

## 時空間画像を用いたカメラの軌道制御

遠藤公誉 田中弘一 武川直樹\*

NTT ヒューマンインタフェース研究所, 東京都武蔵野市

本報告は対象中心座標系で記述された目標軌道を追従するようにカメラの運動を制御する方法を提案する。カメラ画像はカメラと撮影対象との相対的な位置関係によって決定されることから、本手法ではカメラが目標軌道上を運動した時に撮影される時空間画像から抽出した画像特徴点の軌跡を、カメラの目標軌道の表現として用いる。カメラ運動の制御時にはカメラが撮影する画像から求められる画像特徴点の軌跡と目標とする画像特徴点の軌跡との二乗誤差が最小になるようにカメラの運動の制御を行なうことにより、カメラを対象中心座標系で記述された目標軌道に追従させることが可能になる。またシミュレーションによる実験で本手法の有効性を示す。

## Trajectory Tracking Control for a Camera Using Spatio-Temporal Images

Kimitaka Endo, Koichi Tanaka and Naoki Mukawa\*

NTT Human Interface Laboratories, Musashino-shi, Tokyo, Japan

We propose a camera motion control method for tracking a reference trajectory in an object centered coordinate system. In our method, the trajectory is represented by the several locuses of image feature points on an object extracted from spatio-temporal images, which are taken by a camera moving along the trajectory, because a image is determined by the geometric relation between the camera and the object. Camera motion is controlled by minimizing the distances between the image feature points extracted from camera images and the reference ones. The camera thus can track the reference trajectory. The simulation results show the effectiveness of the proposed method.

---

\*現在 NTT データ通信 (株), 神奈川県川崎市  
Now, NTT DATA Communication Systems Co.,  
Kawasaki-shi, Kanagawa, Japan

## 1 はじめに

本報告ではビデオカメラ(以下カメラと略)によって撮影される時空間画像を用いることにより、対象物体に対し相対的に設定された目標軌道を追従するようにカメラの運動を制御する方法について述べる。

まず最初に本研究で解決しようとする問題の位置づけを明らかにするため、ロボットのマニピュレータに作業を行なわせる場合を考える。マニピュレータに作業を行なわせる時、グリッパやツール類を取り付けたマニピュレータの手先にどのような運動をさせるかを表す運動軌道は作業対象に対して相対的に指定される。そのためマニピュレータは作業対象の位置、姿勢の変化や位置決め誤差、またマニピュレータ自体の動作誤差を吸収するために、作業対象に相対的に指定された軌道を追従するように運動を制御できる能力が必要となる。

すなわちマニピュレータの手先の運動軌道は作業対象を中心とする座標系で記述されるということができ、マニピュレータはその軌道を追従できる能力が必要となる。

このようにある対象に対して相対的に指定された軌道を追従するように、あるものの運動を制御することは重要な技術である。その実現のためには以下の条件を満たすことが必要となる。

- 制御する物体と対象との相対的な関係で制御が行なわれること
- 位置と姿勢を含めた空間的、時間的に渡る軌道を追従できること
- 軌道を追従する時の運動の滑らかさなど、運動全体の最適性が保たれること

これらの点から従来の手法について考察する。外界に存在する対象を基準に制御を行なうためには外界センサが必要になるが、その外界センサとしてはカメラを用いたものが多い。そのカメラをマニピュレータ等に取り付け、カメラからの画像によってカメラ自体の運動を制御するビジュアルサーボと呼ばれる方法がある[1][2]。これはカメラが撮影する対象物体の画像特徴が目標とする画像特徴に一致するようにカメラの位置と姿勢を制御するものである。この方法はカメラと対象物体の相対的な位置関係で決定される画像のみを用いて制御されるため、対象物体に相対的な制御が実現される。

しかしこのビジュアルサーボは位置制御であるため、位置や姿勢が時々刻々と変化していく軌道に追

従するような制御にはそのままでは適用不可能である。このビジュアルサーボによってカメラを与えられた目標軌道に沿って運動するように制御しようとする場合、あらかじめある時間間隔をおいた時刻における軌道上の点からカメラが撮影した画像列を用意しておき、制御時にビジュアルサーボの目標とする画像を逐次切替えていくという方法が考えられる。しかしこの方法も各時刻ではその時点での目標画像しか用いない局所的なフィードバック制御をつなげたものに過ぎないため、制御が不連続になり滑らかに軌道を追従することができない。

