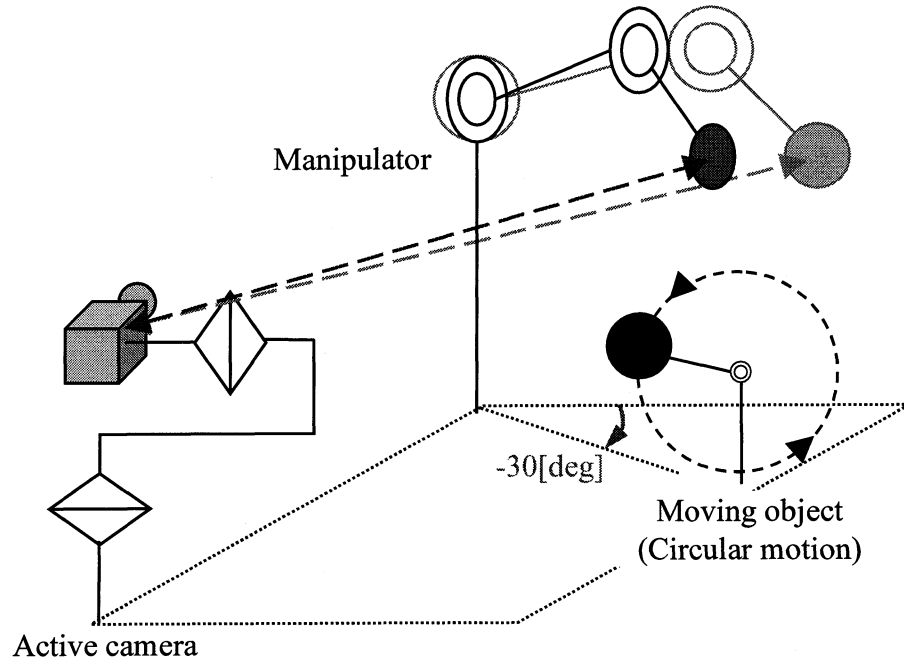


図6.1 実験条件
Fig. 6.1 Experimental setup

表6.1 実験パラメータ

Table 6.1 Experimental parameters		
Size of image plane	[pixel ²]	640 × 480
Unit cell size	[μm]	8.4(H) × 9.8(V)
Number of effective Pixels		768(H) × 494(V)
Image sensor cell size	[mm]	6.4(H) × 4.8(V)
Focal length	[mm]	0.008
Image processing period	[ms]	100.0
Control period	[ms]	1.0
$K_{PM} = \text{diag}()$	[s ⁻²]	(15, 10)
$K_{VM} = \text{diag}()$	[s ⁻¹]	(13, 10)
$K_{PC} = \text{diag}()$	[s ⁻²]	(60, 60)
$K_{VC} = \text{diag}()$	[s ⁻¹]	(40 , 40)
Radius of circular motion	[m]	0.15
Angular velocity of circular motion	[rad/s]	1.25π

6.1.2 実験結果および考察

実験結果を図 6.2～図 6.7 に示す。図 6.2～図 6.5 における対象物体に関しては、キャリブレーションを施していない場合でもほぼ同じ結果となるため一つしか載せていない。

図 6.2 はイメージセンサに撮像された、特徴量座標の 40s 分の結果であり、図 6.3 は図 6.2 において、マニピュレータ手先に関して拡大した図である。これらの図よりキャリブレーションを付加しない場合に比べて、提案手法のキャリブレーションを付加した場合は、移動物体への手先の追従が良くできていることがわかる。

図 6.4 は図 6.2 の結果を時間軸で表した結果であり、図 6.5 は図 6.4 の結果より求めた手先のトラッキング誤差である。ここで、手先のトラッキング誤差とは、画像上の移動物体と手先の偏差を意味する。参考までに、キャリブレーション誤差が存在しない場合のトラッキング誤差も示してある。これらの図からも、キャリブレーションを付加した事により、移動物体に対して偏差が小さくなり画面の中心に近付いていることがわかる。初期位置がわずかに異なるため単純な比較は出来ないが、図 6.5 の点線がコントローラの性能など制御系の特性に起因するトラッキング誤差で、それ以上の誤差が、キャリブレーション誤差が原因となるトラッキング誤差であるといえる。キャリブレーションを付加しなかった場合、誤差は多いところで約 30pixel である。一方、キャリブレーションを付加した場合は約 20pixel ほどである。キャリブレーション誤差が存在しない場合の結果は約 20pixel であることを考えると、実験後半において、提案手法を付加することでそれに近い結果になっているといえる。

