

## 仮想環境と実環境の重ね合わせの一手法

大山 英明\*<sup>1</sup> 常本 直貴\*<sup>1</sup> 舘 暲\*<sup>2</sup> 井上 康之\*<sup>3</sup>

### A Method for Superimposing Virtual Environment and Real Environment

Eimei Oyama\*<sup>1</sup>, Naoki Tunemoto\*<sup>1</sup>, Susumu Tachi\*<sup>2</sup> and Yasuyuki Inoue\*<sup>3</sup>

Tele-existence is an advanced type of teleoperation system that enables a human operator to perform remote manipulation tasks dexterously with the feeling that he or she exists in the remote anthropomorphic robot in the remote environment. In order to extend the function of a human, an extended tele-existence system with virtual reality technology was proposed. For example, when visual information cannot be used because of smoke due to fire, then a human operator still can see the virtual environment of the scene and can control the robot. The model of the environment can be constructed from the design data of the environment. Usually, there are errors in the model of the environment. Large errors in parameters may fail operations using the virtual environment. In order to compensate for the errors, the calibration technique using model based image measurement is proposed for matching the real image and the virtual image. After the calibration, an experimental operation in the almost invisible environment was successfully conducted.

**Key Words:** Virtual Reality, Tele-Existence, Virtual Environment, Poor Visibility Environment, Superimpose

#### 1. はじめに

テレイグジスタンスは、人間があたかもロボットになったかのような臨場感をもって、遠隔地のロボットを操作することにより、ロボットに人間のような高度で柔軟な作業を行わせることを目指している[1][2][3].

著者らは、人間の柔軟な問題解決能力を遠隔地のロボットに実現するばかりでなく、ロボットの持つセンサーや計算機の助けを借りて、仮想環境[4]を構成し、人間の機能の拡張を行うことを提案している[5][6]. 例えば、煙の中で作業を行う場合に、人間には見ることでできない環境情報をコンピュータグラフィクスによって生成し、人間がその仮想環境中で作業を行うようにすれば、煙の中で目が見えるかのごとく、ロボットを操ることが可能になる。

この仮想環境を組み込んだ拡張テレイグジスタンスによる人間機能の拡張の効果を実証するため、テレイグジスタンス実験システム[7][8]に人工現実感システムを組み込み、拡張型テレイグジスタンスシステムを構成した[9]. このシステムを用

いて臨場感の高い不可視情報の提示を行うことにより、不可視環境下で作業を実行するためには、不可視情報を提示する仮想環境と実環境の整合性を保つ必要がある。特に接触作業の実行は、両者が十分に一致していなければ困難である。

本稿では、拡張型テレイグジスタンスシステムの仮想環境を構成する環境モデルのパラメータ誤差修正法を示す[10][11]. さらに、環境モデル修正の後、煙による不可視環境下において、レバーを回す作業実験を行い、仮想環境を用いて不可視情報の提示を行うことにより、不可視環境下での作業が可能となることを示す。

#### 2. 不可視情報提示のための拡張型テレイグジスタンス

著者らは、仮想環境[4]をテレイグジスタンスシステムに組み込み、人間の機能の拡張を行うことを提案している[5][6]. 仮想環境の利用により、作業訓練、不可視情報提示、予測環境の利用による時間遅れの克服等が期待できる。

不可視情報提示による不可視環境下での作業は、仮想環境の有効な利用法の1つである。例えば、工場における事故現場や火災現場においては、煙等によって視覚情報を得ることが難しくなり、オペレータの作業能力を十分発揮できなくなる。このような場合、作業現場の環境モデルによって仮想環境を構成し、ロボットの視覚センサからは得ることのできない不可視情報をオペレータに提示する。仮想環境中で作業を行うオペレータの動作に従ってスレーブロボットを動かすことを考える。仮

原稿受付 1992年12月28日

\*<sup>1</sup>機械技術研究所

\*<sup>2</sup>東京大学先端科学技術研究センター

\*<sup>3</sup>安川電機製作所株式会社

\*<sup>1</sup>Mechanical Engineering Laboratory

\*<sup>2</sup>RCAST, University of Tokyo

\*<sup>3</sup>Yasukawa Electric Corporation

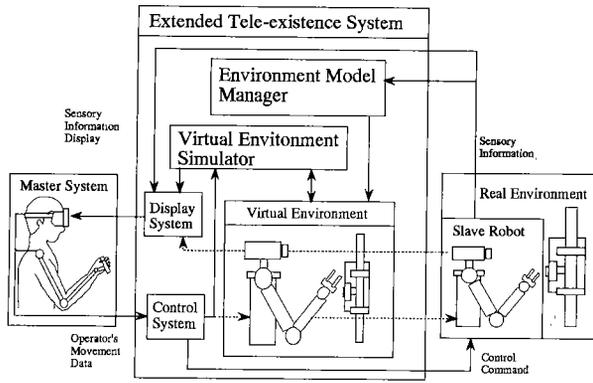


Fig. 1 Extended tele-existence system with virtual reality

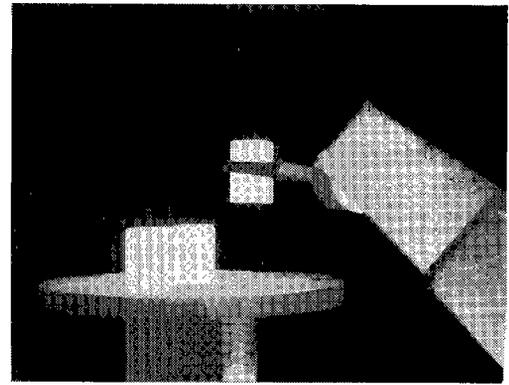


Fig. 2 Block stacking in virtual reality

想環境が実環境を十分良く近似しているならば、オペレータは煙の中で目が見えるかのごとく、ロボットを操ることができる。

仮想環境を組み込んだ拡張型トレイグジスタンスシステムの概念図を Fig. 1 に示す。拡張型トレイグジスタンスシステムは仮想環境シミュレータと環境モデルマネージャを持つ。仮想環境シミュレータは、オペレータの動作と物理法則から、仮想環境を更新する。環境モデルマネージャは、スレーブロボットからのセンサ情報を統合して、仮想環境を実環境に近づける。拡張型トレイグジスタンスシステムでは、オペレータは仮想環境中に存在する仮想ロボットに接続される。仮想環境シミュレータと環境モデルマネージャにより、仮想環境は実環境を近似しており、オペレータは実環境中のスレーブロボットに間接的に接続される。仮想環境の近似精度が高ければ、オペレータは実環境中で作業を行っているかのような現実感で、仮想環境において作業を行うことが可能であり、同時にスレーブロボットを操作することが可能である。