そこで我々は対象物体に対して設定された目標軌道上をカメラが運動した時に得られる時空間画像から画像特徴点の軌跡を抽出したものを目標軌道の表現として用いる。そしてそれを元に最適制御の手法を用いて制御時に画像から求まる画像特徴点が目標とする画像特徴点の軌跡に追従するようにカメラの運動を制御する方法を提案する。

本手法では、各時刻における制御はその時刻以降の制御結果がある評価関数を最小にするように決定されるので、軌道全体を考慮した大局的な軌道制御が可能になる。

本報告ではまず2章において記法の説明やモデル化等の準備を行なう。次に3章では制御のアルゴリズムについて述べ、4章ではその具体的手法について述べる。5章ではシミュレーション実験の結果について述べ、最後に6章で結論を述べる。

## 2 準備

本章では次章でのアルゴリズムの説明のための準備として記法の説明、モデル化等を行なう。

カメラを中心とする座標系( $\Sigma_C$ )とカメラの画像面の座標系( $\Sigma_S$ )を図1のように設定する。その際 $\Sigma_S$ の $X-Y$ 平面は $\Sigma_C$ の $x-y$ 平面と平行、 $\Sigma_C$ の $z$ 軸はカメラの光軸と一致し $\Sigma_S$ の原点を通るように設定する。次に $\Sigma_C$ の各座標軸に沿ったカメラの並進速度をそれぞれ $V_x, V_y, V_z$ 各座標軸回りの回転角速度をそれぞれ $V_\theta, V_\phi, V_\psi$ と設定する。

カメラで撮影しようとする対象物体上に少なくとも4個は同一直線上にない $n$ 個( $n \geq 4$ )の特徴点 $p_i$ ( $i = 1, 2, \dots, n$ )を設定し $\Sigma_C$ における座標を $(x_i, y_i, z_i)$ とする。次に特徴点 $p_i$ が画像面に投影された点を画像特徴点 $P_i$ で表し $\Sigma_S$ における座標を $(X_i, Y_i)$ と表す。中心投影を仮定しているため $p_i$ の座標と $P_i$ の座標との間には(1)式の透視変換の関係が成り立つ。

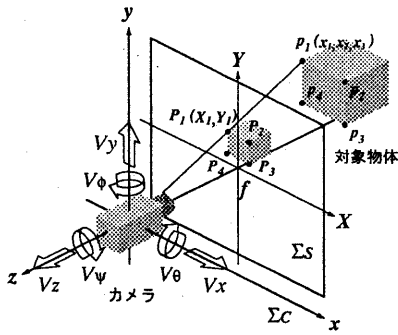


図 1: カメラモデル

$f$  はカメラの焦点距離である。

$$\begin{cases} X_i = f \cdot x_i / z_i \\ Y_i = f \cdot y_i / z_i \end{cases} \quad (1)$$

ここで2つのベクトルを導入する。 $n$ 個の画像特徴点  $P_i$  の  $X, Y$  座標を並べた  $2n$ 個の要素からなる画像特徴ベクトル  $\boldsymbol{x} = (X_1, Y_1, X_2, Y_2, \dots, X_n, Y_n)^T$  とカメラの各軸の並進速度、回転角速度を並べた6個の要素からなるカメラ制御ベクトル  $\boldsymbol{u} = (V_x, V_y, V_z, V_\theta, V_\phi, V_\psi)^T$  である。

さらにサンプリング間隔を  $\Delta T$  とし、 $t_j = j \cdot \Delta T$  ( $j$  は0以上の整数) と表される時刻  $t_j$  を以降単に  $j$  と表すことにする。時刻  $j$  における画像特徴ベクトルとカメラ制御ベクトルをそれぞれ  $\boldsymbol{x}_j, \boldsymbol{u}_j$  と表すと、これらのベクトルの間には(2)式で表される関係がある。

$$\boldsymbol{x}_{j+1} = \boldsymbol{x}_j + G_j \boldsymbol{u}_j \quad (2)$$

行列  $G_j$  は Interaction Matrix[1] または Image Jacobian[2] と呼ばれるもので、カメラの並進速度、回転角速度と画像特徴点の座標の変化量との関係を表した Jacobian である。

この行列  $G_j$  はカメラの並進速度及び回転角速度が微小であれば(3)式のように近似される。ここで  $X_{ij}, Y_{ij}$  はそれぞれ時刻  $j$  における画像特徴点  $P_i$  の  $X, Y$  座標、 $z_{ij}$  は時刻  $j$  における特徴点  $p_i$  の  $z$  座標である。この式を見てわかるように  $G_j$  は画像特徴点の  $X,$