図 6.6 にキャリブレーションパラメータの更新結果を示す。この図より、各パラメータが 40s である程度収束していることが確認できる。各パラメータはおおよそ 0.9～1.1 の範囲に収まった。

図 6.7 は上図が評価関数に用いた推定誤差の時間変化であり、下図がこれの後半 10s を拡大した図である。図 6.6 のように更新されたキャリブレーションパラメータを用いて特徴量の推定を行うことにより、図 6.7 に示すように、大部分の時間において推定誤差が低減できている。言い換えると推定精度が向上しているので、その結果として、前述のようにトラッキング誤差が低減できている(図 6.4～図 6.5 参照)。

以上より、提案するキャリブレーションを付加することで、キャリブレーション誤差による推定値の誤差を低減でき、その結果移動物体トラッキングの精度が向上することを確認できた。

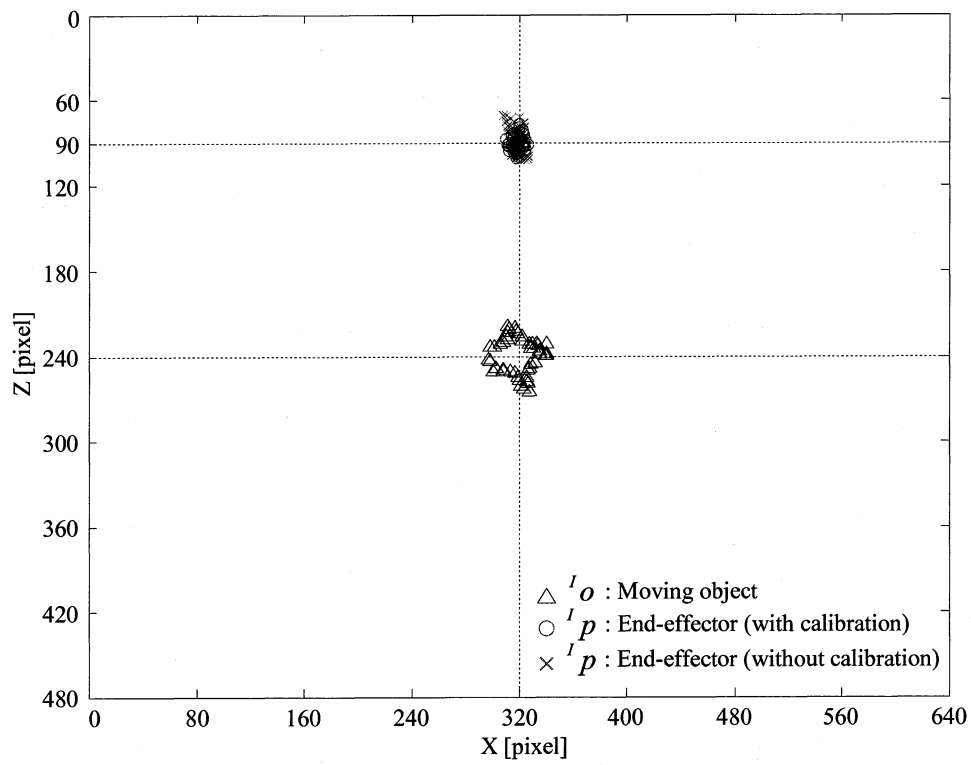


図6.2 画面遷移
Fig. 6.2 Screen transition

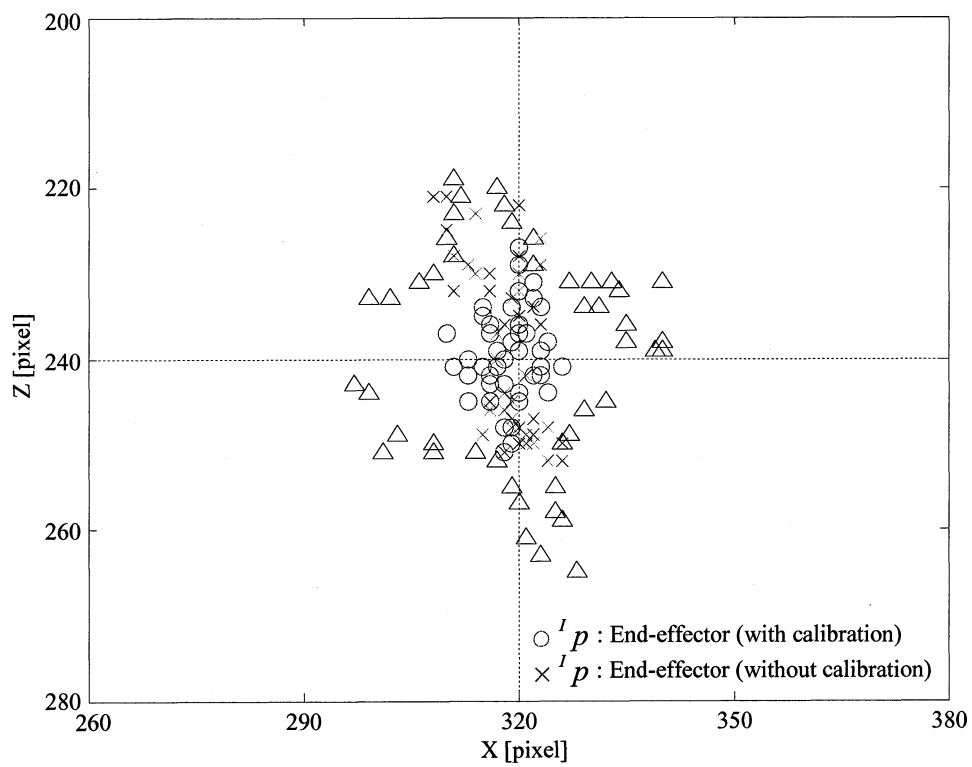


図6.3 画面遷移(拡大)
Fig. 6.3 Screen transition (An enlarged figure)

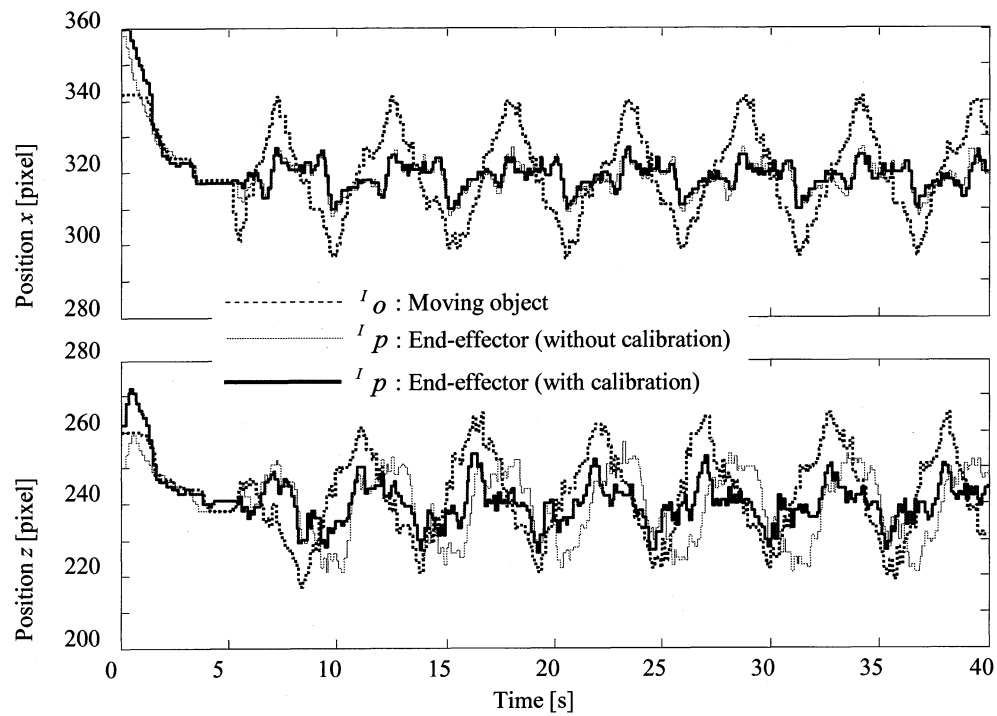


図6.4 手先と物体の特徴量の時間応答(上：水平方向，下：垂直方向)
Fig. 6.4 Time response of image feature of end-effector and moving object