ロボットの作業場には、工場内のように、設計データから環境モデルの作成を比較的容易に行うことができる場所も多い。このような場所では設計データを基にした環境モデルの作成が可能である。しかしながら、一般に環境モデルは完全ではなく、環境中に存在する物体の形状や位置、姿勢等には誤差が存在する。そのため実環境における物体と仮想環境における物体、あるいは実環境におけるスレーブアームと仮想環境におけるスレーブアームには、ずれが生じる。実環境からの視覚情報を利用する場合、物体やアームのずれは、操縦者の視覚フィードバックによって補正されるため、スレーブアームの操縦に慣れるまでの時間が余分にかかることを除けば、ほとんど問題とはならないが、仮想環境の情報のみを用いて作業を行う場合、このようなずれは、作業に大きな影響を与える。見えない部分を仮想環境によって可視化して作業を行うためには、仮想環境と実環境とを可能な限り近づけ、整合性を保つことが不可欠である。そのため環境モデルマネージャは、常時、スレーブロボットからのセンサ情報より、環境モデルの誤差を修正し、仮想環境と実環境の整合性を保つ。

なお、スレーブロボットのセンサ情報により環境モデルの修正を行っても、実環境と仮想環境を完全に一致させることは困

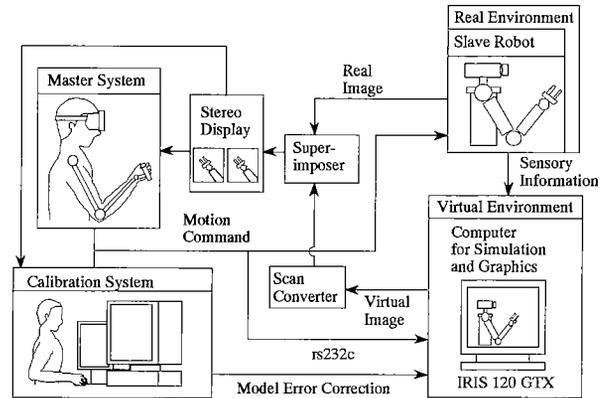


Fig. 3 System configuration of experimental extended tele-existence system

難である。スレーブロボットが、仮想環境においてオペレータが行った動作をそのまま行うのではなく、オペレータの動作を基準にしつつ、オペレータの動作の目的を達成するような自律性がスレーブロボットに必要な。

著者らはトレイグジスタンスによるロボット制御の有効性を実証するために作成したトレイグジスタンス実験システム [7] [8] に、作業訓練用の人工現実感システムを組み込み、拡張型トレイグジスタンス実験システムを構成した [9]。Fig. 2 に、仮想環境中で積み木作業の訓練を行っている様子を示す。拡張型トレイグジスタンス実験システムの具体的な構成を Fig. 3 に示す。拡張型トレイグジスタンス実験システムには、不可視環境下での作業を行うために、画像情報を利用したキャリブレーションシステムを組み込んでいる [10] [11]。著者らは、Fig. 1 のシステムを実現することを目指して、拡張型トレイグジスタンス実験システムを構成したが、現時点において、実験システムは未完成であり、環境モデルマネージャを実現できていない。環境モデルマネージャの機能の一部をキャリブレーションシステムとして実現している。このキャリブレーションシステムは画像情報を利用しているため、不可視環境下での作業中に環境モデルの修正を行うことは不可能であるが、作業前のキャリブレーションには利用できる。次章では、このキャリブレ

ーションシステムの構成について具体的に述べる。

キャリブレーション後に残るモデル化誤差の問題に対して、拡張型テレイグジスタンス実験システムでは、スレーブロボットの制御に、作業座標において設定した機械的インピーダンス特性を実現するインピーダンス制御を用いて、対象物体に位置誤差がある場合にも接触作業を可能にし、この問題を部分的に解決している[12]。

### 3. キャリブレーションシステム

人工現実感システムは、オペレータの視覚情報だけでなく、触覚や力感覚の情報を提示することによって、より高度の現実感を持った仮想環境を創出することが可能になるが、現在の拡張型テレイグジスタンス実験システムでは、専ら視覚情報の提示によって、仮想環境を創出する。よって、実環境画像と仮想環境画像とをびったりと重ね合わせることに、実環境と仮想環境を重ね合わせることが可能である。

実環境と仮想環境がずれる主たる原因は、仮想環境を構成する環境モデルにモデル化誤差があるためである。この誤差を、モデルの構造に起因するものと、モデルを構成する物体の形状や位置、姿勢といったパラメータの数値誤差に起因するものに分けて考える。モデルの構造に起因する誤差とは、例えば、立方体を四面体でモデル化してしまった場合が挙げられる。このような場合、モデル化誤差の修正を、センサ情報から自動的に行うことは困難である。人間が仮想環境のモデルを再定義することによって、解決を図らざるを得ない。一方、環境モデルのパラメータ誤差に起因する誤差は、適切なパラメータの推定法を用いることによって、自動的に修正できる。本章では、環境モデルのパラメータ誤差を推定するキャリブレーションシステムの構成について述べる。

#### 3.1 モデルベースト画像計測システム

筆者らは、対象物体の画像と対象物体の形状モデルとの対応から、非線形最小二乗法を用いて物体の姿勢や位置といった未知パラメータの推定を行うモデルベースト画像計測システムを作成した[11]。このシステムは、オペレータの補助によって、観測を行い、観測ベクトルを構成する。環境モデルはパラメータを用いた形状モデル記述言語によって記述され、その記述に従って、システムは観測方程式を作成し、非線形最小二乗法によって観測方程式を解き、自動的にパラメータの値を推定する。このモデルベースト画像計測システムを中心とするキャリブレーションシステムをテレイグジスタンス実験システムに組み込み、パラメータ誤差の推定を行い、実画像と仮想環境画像の重ね合わせを行う。このキャリブレーションシステムの概念図を Fig. 4 に示す。

特に、人工現実感システムに組み込むに当たり、モデルベースト画像計測システムのモデリング機能を用いて、形状モデル記述言語で記述した環境モデルを仮想環境の描画プログラムにコンパイルして、高速な描画を可能にし、また、その描画プログラムをそのまま用いて、観測方程式の観測モデルを計算し、数値微分を用いて、非線形最小二乗法で用いるヤコビ行列を計算するようにした。それにより、以下のような特徴がある。