$Y$  座標と特徴点の  $z$  座標で記述することができる。

$$G_j \simeq \Delta T \begin{pmatrix} -f/z_{1j} & 0 & X_{1j}/z_{1j} \\ 0 & -f/z_{1j} & Y_{1j}/z_{1j} \\ -f/z_{2j} & 0 & X_{2j}/z_{2j} \\ 0 & -f/z_{2j} & Y_{2j}/z_{2j} \\ \vdots & \vdots & \vdots \\ -f/z_{nj} & 0 & X_{nj}/z_{nj} \\ 0 & -f/z_{nj} & Y_{nj}/z_{nj} \\ fX_{1j}Y_{1j}/z_{1j}^2 & -f(1+X_{1j}^2/z_{1j}^2) & Y_{1j} \\ -f(1+Y_{1j}^2/z_{1j}^2) & -fX_{1j}Y_{1j}/z_{1j}^2 & -X_{1j} \\ fX_{2j}Y_{2j}/z_{2j}^2 & -f(1+X_{2j}^2/z_{2j}^2) & Y_{2j} \\ -f(1+Y_{2j}^2/z_{2j}^2) & -fX_{2j}Y_{2j}/z_{2j}^2 & -X_{2j} \\ \vdots & \vdots & \vdots \\ fX_{nj}Y_{nj}/z_{nj}^2 & -f(1+X_{nj}^2/z_{nj}^2) & Y_{nj} \\ -f(1+Y_{nj}^2/z_{nj}^2) & -fX_{nj}Y_{nj}/z_{nj}^2 & -X_{nj} \end{pmatrix} \quad (3)$$

また(2)式は画像特徴ベクトルとカメラ制御ベクトルをそれぞれこのシステムの状態ベクトル、入力ベクトルと見ると離散時間システムの状態方程式と考えることができる。

### 3 制御アルゴリズム

本章では時空間画像を用いて対象中心座標系で設定された目標軌道にカメラを追従させる制御アルゴリズムについて説明する。

#### 3.1 目標軌道の設定

対象物体に対してカメラにどのような運動をさせるかを、対象を中心とする座標系 ( $\Sigma_0$ ) で目標軌道として設定する(図2)。この軌道は時間の関数として表され、開始時刻を0、終了時刻を  $N$  ( $N \geq 0$ ) と設定する。またこの軌道にはカメラの位置だけでなくカメラがどの方向を向いているかを表す姿勢も含んでいる。

カメラが撮影する画像は対象物体とカメラとの相対的な位置関係だけで決定されるので、上記のような  $\Sigma_0$  での軌道が与えられると、その軌道に従ってカメラが運動した時の時空間画像を求めることができる。

そこでその時空間画像の時刻  $j$  における画像から画像特徴点の座標を求め画像特徴ベクトルとして表したものを参照画像特徴ベクトル  $\boldsymbol{x}_{dj}$  とし、それらを全時間に渡って時間順に並べたセットを参照画像特徴ベクトル列  $\{\boldsymbol{x}_{d0}, \boldsymbol{x}_{d1}, \dots, \boldsymbol{x}_{dN}\}$  とする。

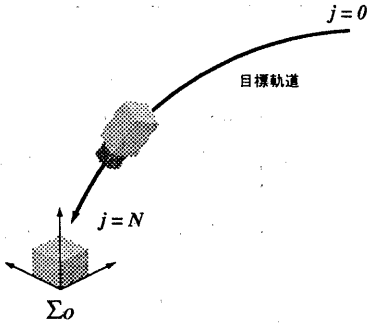


図 2: 目標軌道の設定

対象物体の幾何モデルが与えられていれば、その対象物体の同一直線上にない最少 4 点の画像特徴点の座標から対象物体に対するカメラの 3 次元的位置、姿勢は唯一に決定される。ゆえに参照画像特徴ベクトル  $\mathbf{x}_{d_j}$  は時刻  $j$  での対象中心座標系  $\Sigma_0$  におけるカメラの目標位置、姿勢を表しており、それらの全時間に渡るセットである参照画像特徴ベクトル列  $\{\mathbf{x}_{d_0}, \mathbf{x}_{d_1}, \dots, \mathbf{x}_{d_N}\}$  は対象中心座標系  $\Sigma_0$  におけるカメラの目標軌道を表現していることになる。