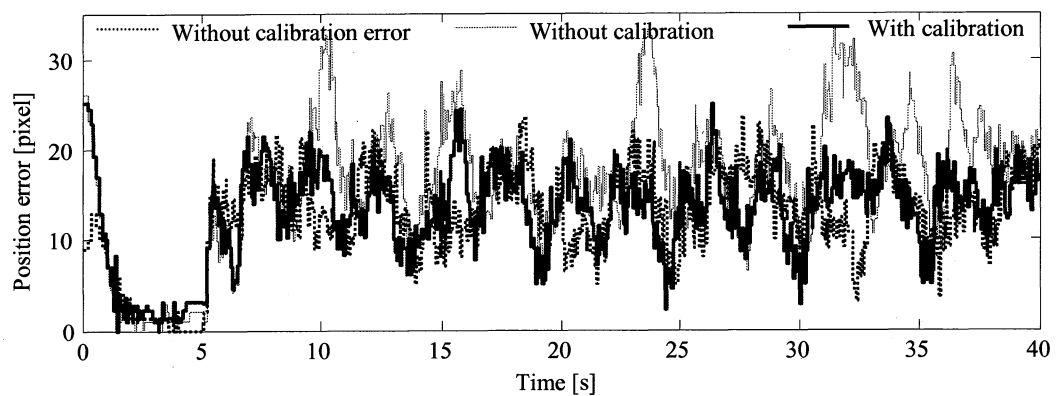


図6.5 手先のトラッキング誤差（画像上の手先と物体の距離）
Fig. 6.5 Tracking error of end-effector
(The gap between end-effector and moving object in the image)