(1) 環境モデルの記述に基づいて、パラメータ値の修推定

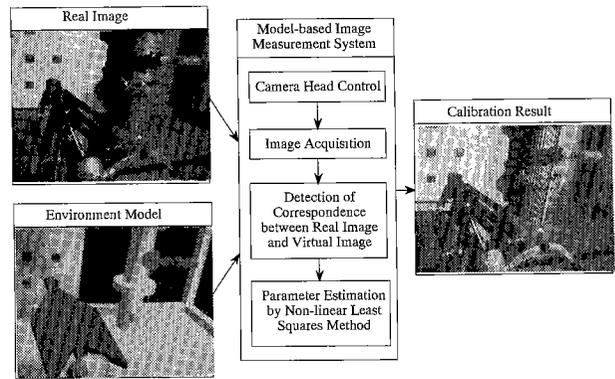


Fig. 4 Model based image measurement system

が可能である。

- (2) 観測方程式に可観測性があれば、任意のパラメータの値の推定が可能である。
- (3) 観測モデルを数式モデルで表すことのできる観測量だけでなく、コンピュータプログラムで、値を計算できる観測量であれば、コンピュータプログラムを用意することにより、任意の観測量を利用できる。

人工現実感システムのキャリブレーションにおいては、環境モデルのパラメータの修正を、環境モデルの記述に基づいて行う必要がある。例えば、ある回転軸の回りに回転するレバーの想定した座標系における位置、姿勢は、適当なセンサと推定プログラムによって得ることができ、人工現実感を創出するためのシミュレーションに必要であるのは、レバーの回転軸回りの回転角であり、計測は、最終的には、回転角を推定するものでなければならない。また、人工現実感システムの環境モデルのパラメータには、物体の寸法誤差、位置誤差、姿勢誤差等、色々な種類があり、提案した手法は、環境モデルのパラメータ誤差修正に非常に適している。

現時点においては、Fig. 3 に示すように、スレーブロボットのエペレータとは別のオペレータが、キャリブレーションを行っている。画像を処理して、自動的にキャリブレーションを行うことが望ましいが、画像から自動的にパラメータを推定するには、画像上の特徴点と物体の形状モデル上の点とを対応づける画像認識処理が必要となり、現時点において、十分信頼性の高い画像認識プログラムの開発に成功しておらず、そのため、対応付けは、オペレータの認識能力に依存している。

#### 3.2 未知パラメータの推定法

画像によって観測される物体上の点の位置や線の傾き、線の位置等によって構成される観測ベクトル  $y$  を以下のようにモデル化する。

$$y = f(p, x) + \omega \quad (1)$$

観測量ベクトル  $y$  は、既知のパラメータベクトル  $p$ 、未知パラメータベクトル  $x$  によって構成される観測量のモデル  $f(p, x)$  と、それに加わる観測ノイズ  $\omega$  の和によってモデル化することができる。

観測ノイズが正規分布に従うものと仮定する。ノイズの共分散

散行列を

$$\Sigma_y = E(\omega\omega^T) \quad (2)$$

とする。\$E(A)\$は行列\$A\$の期待値である。観測量の重み行列を次式のように定義し、

$$W = \Sigma_y^{-1} \quad (3)$$

評価関数

$$S = (y - f(p, x))^T W (y - f(p, x)) \quad (4)$$

を最小化することにより、観測ノイズが正規分布に従うという仮定が成立すれば、未知パラメータ \$x\$ の値の最適推定を行うことができる。厳密にはこの仮定は成立しない場合も多いが、その場合でも最尤推定値に近い解を得られることが多い。このような形式の評価関数の最小化問題の解法は非線形最小二乗法と呼ばれている。

観測モデルのヤコビ行列を \$H\$ とおく。\$H\$ は次式のように定義される。

$$H = \partial f(p, x) / \partial x \quad (5)$$

\$x\$ の値を多様に反映する十分な個数の観測量があれば、真の解の付近でヤコビ行列 \$H\$ はフルランクとなり、未知パラメータベクトル \$x\$ の推定が可能となる。

代表的な非線形最小二乗法である Marquardt 法 [13] の反復改良演算は

$$x_{n+1} = x_n + (H^T W H + \lambda I)^{-1} H^T W (y - f(x_n)) \quad (6)$$

のように \$x\$ の値を反復改良する。\$\lambda\$ は非線形性の度合いに応じて調節するスカラーパラメータである。収束の終盤においては、システムは通常線形に近くなり、その場合 \$\lambda\$ は減少傾向にある。収束時、\$\lambda\$ が 0 で無い場合には、\$\lambda\$ を 0 として式 (6) を計算する。\$H^T W H\$ の逆行列が計算できない場合には、\$H\$ がフルランクでなく、この観測方程式による \$x\$ の推定は不可能とみなせる。

### 3.3 画像の観測モデル

仮想環境画像と実画像との重ね合わせを行う場合において、重要なことは、色や光沢、明暗といった画素の性質ではなく、画像を構成する点や線が重なることである。よって、本システムでは、画像を構成する点の位置や直線の傾き、切片等を観測量として観測方程式を定義する。本節では、最も単純な画面上の点の位置の観測モデルについて述べる。

物体上の点の画面上の位置の観測モデルを定義するために、3次元ベクトル

$$X = (x, y, z)^T \quad (7)$$

を用いて、以下のように代表的な変換をベクトル関数として定義する。

(a) スケーリング変換

$$scale(s, X) = (s_x x, s_y y, s_z z)^T \quad (8)$$

\$s = (s\_x, s\_y, s\_z)^T\$ : スケーリングベクトル

(b) 並進変換

$$trans(X_p, X) = X + X_p \quad (9)$$

\$X\_p = (x\_p, y\_p, z\_p)^T\$ : 並進を定義する3次元ベクトル

(c) 回転変換

$$rotate(\theta, X) = C(\theta) X \quad (10)$$

\$\theta = (\phi, \theta, \psi)^T\$ : 回転を定義するオイラー角ベクトル

\$C(\theta)\$ : 3行3列の回転変換行列 (オイラー角の三角関数の積和によって表される。)