そこで制御時の時刻  $j$  にカメラが撮影する画像から求められる画像特徴ベクトル  $\mathbf{x}_j$  が参照画像特徴ベクトル  $\mathbf{x}_{d_j}$  に一致するように時刻  $j-1$  におけるカメラ制御ベクトル  $\mathbf{u}_{j-1}$  を決定し、その  $\mathbf{u}_{j-1}$  に従ってカメラの運動を制御すれば、カメラは  $\Sigma_0$  であらかじめ与えられた目標軌道に追従して運動すると考えられる。

そこでこの考えを具体化するために次の評価関数を導入する。

$$I_N = \sum_{j=1}^N \{(\mathbf{x}_j - \mathbf{x}_{d_j})^T \mathbf{Q}_j (\mathbf{x}_j - \mathbf{x}_{d_j}) + \mathbf{u}_{j-1}^T \mathbf{H}_{j-1} \mathbf{u}_{j-1}\} \quad (4)$$

$\mathbf{Q}_j, \mathbf{H}_{j-1}$  は正値対称行列である。

この評価関数の第 1 項は制御時の画像特徴ベクトル  $\mathbf{x}_j$  と目標である参照画像特徴ベクトル  $\mathbf{x}_{d_j}$  との誤差を表し、また第 2 項はカメラの運動エネルギーを表している。 $\mathbf{Q}_j, \mathbf{H}_{j-1}$  はそれぞれの項に対する重み付け行列であり、評価関数に対して画像特徴ベクトルの誤差の効果を大きくするか、またはカメラの運動エネルギーの効果を大きくするかを設定できる。

この評価関数を小さくするためには、第 1 項から制御時の画像特徴ベクトル  $\mathbf{x}_j$  を参照画像特徴ベクトル  $\mathbf{x}_{d_j}$  に収束させることが要請され、また第 2 項からカメラ制御ベクトル  $\mathbf{u}_j$  が過大になり制御入力の範

囲を越えないように抑制される。

この評価関数を最小にするようなカメラ制御ベクトル列  $\{\mathbf{u}_0, \mathbf{u}_1, \dots, \mathbf{u}_{N-1}\}$  を求めることができれば、それに従ってカメラの運動を制御することにより制御時の画像特徴ベクトル  $\mathbf{x}_j$  が参照画像特徴ベクトル  $\mathbf{x}_{d_j}$  に収束していき、ひいてはあらかじめ与えられている対象中心座標系  $\Sigma_0$  での目標軌道に従ってカメラが運動すると考えられる。

### 3.2 カメラ制御ベクトルの決定法

本節では (4) 式の評価関数を最小にするカメラ制御ベクトル列  $\{\mathbf{u}_0^o, \mathbf{u}_1^o, \dots, \mathbf{u}_{N-1}^o\}$  を求める方法を説明する。

(2) 式を離散時間システムの状態方程式と捉えたと、評価関数 (4) を最小にするシステムへの入力  $\mathbf{u}_j^o$  は最適制御の手法を用いて求めることができる (参考文献 [3])。

すなわち (4) 式の評価関数を最小にするカメラ制御ベクトル  $\mathbf{u}_j^o$  を求めるための最適制御則は (5) 式によって与えられる。

$$\mathbf{u}_j^o = -\mathbf{R}_{N-j} \mathbf{x}_{d_{j+1}} - \mathbf{S}_{N-j} \mathbf{x}_j \quad (5)$$

ただし行列  $\mathbf{R}_{N-j}, \mathbf{S}_{N-j}$  は以下の式に従って漸化的に求めることができる。

$$\mathbf{D}_0 = \mathbf{Q}_N \quad (6)$$

$$\mathbf{E}_0 = \mathbf{O} \quad (7)$$

$$\mathbf{F}_0 = -2\mathbf{Q}_N \quad (8)$$

$$\begin{aligned} \mathbf{R}_{N-j} = & \{ \mathbf{G}_j^T (\mathbf{D}_{N-(j+1)} + \mathbf{D}_{N-(j+1)}^T) \mathbf{G}_j + 2\mathbf{H}_j \}^{-1} \cdot \\ & \mathbf{G}_j^T \mathbf{F}_{N-(j+1)}^T \end{aligned} \quad (9)$$

$$\begin{aligned} \mathbf{S}_{N-j} = & \{ \mathbf{G}_j^T (\mathbf{D}_{N-(j+1)} + \mathbf{D}_{N-(j+1)}^T) \mathbf{G}_j + 2\mathbf{H}_j \}^{-1} \cdot \\ & \mathbf{G}_j^T (\mathbf{D}_{N-(j+1)} + \mathbf{D}_{N-(j+1)}^T) \end{aligned} \quad (10)$$