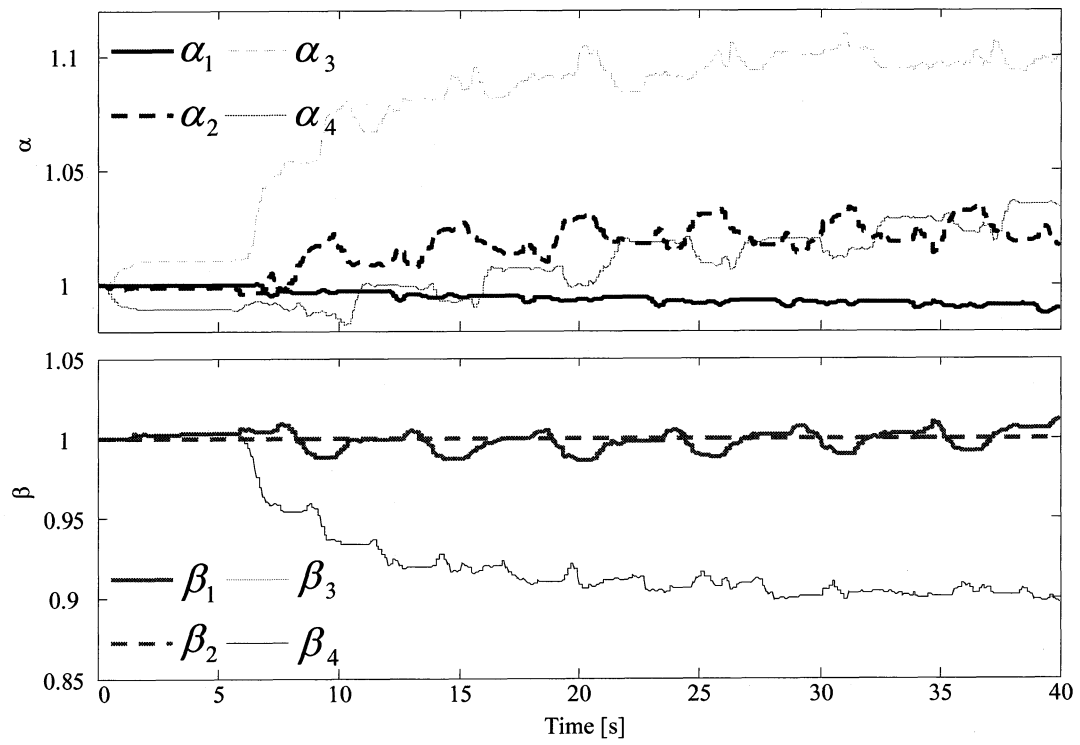


図6.6 キャリブレーションパラメータ
Fig. 6.6 Calibration parameters

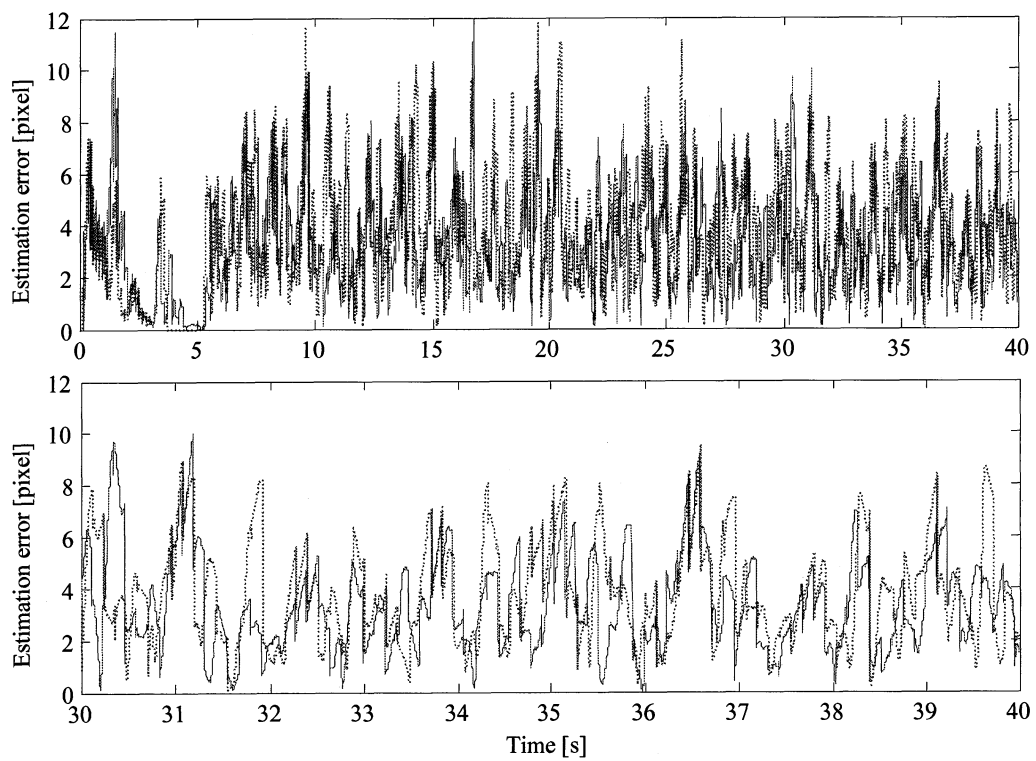


図6.7 手先の特徴量の推定誤差
Fig. 6.7 Estimation error of end-effector

第7章

結言

本研究では、従来提案されていた推定画像特徴量に簡易なオンラインキャリブレーションを付加することを新たに提案した。先行研究では高精度なビジュアルサーボを実現するために画像データの時間遅れを推定して補償していたが、この推定画像特徴量はキャリブレーション誤差の影響を考慮していなかった。そのため、新たにビジュアルサーボに適したオンラインでのキャリブレーションを付加することで、キャリブレーション誤差による推定誤差の低減を目的とした。

画像ヤコビ行列に含まれる各パラメータをキャリブレーションする代わりに、画像ヤコビ行列の要素をキャリブレーションすることでパラメータ誤差を含んだモデルを簡易に表し、アルゴリズムを簡易にした。誤差モデルを簡易にすることで、誤差が含まれるパラメータの情報を必要としなくなった。

簡易なモデルに基づいてキャリブレーションを行っても推定誤差を低減でき、ビジュアルサーボの精度を上げることができることを、移動物体トラッキングの実験で確認した。

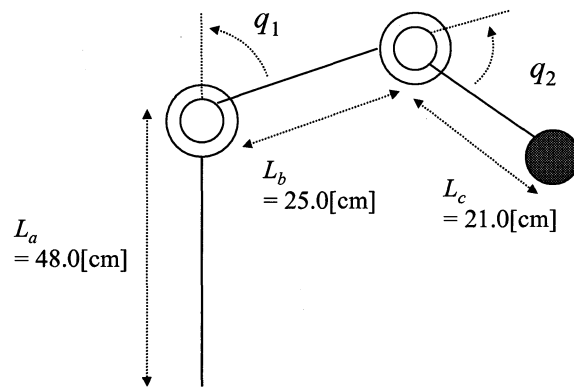
今後の課題は、キャリブレーションパラメータの更新の際に局所解について考慮していないことが挙げられる。