さらに、並進並びに回転変換において逆変換を考える。

(b)' 逆並進変換

$$trans^{-1}(X_p, X) = trans(-X_p, X) \quad (11)$$

(c)' 逆回転変換

$$rotate^{-1}(\theta, X) = C(\theta)^T X \quad (12)$$

物体上の点の3次元の位置ベクトルは、適当な基本ベクトルに、スケーリング変換、回転変換、並進変換を施せば、得ることができる。可動部分を持つ物体については、可動部を適当な変換に対応付け、適当なパラメータを導入することによって、定義できる。一般的な物体上の点 \$k\$ の3次元位置 \$X\_{3d}^{(k)}\$ は、適当な基本ベクトル (例えば、\$b = (1, 1, 1)^T\$) に、スケーリング変換を施し、さらに、\$m\$ 回の並進変換、並びに \$n\$ 回の回転変換を適当な順序で施すことによって、次のように得ることができる。

$$X_{3d}^{(k)} = trans(X_{p1}^{(k)}, rotate(\theta^{(k)}, trans(X_{pi}^{(k)}, rotate(\theta^{(k)}, trans(X_{pm}^{(k)}, rotate(\theta_n^{(k)}, scale(s^{(k)}, b)))))) \quad (13)$$

\$s^{(k)}, X\_{pi}^{(k)} (0 \le i \le m), \theta\_j^{(k)} (0 \le j \le n)\$ は、それぞれ、スケーリングベクトル、並進ベクトル、オイラー角ベクトルである。

仮想環境の基準となる世界座標系上における物体上の点 \$k\$ の位置 \$X\_w^{(k)}\$ は、物体の位置が \$X\_p^{(k)}\$、その姿勢がオイラー角ベクトル \$\theta^{(k)}\$ で表されるとすると、

$$X_w^{(k)} = trans(X_p^{(k)}, rotate(\theta^{(k)}, X_{3d}^{(k)})) \quad (14)$$

のように得ることができる。さらに、世界座標系とカメラ座標系の関係が、

$$X_{3d} = C(\theta_c) X_c + X_{pc} = trans(X_{pc}, rotate(\theta_c, X_c)) \quad (15)$$

であるとすると、次式を得る。

$$X_c^{(k)} = (x_c^{(k)}, y_c^{(k)}, z_c^{(k)})$$

$$\begin{aligned}
&= C(\Theta_c)^{-1}(X_w^{(k)} - X_{pc}) \\
&= \text{rotate}^{-1}(\Theta_c, \\
&\quad \text{trans}^{-1}(X_{pc}, X_w^{(k)})) \quad (16)
\end{aligned}$$

カメラ座標系の  $z$  軸が<sup>3</sup>、カメラの視線方向と一致しており、画面座標系の  $x$  軸が、カメラ座標系の  $x$  軸に、画面座標系の  $y$  軸が、カメラ座標系の  $y$  軸にそれぞれ一致しているものとするれば、カメラ座標系の  $z=K$  に存在する画面座標系上への透視変換は次のように定義できる。

(d) 透視変換

$$\begin{aligned}
\text{perspec}(X) &= (x_i, y_i)^T \\
&= (Kx/z, Ky/z) \quad (17)
\end{aligned}$$

実画面上で観測した点  $k$  の位置  $X_k^{(k)} = (x_k^{(k)}, y_k^{(k)})^T$  を観測量とすると、点  $k$  の画面上の位置の観測モデル  $X_k^{(p)}$  は、式 (16) で定義される  $X_c^{(k)}$  を用いて、以下のように定義できる。

$$\begin{aligned}
X_k^{(p)} &= (x_k^{(p)}, y_k^{(p)})^T \\
&= \text{perspec}(X_c^{(k)}) \quad (18)
\end{aligned}$$

$v$  個の実画面上の点について、観測を行うこととすれば、 $2v$  個の要素を持つ観測ベクトル  $y$  を以下のように定義できる。

$$y = (X_k^{(1)T}, \dots, X_k^{(v)T})^T \quad (19)$$

これに対応する観測モデルは式 (18) で定義される  $X_k^{(p)}$  を用いて、以下のように定義できる。

$$f(p, x) = (X_k^{(p)T}, \dots, X_k^{(v)T})^T \quad (20)$$

これまでに用いられたパラメータをまとめて、以下のような点  $k$  に関するパラメータベクトルを作る。

$$\begin{aligned}
P^{(k)} &= (s^{(k)T}, X_{pc}^{(k)T}, \dots, X_{pm}^{(k)T}, \Theta_1^{(k)T}, \dots, \Theta_n^{(k)T}, \\
&\quad X_p^{(k)T}, \Theta^{(k)T})^T \quad (21)
\end{aligned}$$

さらに、観測した  $v$  個の点について、パラメータベクトルをまとめて、最終的に以下のようなパラメータベクトルを作る。

$$P = (x_{pc}^T, \Theta^T, P^{(1)T}, \dots, P^{(v)T})^T \quad (22)$$

$P$  の要素のうち、推定したいパラメータをまとめて、未知パラメータベクトル  $x$  とし、残りを既知パラメータベクトル  $p$  とする。逆に  $x$  と  $p$  から  $P$  を構成するベクトル関数を  $\text{param}(p, x)$  とすれば、

$$P = \text{param}(p, x) \quad (23)$$

として、パラメータの値を設定し、式 (13) (14) (16) (18) (20) の定義を用いて、観測モデル  $f(p, x)$  の値を計算できる。

### 3.4 インプリメンテーション

本節では、キャリブレーションシステムの各処理が、具体的にどのように実現されているか述べる。

キャリブレーションシステムはスレーブロボットに指令を送って適切な位置、姿勢をとらせ、スレーブロボットに登載されているビデオカメラからの画像を取り込む。次に、オペレータの指示に従って、カメラからの実画像と仮想環境画像との間で対応付けを行い、画像の観測方程式を構成する。画像上の点の位置の取得においては、拡張テイレイグジスタンスシステムの画像提示システムの特徴を有効に利用している。Fig. 3 に示すように拡張テイレイグジスタンス実験システムの画像提示システムは、スレーブロボットからの実画像とコンピュータによって生成した仮想環境画像の両方をスーパーインポザーによって切り替えて表示でき、また、実画像とコンピュータグラフィクスを重ねて同時に表示することができる。コンピュータグラフ

イクスの RGB 各色の明るさを 0 から 1 で表すものとする、スーパーインポザーは、コンピュータグラフィクスで生成した画像において (R, G, B) = (0, 0, 1) となる画素では実画像を表示しそれ以外の画素では仮想環境画像をそのまま表示するよう設定している。実画像を表示するためにグラフィクス画面を青色にしたとき、マウスカーソルの色を青以外の色にしていれば、マウスカーソルと実画像を同時に表示でき、マウスの位置情報を得ることにより、実画像上の点の位置  $X_i$  を計測できる。また、2 点を指定することにより、画面上の直線を指定できる。