$$\begin{aligned} \mathbf{D}_{N-j} = & \frac{1}{2} (\mathbf{D}_{N-(j+1)} + \mathbf{D}_{N-(j+1)}^T) (\mathbf{I} - \mathbf{G}_j \mathbf{S}_{N-j}) \\ & + \mathbf{Q}_j \end{aligned} \quad (11)$$

$$\mathbf{F}_{N-j} = \mathbf{E}_{N-j} - 2\mathbf{Q}_j \quad (12)$$

$$\mathbf{x}_{d_j}^T \mathbf{E}_{N-j} = \mathbf{x}_{d_{j+1}}^T \mathbf{F}_{N-(j+1)} (\mathbf{I} - \mathbf{G}_j \mathbf{S}_{N-j}) \quad (13)$$

$\mathbf{O}$  は零行列、 $\mathbf{I}$  は単位行列を表す。

(5) 式的最適制御則はフィードバック行列  $\mathbf{R}_{N-j}, \mathbf{S}_{N-j}$  が時間とともに変化する時変係数系のフィードバック制御となっていることがわかる。

## 4 具体的手順

本章では前章で説明した制御アルゴリズムの具体的な実現手法を説明する。

### 4.1 参照画像特徴ベクトル列を生成する

まず対象中心座標系  $\Sigma_O$  であらかじめ与えられた目標軌道に従ってカメラを運動させた時に撮影される時空間画像を求める。次にその時空間画像の時刻  $j$  の画像から求めた画像特徴点の座標から参照画像特徴ベクトル  $\mathbf{x}_{d_j}$  を求め、それらを全時間に渡って並べた参照画像特徴ベクトル列  $\{\mathbf{x}_{d_0}, \mathbf{x}_{d_1}, \dots, \mathbf{x}_{d_N}\}$  を生成する。

画像から画像特徴点を求めるには、実際にカメラを目標軌道に従って運動させ、その時にカメラが撮影する画像から画像処理等によって画像特徴点を抽出して求める方法、あるいは対象物体の幾何モデルとカメラモデルを与えることによりコンピュータグラフィックスの手法を用いて計算機で計算によって求める方法などがある。

### 4.2 フィードバック行列を求める

次に(6)式から(13)式までの式を用いフィードバック行列  $R, S$  を求める。制御実行前に  $\{\mathbf{x}_{d_0}, \mathbf{x}_{d_1}, \dots, \mathbf{x}_{d_N}\}, \{G_0, G_1, \dots, G_{N-1}\}, \{Q_1, Q_2, \dots, Q_N\}, \{H_0, H_1, \dots, H_{N-1}\}$  を与えておくことにより、 $\{R_1, R_2, \dots, R_N\}, \{S_0, S_1, \dots, S_N\}$  も全て求めておくことができる。

ところでフィードバック行列を求めるためには行列  $G_j$  を与えなければならないが(3)式からわかるように  $G_j$  を求めるためには時刻  $j$  においてどのような画像が撮影されるか、すなわちカメラが対象物体に対してどのような運動をするかということがわかっている必要がある。しかし制御を行なう前にはカメラが実際にどのような運動をするかを知ることができない。そこで制御時にはカメラは与えられた目標軌道通りに運動するものと仮定して、その時に撮影される画像から求められる画像特徴点の座標  $X, Y$  すなわち参照画像特徴ベクトル  $\mathbf{x}_{d_j}$  及び特徴点の  $z$  座標を用いて  $G_j$  を計算して与えた。

また(3)式を見ると  $G_j$  を求めるためにはカメラ座標系  $\Sigma_C$  での対象上の特徴点の  $z$  座標の値が必要である。しかし一台のカメラによる単眼視を仮定する場合、この  $z$  座標の値は直接に求めることができない。この問題を解決する方法として  $\Sigma_O$  でのカメラの目標軌道が陽に記述されていれば、それから  $\Sigma_C$  で

の対象上の特徴点の  $z$  座標値を求めることができる。一方目標軌道が陽に記述されていない場合でも対象上の最少6点(同一平面上にあれば最少4点)の特徴点からカメラのキャリブレーションの手法を用いて特徴点の  $z$  座標値を求めることが可能である[2]。

このフィードバック行列を求める処理は比較的計算量が大さいが、ここまでは制御を行なうための準備の処理であり、実時間で処理を行なわなければならない制御時の処理と異なり時間の制約は緩く、時間のかかる計算量の大きい処理も許容できる。