また、ハードウェアの問題ではあるが、画像処理ボードが1回の撮像につきひとつの特徴量しか抽出できないため、移動物体、手先それぞれの画像更新周期が100 ms となっている。もし、1回の撮像で得られた画像データからソフトウェア的に特徴量抽出ができれば、画像更新周期は最速33msまで改善できるはずである。そうなれば、移動物体の運動モデルに等速度運動ではなく、より現実的な仮定である、等加速度運動を適用し推定精度や追従精度を改善できると考える。

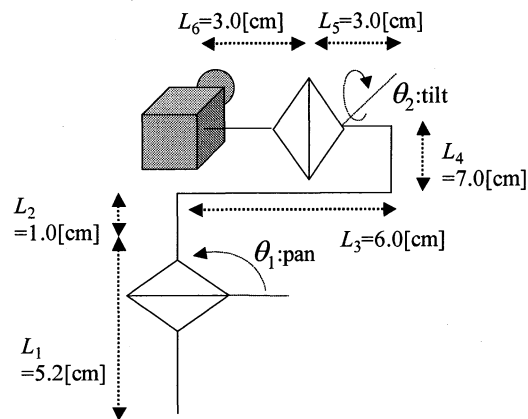
付録

A. 各ロボットのパラメータ

本研究では平面 2 自由度のマニピュレータと pan, tilt の 2 自由度のアクティブカメラを用いている．それぞれのリンクパラメータを図 A.1, 図 A.2 に示す．



図A.1 マニピュレータのリンクパラメータ
Fig. A.1 Link parameters of manipulator



図A.2 アクティブカメラのリンクパラメータ
Fig. A.2 Link parameters of active camera

参考文献

- [1] 松日楽信人, 大明準治: “わかりやすいロボットシステム入門”, オーム社 (2003)
- [2] “ロボットに関する特許出願技術動向調査”, 特許庁 (2002)
<http://www.jpo.go.jp/shiryou/index.htm>
- [3] 橋本浩一: “視覚と制御”, SICE 制御部門大会前日のワークショップ, pp.37-68 (2001)
- [4] 石川正俊: “ビジュアルサーボイングの現状と将来”, 計測と制御, Vol.40, No.9, pp.615-617 (2001)
- [5] 中坊嘉宏, 石川正俊: “1ms 高速ビジョンを用いたビジュアルサーボイング”, 計測と制御, Vol.40, No.9, pp.636-640 (2001)
- [6] 大明準治: “1000fps 高速度画像処理のアクティブカメラへの応用 – 通常照明・複雑背景での非特定移動物体トラッキング –, 日本ロボット学会誌, Vol.23, No.3, pp.252-285 (2005)
- [7] 遠藤公誉, 田中弘一, 荒川賢一, 武川直樹: “予見制御による視覚軌道追従制御法”, 日本ロボット学会誌, Vol.15, No.4, pp.564-572 (1997)
- [8] 猿田訓彦, 藤本博志, 堀洋一: “運動推定オブザーバによる特徴量予測を用いたビジュアルサーボ”, 電学論 D, Vol.122, No.5, pp.516-521 (2001)
- [9] A.Elnagar, K.Gupta: “Motion Prediction of Moving Objects Based on Autoregressive Model”, IEEE Transactions on Systems, Man, and Cybernetics, Vol.28, No.6, pp.803-810 (1998)
- [10] 駒田諭, 吉田将也, 堀孝正: “推定画像特徴量を用いたロボットのビジュアルサーボ”, 電学論 D, Vol.123, No.10, pp.1200-1205 (2003)