観測方程式を構成する際において、画面上の点と形状モデル上の点とが同一のものでなければならない。画面上に見える 1 点を、計算機内の形状モデル上の点に対応付ける問題は画像認識問題である。現時点では画像認識技術は信頼性が低いため、自動的な対応付けは困難であり、人間が画面上の点に対応する仮想環境画像の物体上の点をマウスで指示することによって、対応付けを行う。実画像上の点に対応する仮想環境画像上の点をマウスで指示したとき、マウスカーソル位置にもっとも近いモデル上の頂点が選択され、実画像と環境モデルとの間で対応付けが行われる。

対応付けによって得られた観測方程式を解いて、パラメータ誤差を推定する非線形最小二乗法においては、Marquardt 法を利用した。観測モデル  $f(p, x)$  の値を計算し、さらに、数値微分によってそのヤコビ行列  $H$  を計算し、未知パラメータ  $x$  の値の推定を行う。画面上における点の位置の観測モデルは、前節で述べたように、基本ベクトルに、種々の変換を施すことによって、得ることができるが、この変換処理は、人工現実感システムの描画プログラムと同じである。本システムでは式 (20) を計算するために、仮想環境の描画プログラムをそのまま利用している。そのために、前節のパラメータ値設定関数  $\text{param}(p, x)$  を作成し、仮想環境の描画プログラムの物体の位置や姿勢などのパラメータの値を、外部から変更することができるようにした。未知パラメータ  $x$  の値を設定し、描画プログラムを動作させれば、画面上の点の位置を得ることができる。ヤコビ行列  $H$  は、 $d$  を適度に小さな実数として、ベクトル  $d^{(j)}$  を定義し、以下のような数値微分によって、計算している。

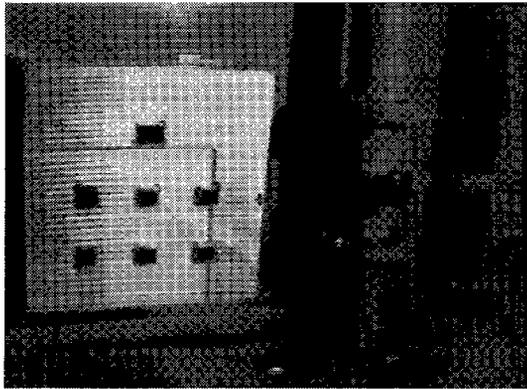
$$\begin{aligned}
H_{ij} &= (f_i(p, x + d^{(j)}) - f_i(p, x)) / d \\
d_i^{(j)} &= \begin{cases} d & (i=j) \\ 0 & (i \neq j) \end{cases} \quad (24)
\end{aligned}$$

さらに、対話的に計測を行うためにヒューマンインターフェイスを作成し、カメラに登載しているスレーブロボットの位置、姿勢、画像で計測を行う対象物体、推定するパラメータ等をマウスを用いて、対話的に指定できるようにしている。

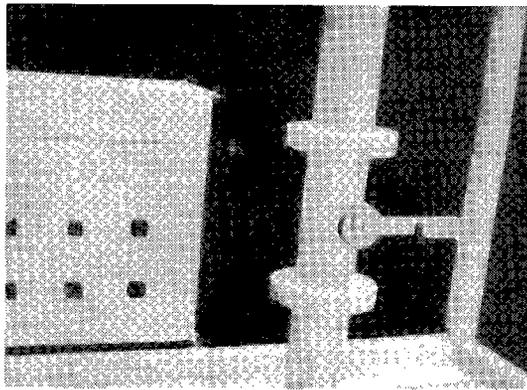
### 3.5 推定の手順

本節では、モデルベース画像計測における未知パラメータ推定のために、オペレータが行う操作手順を示す。現在の操作手順は以下の通りである。

- (1) 計測対象が画面上に来るようにカメラヘッドに指令値を送る。
- (2) 画面を実画像表示し、画面上の頂点位置をマウスで指



(a) Camera image



(b) Virtual image

Fig. 5 Real image and virtual image during correspondance detection

示ることによって、画面上の頂点の位置を計測する (Fig. 5 (a) 参照)。

- (3) 画面を仮想環境画像に切り替え、実画像上の点に対応する点をマウスで指示する (Fig. 5 (b) 参照)。
- (4) 十分な数の観測量が得られるまで、(1)から(3)の手順を繰り返す。(2)と(3)の順序は逆にしてもよい。
- (5) 推定すべきパラメータを選択する。
- (6) 十分な数の観測量が得られた後、推定アルゴリズムによって、未知パラメータの値を推定する。複数の初期値より推定を行い、最も残差の小さいものを選ぶ。十分な推定が行えなかった場合には(1)から(3)を繰り返して、十分な観測量を得る。

Fig. 5 (a) に実画像と仮想環境画像を対応付けている際の実画像を示す。Fig. 5 (b) に、仮想環境画像を示す。画像中の物体の頂点に付いている円は、実画像と仮想環境画像との間で対応付けを行った点を示している。十分な個数の対応付けが完了した後、非線形最小二乗法により推定を行う。

#### 4. キャリブレーション実験

画像情報を用いて、パラメータ誤差を推定し、実画像と仮想環境を重ね合わせるキャリブレーションの実験を行った。

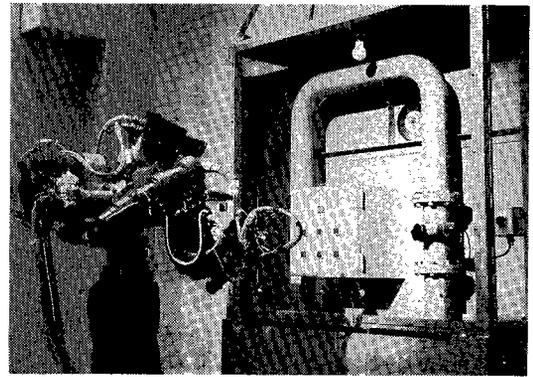


Fig. 6 Tele-existence experimental environment

#### 4.1 模擬プラント環境

著者らはテレイグジスタンスによるロボット制御の有効性を検証するためのテレイグジスタンス実験環境として、模擬プラント環境を作成した[9]。この環境は、工場内の部屋を想定し、パイプやレバー等によって構成されている。Fig. 6 にその外観を示す。この実験環境は、スモークマシンによって、煙を発生させることができ、写真の右の方に見えるレバーを回すと煙が止まるようになっている。また左の箱にあるボタンは、換気扇のスイッチになっている。