### 4.3 カメラ制御ベクトルを求める

ここからは実際に実時間で制御を行なう処理である。4.1節で求めた参照画像特徴ベクトル列  $\{\mathbf{x}_{d_0}, \mathbf{x}_{d_1}, \dots, \mathbf{x}_{d_N}\}$ , 4.2節で求めたフィードバック行列  $\{R_0, R_1, \dots, R_N\}, \{S_0, S_1, \dots, S_N\}$  を読み込み記憶しておく。制御時の時刻  $j$  においてカメラが撮影する画像から画像処理などによって対象の画像特徴点を抽出しその座標を求め画像特徴ベクトル  $\mathbf{x}_j$  を求める。この  $\mathbf{x}_j$  を(5)式に代入することによりカメラ制御ベクトル  $\mathbf{u}_j$  を得ることができる。

カメラ制御ベクトル  $\mathbf{u}_j$  を求めるためには対象に関する3次元的情報を復元する必要はなく、また画像特徴ベクトルが求めればその後は行列とベクトルの積が2回、ベクトルの和が1回という少ない計算量でカメラ制御ベクトルを求めることができる。

### 4.4 カメラ制御ベクトルに従ってカメラを制御する

4.3節で求めたカメラ制御ベクトル  $\mathbf{u}_j$  に従ってカメラを制御すればカメラは対象中心座標系であらかじめ与えられた目標軌道に追従して運動する。

## 5 実験結果

本手法の有効性を確認するためにワークステーション上でシミュレーション実験を行なった。対象物体の幾何モデル、カメラの位置及び姿勢情報、カメラパラメータを設定することにより、カメラが撮影する画像を計算によってシミュレートして用いた。対象物体としては  $40 \times 40 \times 50$  の直方体の物体を想定し、世界座標系の  $(900, 900, 0)$  の位置に底面の中心が来るように置いた(図3)。またカメラの焦点距離は50に設定した。

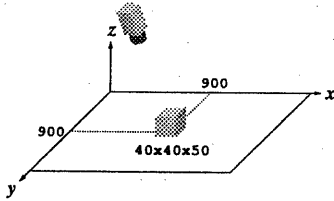


図 3: 設定した状況

## 5.1 目標軌道の設定

まず対象物体に対してカメラにどのような運動を行なわせるかを表す目標軌道のうちカメラの位置を(14)式のように設定した。またカメラの姿勢はカメラの視方向が常に対象物体である直方体の上面の中心を向くように設定した。

$$\begin{aligned}
 x &= \begin{cases} 1.5j + 800 & (0 \leq j < 100) \\ 950 & (100 \leq j < 180) \\ -2.5j + 1400 & (180 \leq j \leq 200) \end{cases} \\
 y &= \begin{cases} 2j + 400 & (0 \leq j < 100) \\ 3.125j + 287.5 & (100 \leq j < 180) \\ 850 & (180 \leq j \leq 200) \end{cases} \\
 z &= \begin{cases} -5j + 1000 & (0 \leq j < 100) \\ -3.75j + 875 & (100 \leq j < 180) \\ 200 & (180 \leq j \leq 200) \end{cases}
 \end{aligned} \quad (14)$$

設定した目標軌道を図 4 に示す。これは時刻 0 に座標 (800, 400, 1000) からスタートし、時刻 200 (= N) に座標 (900, 850, 200) のゴールに達する軌道である。

次にこの目標軌道に従ってカメラを運動させた時にカメラが撮影する対象物体の画像特徴点の軌跡を求めた。画像特徴点としては対象物体の上面の 4 つの頂点を用いた。各画像特徴点の軌跡を図 5 に示す。上で求めた画像特徴点の軌跡から参照画像特徴ベクトル列を求め(6)式から(13)式までを用いてフィードバック行列  $R, S$  を求めた。その時  $G_j$  を求めるのに必要な各特徴点のカメラ座標系における  $z$  座標の値は 4.2 節で述べたように 4 点の画像特徴点の座標から計算によって求めた。また評価関数の重み付け行列  $Q, H$  は次式のように単位行列の定数倍として設定した。

$$Q_j = \begin{pmatrix} 10^6 & & & 0 \\ & 10^6 & & \\ & & \ddots & \\ 0 & & & 10^6 \end{pmatrix} \quad (15)$$