-
- [11] D.Nishio, M.Nkamura, S.Komada, and J.Hirai: "Tracking of Moving Object by Manipulator Using Estimate Image Feature and Its Error Correction on Image Planes", in Proc. International Workshop on Advanced Motion Control, pp.653-657 (2004)
- [12] R.Y.Tsai: "A versatile camera calibration technique for high- accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses", IEEE Journal of Robotics and Automation, Vol.RA-3, No.4, pp.323-344 (1987)
- [13] Z.Zhang: "A flexible new technique for camera calibration", Technical Report, MSR-TR-98-71, Microsoft Research (1998)
- [14] 金原一史, 駒田諭, 平井淳之: "画像特徴量推定のための簡易なオンラインキャリブレーションを用いたビジュアルサーボ", 電気学会産業計測制御研究会 (2006-3 静岡)
- [15] 出口光一郎: "ロボットビジョンの基礎", コロナ社 (2000)
- [16] 川崎晴久: "ロボット工学の基礎", 森北出版 (1991)
- [17] 鈴木隆: "アダプティブコントロール", コロナ社 (2001)
- [18] T.Murakami et al.: "Advanced Motion Control in Mechatronics -A Tutorial-", in Proc.IEEE Workshop on Intelligent Motion Control, Vol.1, (SL)pp.9-17 (1990)
- [19] J.M.Borg, M.Mehrandezh, R.G.Fenton and B.Benhabib :
"Navigation-Guidance- Based Robotic Interception of Moving Objects in Industrial Settings", Journal of Intelligent and Robotic Systems, Vol.33 (2002)
- [20] 武居直行: "ART-Linux でリアルタイム制御"
http://www.toyota.nitech.ac.jp/~takesue/rt_ctrl/art0400.html
- [21] 谷内田正彦: "ロボットビジョン", 昭晃堂 (1990)

謝辞

本論文は、筆者の三重大学大学院工学研究科 博士前期課程電気電子工学専攻在学中における研究活動の成果をまとめたものであります。

本研究の遂行および本論文の作成にあたり、熱心な御指導と適切な御意見を賜りました三重大学教授工学博士 平井淳之先生に深く感謝いたします。また先生には機会のある度に技術者、研究者、そして社会人の先輩として貴重な御意見を頂き、併せてこの場を借りて感謝いたします。

本研究の遂行および本論文の作成にあたり、適切な御指導と御助言を頂きました同大学準教授 駒田諭先生に深く感謝いたします。

また日頃から本研究の遂行および学生生活において、貴重な御意見を頂きました同大学準教授 弓場井一裕先生、同大学技術職員 中村勝氏に深く感謝いたします。

本論文の作成にあたり、査読頂き貴重な御意見を頂きました同大学助教授 松井博和先生に深く感謝いたします。

本研究の遂行にあたり、ビジュアルサーボグループの先輩として貴重な御指導を頂きました同大学院卒業者 蛭川達彦氏、中村亮太氏に感謝いたします。

また、面識はありませんが、この研究の前任者として多くの研究資料を残して下さった、同大学院卒業者 金原一史氏に感謝いたします。

研究室の同期として時には切磋琢磨し合い、時には互いを励まし合い、また時には馴れ合いながら研究を遂行してきた上村章仁君、宇佐見秀徳君、灰屋和勇君、村上洋輔君に深く感謝いたします。

同じ研究グループとして協力し合った、山口彰紀氏、鈴木勇介氏、山口敦由君、小池堂夫君、臼井伸充君、朴木豊君、近藤秀映君に感謝します。皆様の今後の活躍を心より期待しております。

また、何かと至らない自分を受け入れ、多くの御協力を頂き、学部から大学院までの三年間の研究生活を充実したものにしてくれた電機システム研究室の皆さんに深く感謝いたします。電機システム研究室の更なる発展を心より願っております。

最後に、大学院まで進学する機会を与えて下さり、さらには何一つ不自由なく学生生活を送らせて頂いた両親に、また自分の心の支えとなってくれた愛猫に心から感謝いたします。

論文目録

- [1] Satoshi Komada, Kousuke Kinoshita, Tatsuhiko Hirukawa, and Junji Hirai :
“Image Feature based Navigation of Nonholonomic Mobile Robots with
Active Camera”, Proceedings of the 17th IFAC World Congress in Seoul
(2008-8) FrA20.1, pp.14714--14719

- [2] 木下 弘輔, 駒田 諭, 平井 淳之 : 「自律移動ロボットの誘導指令を考
慮した測距センサによる障害物回避」, 平成 20 年度電気学会産業応用部
門大会 講演論文集, Y-70, p70, (2008.8)

- [3] 木下弘輔, 駒田諭, 平井淳之 : “画像特徴量推定のための簡易なオンラ
インキャリブレーションを用いたビジュアルサーボの実機実装”, 電気
学会産業計測制御研究会 (2010-3) (予定)