今回の実験環境は、単純な形状要素の組合せによって構成されており、計算機に形状データを入力することは比較的容易である。形状データはサーフェスモデルによって、記述されている。Fig. 5 (b) は環境モデルをコンピュータグラフィクスによって表示したものである。仮想環境においても、煙を止めるためのレバーと、換気扇を回すためのボタンを操作することができる。ボタンを押すとピープ音になるようになっている。実験環境の形状データが得られるため、推定すべきパラメータは、環境モデルの可動部のパラメータの値と環境とロボットの相対位置誤差、相対姿勢誤差である。

#### 4.2 キャリブレーションの手順

現時点において、テレイグジスタンスシステムのスレーブロボットの首とアームには、軽量化のため関節角のゼロ点がなく、停止状態でアームを動かすと、原点位置がずれ、仮想アームと実アームでずれが生じてしまうという問題点があり、首とアームの関節角オフセットも作業のために推定する必要がある。通常のロボットでは、ロボットの首やアームのオフセット等は、オフライン計測で、あらかじめ測定しておくものであるが、本システムでは、キャリブレーションによってオフセット値を推定する。

今回のキャリブレーションにおいては、既知パラメータは、ロボットの位置、姿勢、首やアームの関節角であり、観測量は、オペレータの画像上のアームや対象物の特徴点や特徴線の画面上での位置であり、推定すべきパラメータは、模擬プラント環境の位置と姿勢の誤差、スレーブロボットの首やアームの関節角オフセットである。

実画像と仮想環境画像の重ね合わせのためのキャリブレーションは、以下のように行う。

- (1) 首のキャリブレーションを壁に張ってあるマーカーを見て行う。
- (2) スレーブロボットに指令を送って、首を動かし、視界に模擬プラント環境を持ってくる。
- (3) 模擬プラント環境の位置誤差、姿勢誤差をキャリブレーションシステムにより推定する。
- (4) コンピュータより、指令を送って、首とアームを動かし、視界にアームを持ってくる。
- (5) アームにいろいろな姿勢をとらせ、画像計測によって、アームの間接角オフセットを推定する。反復計算の収束時において、観測方程式のヤコビ行列がフルランクで無い場合には、アームの姿勢を変え、新たな観測値を追加する。

(1)の首のキャリブレーションは、提案法ではなく、ロボットより約7[m]離れた壁にロボットの目の高さ、水平に張った3つのマーカーをカメラの画像中で見ながら、首の関節角をゆっくりと変えて、マーカーが計算通りの画像中の位置にくるよう調節することによって行った。正確な測定装置を持っていないため、このキャリブレーションの誤差は不明である。この誤差は、後の測定において、モデル化していない誤差源となる。

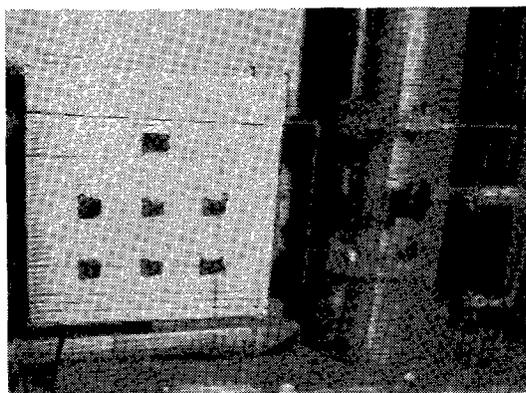
#### 4.3 キャリブレーションの諸条件

キャリブレーション後に残る誤差の評価のために、基準座標系として直交座標系を模擬プラント環境を基準に設定する。模擬プラント環境を正面に見る位置に立った場合、背中の方に  $x$  軸、水平右手方向に  $y$  軸、垂直上方向に  $z$  軸をとり、これを基準座標系と呼ぶことにする。なお、模擬プラント環境の位置・姿勢には誤差があり、キャリブレーションにおいて誤差推定を行うが、誤差を修正した後の仮想環境中の模擬プラント環境を基準として基準座標系を設定する。

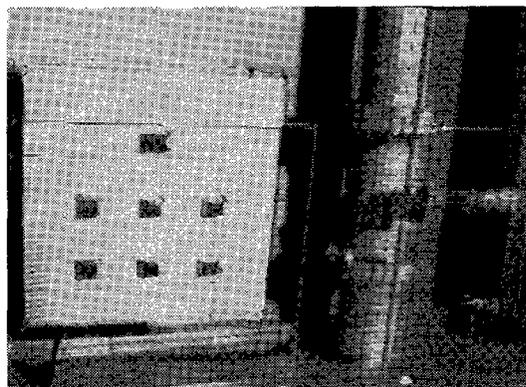
水平に6.5[cm]離れた2つのカメラの画像を用いて画像計測を行う。画像計測に用いた画像は、水平方向669画素、垂直方向453画素であり、水平方向視野角は40度である。画像の中心部の1画素は、カメラの1[m]前方では0.11[cm]に相当する。普通のオペレータは、明確に点が表示されている場合において、マウスによる通常のポインティングにより、各軸方向に対して、RMS (Root Mean Square) エラー1.5画素程度のポインティング精度を持っているが、実際の画像においては、点の位置が明確でなくなるため精度は悪化する。

#### 4.4 キャリブレーション結果

Fig. 7, 8に模擬プラント環境の位置・姿勢の計6自由度のパラメータ誤差の推定によるキャリブレーションの結果の例を示す。Fig. 7は模擬プラント環境のレバー位置より、約1.5[m]離れた地点から計測したものであり、Fig. 8は、約0.7[m]離れた地点から計測したものである。それぞれ、(a)にパラメータ誤差を推定していない状態で、実画像と仮想環境画像(ワイヤフレーム表示)を重ね合わせたものを示し、(b)に、特徴点や直線の対応付けを行い、最小二乗法によって、パラメータ誤差を推定して、重ね合わせた画像を示す。実画像と仮想環境画像が完全に重なっていないのは、モデル化していない誤差源が