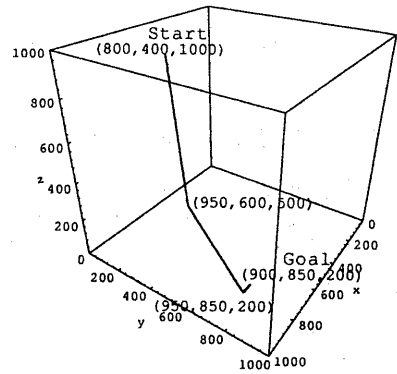


図 4: 設定した目標軌道

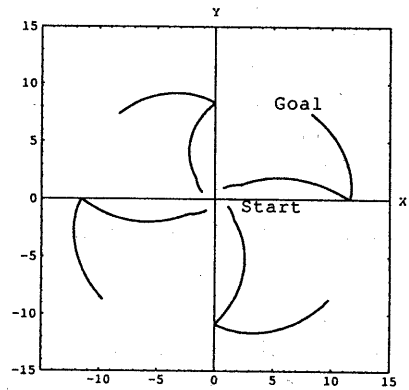


図 5: 画像特徴点の軌跡

$$H_j = \begin{pmatrix} 10^2 & & & 0 \\ & 10^2 & & \\ & & \ddots & \\ 0 & & & 10^2 \end{pmatrix} \quad (16)$$

## 5.2 カメラの制御

以上の準備の後、カメラの制御を行なった。制御時の時刻  $j$  に画像から求めた画像特徴ベクトル  $\mathbf{x}_j$  を(5)式に代入してカメラ制御ベクトル  $\mathbf{u}_j$  を求め、それによってカメラの運動を制御する実験を行なった。

### 5.2.1 実験 1: カメラの初期状態が同じ場合

まず最初に制御時のカメラの初期状態(位置及び姿勢)が目標軌道のカメラの初期状態に等しい場合の実験を行なった。すなわち時刻 0 においてカメラの位

置を(800, 400, 1000)、カメラの視方向は直方体の上面の中心を向いているものとした。実験の結果を図6から図8に示す。図6はカメラの運動が制御された結果の軌道であり、図7はその時の画像特徴点の軌跡である。いずれも目標として与えたものと一致していることがわかる。目標軌道と制御した結果の軌道との各時刻における距離を誤差として表したものを図8に示す。 $j = 100, j = 180$ 以外では小さな値になっている。 $j = 100, j = 180$ で誤差が大きくなる理由は、 $j = 100, j = 180$ では目標軌道が急激に変化している(折れ曲がっている)が、カメラ制御ベクトル $u$ が大きくなるように制御が行なわれる結果、目標軌道から外れてしまうためである。

### 5.2.2 実験2: カメラの初期状態が異なる場合

次に制御時のカメラの初期状態が目標軌道のカメラの初期状態と異なる場合の実験を行なった。すなわち時刻0においてカメラの位置を(850, 350, 950)、またカメラの視方向は直方体の上面の中心から(50, -50, 0)ずれた点を向いているものと設定した。実験の結果を図9から図11に示す。図9はカメラの運動が制御された結果の軌道であり、図10は画像特徴点の軌跡である。制御開始後カメラは目標軌道に収束するように制御されていることがわかる。目標軌道と制御した結果の軌道との各時刻における誤差を図11に示す。制御開始直後は誤差が大きいが時間が経過するにつれて0に近づいていくことがわかる。 $j = 100, j = 180$ で誤差が大きくなるのは実験1と同じ理由による。

## 5.3 考察

4.2節で述べたようにフィードバック行列を求める時にはカメラが目標軌道通りに運動するものと仮定して $G_j$ を与えた。そのため実験1のように目標軌道と制御した結果の軌道が一致する場合は評価関数を最小にするような制御が実現できていると考えられる。

しかし実験2の制御開始直後のように制御軌道が目標軌道から外れている場合、フィードバック行列を求める時の $G_j$ と制御時の実際の $G_j$ が異なるため、制御された結果は必ずしも(4)式の評価関数を最小にするものではなくなる。

実験2からわかるように、実際の軌道と目標軌道とのズレが大きくない場合は(5)式で表されるフィードバック制御の性質から目標軌道に収束するように

制御を実現することができるが、ズレが大きくなると制御が不安定になる可能性も考えられる。制御の安定性の解析的な検討は今後の課題である。

## 6 結論

本報告は対象物体を中心とする座標系で設定された目標軌道を追従するようにカメラの運動を制御する方法について述べた。目標軌道に沿ってカメラが運動した時に撮影される時空間画像から抽出した画像特徴点の軌跡を制御のための目標軌道の表現として用い、制御時にはカメラが撮影する画像から抽出した画像特徴点が目標の画像特徴点の軌跡に追従するようにカメラの運動を制御する方法を提案した。そしてシミュレーション実験によって本提案手法の有効性を示した。