(a) Before calibration



(b) After calibration

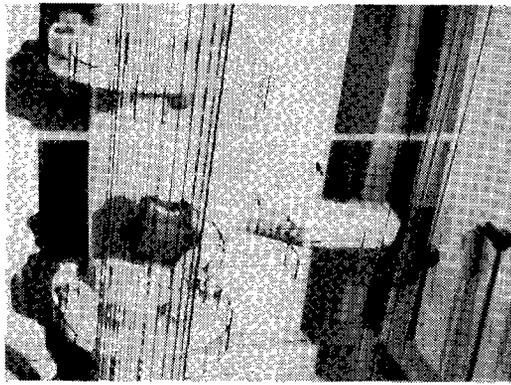
Fig. 7 Calibration of plant environment

存在しているためと考えられる。

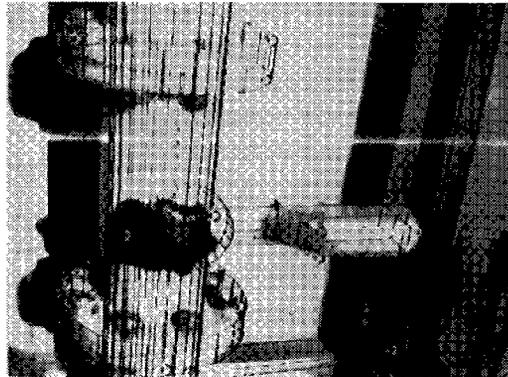
カメラが理想的に取り付けられているとすれば、観測点の画面上での位置から、その点の3次元空間上における位置を推定でき、仮想環境中での対応点との距離誤差を推定できる。Fig. 8の場合、計算の結果、レバーの握り手上に存在する8つの観測点の位置誤差の、カメラの視線方向、画像の水平方向、垂直方向の各軸方向の最大値は、順に、2.8[cm], 1.1[cm], 0.4[cm]であった。基準座標系では、 $x$  軸方向2.4[cm],  $y$  軸方向1.1[cm],  $z$  軸方向1.5[cm]であった。

同様に、アームの7つの関節角のオフセットの推定を行った。そのときの画像を Fig. 9に示す。本アームは7自由度アームであり、手先しか見えない場合には、一度の観測では、オフセットの推定ができないが、アームにいろいろな姿勢をとらせて、計測することによって、オフセットの推定が可能になる。今回は10通りの姿勢をとらせて、各姿勢において約8点を観測点とし、計測を行った。

計測の結果、ハンドの接触部である先端部に存在する6個の観測点の、カメラの視線方向、画像の水平方向、垂直方向の各軸方向の誤差の最大値は、順に、4.5[cm], 1.6[cm], 0.6[cm]であった。基準座標系での各軸の最大誤差は、ロボットがレバー一回転作業を行うときの姿勢をとった状態を想定した場合、 $x$  軸方向3.9[cm],  $y$  軸方向2.2[cm],  $z$  軸方向2.0[cm]であっ



(a) Before calibration

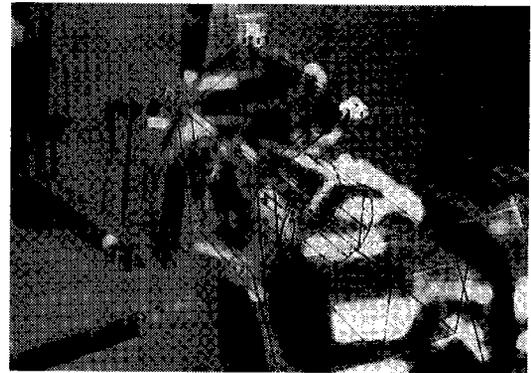


(b) After calibration

Fig. 8 Calibration of plant environment



(a) Before calibration



(b) After calibration

Fig. 9 Calibration of joint angles

Table 1 Calibration conditions and results

Estimated Parameters	Position and Attitude of Plant Environment	Offsets of Joint Angles
Number of Images	1×2(Stereo)	10×2(Stereo)
Number of Attitudes	1	10
Number of Measured Points	24	84
Number of Estimated Parameters	6	7
RMS Error on Image (Pixels) ( $x_i, y_i$ )	8.5(Lever) (8.2, 2.2)	16.5(Hand) (16.2, 4.0)
RMS Error [cm] ( $x, y, z$ )	2.0(Lever) (1.6, 0.7, 1.0)	3.8(Hand) (3.2, 1.5, 1.5)
Max. Error [cm] ( $x, y, z$ )	2.8(Lever) (2.4, 1.1, 1.5)	4.9(Hand) (3.9, 2.2, 2.0)

た。ただし、観測を行っていない姿勢では、誤差が増大した。レバー回転作業時に、実画像と仮想環境画像を重ねて表示した画像を記録し、画面上のハンド先端の誤差を計測した。これに、首とアームの姿勢から計算したカメラとハンドの間の距離情報と組み合わせて、カメラの視線に垂直な方向の位置誤差を

計算した結果、6.9[cm]に達する場合があった。

Fig. 8の模擬プラント環境の位置・姿勢推定とFig. 9のアームの関節角オフセット推定の実験の条件と結果をTable 1にまとめる。Table 1において、推定誤差は、画面座標系( $x_i, y_i$ )と基準座標系( $x, y, z$ )で示した。

### 5. 不可視環境下での作業実験

実環境と仮想環境の重ね合わせが成功したことを確認するために、不可視情報提示による作業実験を行った。まず、模擬プラント環境において、スモークマシンによって、煙を発生させ、スレーブロボットからの視覚情報が利用できない状態にする。続いて、オペレータの視覚入力を実環境から仮想環境に切り替え、仮想環境において作業を行うことにより、不可視環境下に存在するロボットを操った。レバー回転作業という比較的要求精度の低い作業の遂行に成功した。作業中のスレーブロボットの様子をFig. 10に示す。

比較のために、Fig. 11に通常の作業中のカメラ画像に仮想環境のワイヤフレームモデルを重ねて描いた画像を示す。

ハンドが上方からレバーに接触し、十分な力をレバーに加えさえすれば、レバーを回転させることができる。想定しているレバー回転作業では、仮想環境において、ハンド先端の接触部(想定している姿勢において、基準座標系のx軸方向10[cm], y軸方向1.5[cm], z軸方向10[cm]の大きさである。)の中心

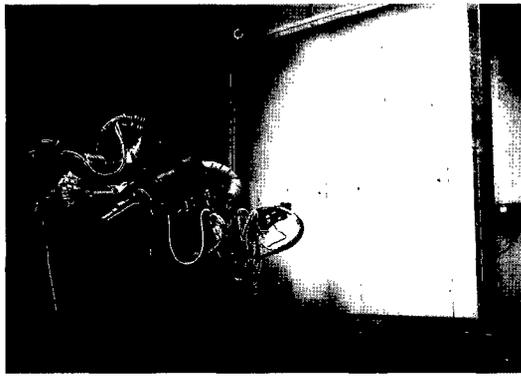
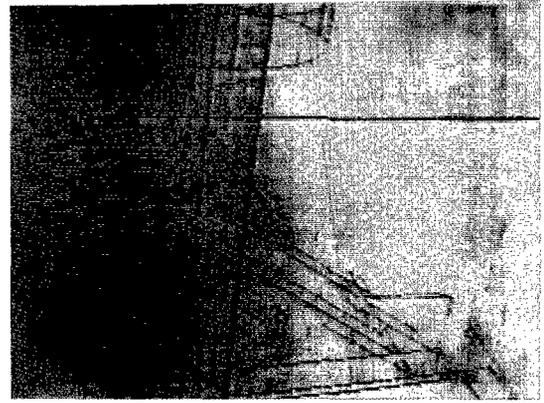