本手法により対象の位置、姿勢が変動しても、またカメラの初期位置、姿勢が異なっても、対象に対して設定された目標軌道をカメラが追従することが可能になる。

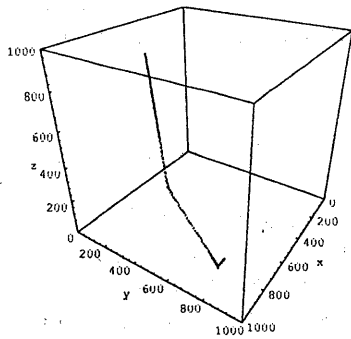
今後の課題としては、1. 評価関数((4)式)の重み付け行列( $Q, H$ )の決定法、2. 画像特徴点が欠落した場合にも安定に制御を行なうための方策、3. 画像特徴点以外の種々の画像特徴(エッジ、濃淡値等)を用いた制御の方法について検討していく予定である。

## 謝辞

本研究の機会を与えて下さいました酒井高志知能ロボット研究部長及び奥平雅士グループリーダーに感謝致します。また議論に加わっていただいた知能ロボット研究部の皆様に感謝致します。

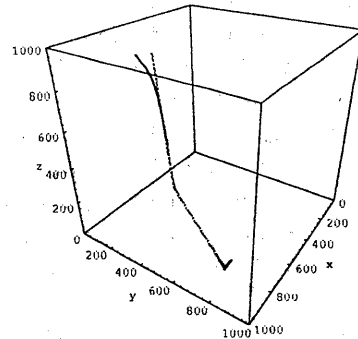
## 参考文献

- [1] F. Chaumette, P. Rives and B. Espiau, "Positioning of a Robot with respect to an Object, Tracking it and Estimating its Velocity by Visual Servoing", Proc. of the 1991 IEEE Intl. Conf. on RA, pp.2248-2253.
- [2] K. Hashimoto, T. Kimoto, T. Ebine and H. Kimura, "Manipulator Control with Image-Based Visual Servo", Proc. of the 1991 IEEE Intl. Conf. on RA, pp.2267-2272.
- [3] Julius T. Tou 著, 中村嘉平, 伊藤正美, 松尾強 共訳, "現代制御理論", コロナ社.



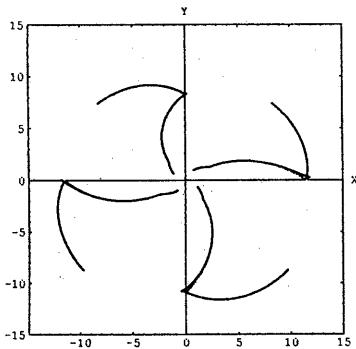
..... 目標軌道 ——— 実際の軌道

図 6: 実験 1: カメラの軌道



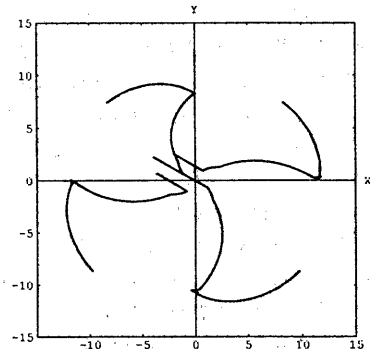
..... 目標軌道 ——— 実際の軌道

図 9: 実験 2: カメラの軌道



..... 目標軌道 ——— 実際の軌道

図 7: 実験 1: 画像特徴点の軌跡



..... 目標軌道 ——— 実際の軌道

図 10: 実験 2: 画像特徴点の軌跡

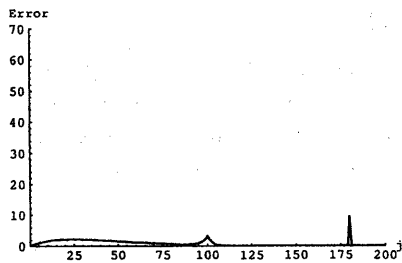


図 8: 実験 1: 制御軌道と目標軌道との誤差

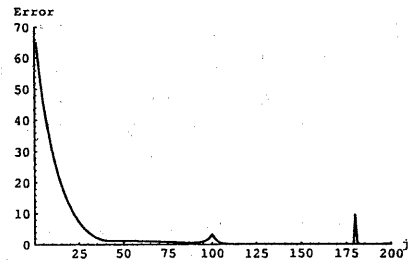


図 11: 実験 2: 制御軌道と目標軌道との誤差