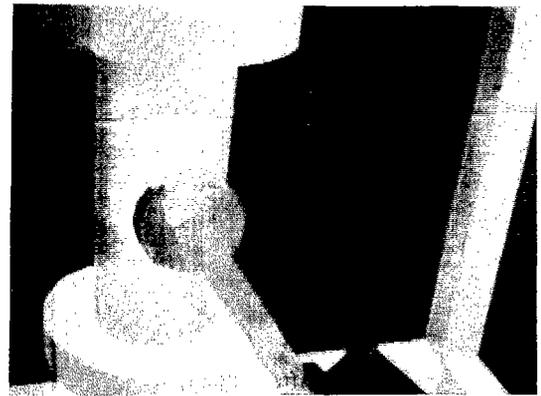
Fig. 10 Operation in almost invisible environment



(a) Real image and virtual image in almost invisible environment



Fig. 11 Operation in visible environment



(b) Virtual image

が、レバーの握り手（基準座標系の  $x$  軸方向 6[cm],  $y$  軸方向 10[cm],  $z$  軸方向 3[cm] の大きさである。）の中心に一致するようにハンドを動かす。この場合、実環境において、ハンドの接触部の中心とレバーの握り手の中心の位置誤差が、 $x$  軸方向 8[cm] 以内かつ  $y$  軸方向 6[cm] 以内であれば、ハンドはレバーに接触する。前述のキャリブレーション精度の推定値が正しいとすれば、ハンドとレバーの中心位置の誤差は、 $x$  軸方向 7[cm] 以内、 $y$  軸方向 4 cm 以内と考えられ、接触の条件を満たしている。

接触の際、インピーダンス制御系は、接触力に応じてハンド位置を変化させ、安定に回転動作を行う。Fig. 11 に示すように、キャリブレーション後も仮想環境と実環境にずれが残っているが、この程度のずれは、ハンドがレバーに接触しさえすれば、インピーダンス制御により補償可能である。

Fig. 12 に、不可視環境下での作業実験時の、ディスプレイ画像を示す。Fig. 12 の (a) は煙による不可視環境下での実際のカメラ画像に仮想環境のワイヤフレームモデルを重ねて描いたものである。実験環境のスモークマシンによる煙によって、実験環境もロボットハンドもほとんど見えない状態になる。(b) は不可視環境下での作業時に、オペレータが見ている仮想環境の画像である。(c) に作業を終えて、煙を排出したときの結果を示す。レバーの回転作業に成功していることが分かる。



(c) Result

Fig. 12 Display images of operation in almost invisible environment

## 6. おわりに

画像計測によりパラメータ誤差を推定することにより、実環境と仮想環境との重ね合わせを行った。重ね合わせが実現されていることを確かめるために、不可視環境下での作業を行い、検証実験に成功した。

現時点では、実画像と仮想環境画像との対応付けは人間が行っているため、キャリブレーションに、非常に時間がかかる。

より精度の高い重ね合わせを実現するためには推定すべき変数を増やすことが必要であり、必要な観測回数はさらに増大する。現在、マーカーの利用による対応付けの自動化を検討中であり、また、既知情報を基に、画像認識を行って、自動的に対応付けを行うことも検討している。

火災が起こった場合等、形状そのものが、完全に変わってしまい、環境モデルのパラメータを修正するだけでは、十分な近似を行うことができない可能性もある。3次元の環境モデルの形状データを得るためには、距離センサが必要である。パラメータでは表現できないモデル化誤差や、破壊による物体形状の構造的変化等の距離センサによる修正は今後の課題である。

謝 辞

本研究において、数多くの助言を下された工業技術院機械技術研究所ロボット工学部感覚制御課課長小森谷清殿に深謝致します。

参 考 文 献

[1] 館, 谷江, 小森谷: “感覚情報提示機能をもったマニピュレータの操縦方式”, 特許 1348263 号, 出願 1981 年 1 月 14 日。  
 [2] 館, 阿部: “テレイグジスタンスの研究第 1 報”, 第 21 回 SICE 学術講演会予稿集, pp. 167-168, 1982。  
 [3] 館, 荒井: “テレイグジスタンスにおける視覚情報提示系の設計と評

価”, 日本ロボット学会誌, vol. 7, no. 4, pp. 314-326, 1989。  
 [4] S. S. Fisher et al.: “Virtual environment display systems,” ACM 1986 Workshop on Interactive 3D Graphics, pp. 1-11, 1986。  
 [5] S. Tachi, H. Arai and T. Maeda: “Tele-existence simulator with artificial reality,” Proceedings of IEEE International Workshop on Intelligent Robotics and System (IROS' 88), pp. 719-724, 1988。  
 [6] 館: “テレロボティクスとテレイグジスタンス”, 計測と制御, vol. 28, no. 12, pp. 1059-1064, 1989。  
 [7] S. Tachi, H. Arai and T. Maeda: “Development of anthropomorphic tele-existence slave robot,” Proceedings of International Conference on Advanced Mechatronics, pp. 385-390, 1989。  
 [8] S. Tachi, H. Arai and T. Maeda: “Tele-existence master slave system for remote manipulation,” IEEE International Workshop on Intelligent Robotics and Systems (IROS' 90), pp. 343-348, 1990。  
 [9] 常本, 館他: “テレイグジスタンスの研究 第 25 報 —作業用仮想環境の構築—”, 日本ロボット学会第 8 回学術講演会予稿集, no. 2, pp. 565-566, 1990。  
 [10] 大山, 館他: “テレイグジスタンスの研究 第 26 報 —仮想環境と実環境の整合—”, 日本ロボット学会第 8 回学術講演会予稿集, no. 2, pp. 567-568, 1990。  
 [11] 大山, 館: “モデルベース画像計測システム”, 機械学会論文集(C), vol. 56, no. 521, pp. 109-115, 1990。  
 [12] 井上, 館: “マスタースレーブ・マニピュレータのインピーダンス制御の一手法”, 日本ロボット学会誌, vol. 10, no. 4, pp. 490-500, 1992。  
 [13] D. W. Marquardt: An algorithm for least square estimation of nonlinear parameters”, Journal of Society of Industrial Applied Mathematics, vol. 11, 1963。



大山英明 (Eimei Oyama)

1962 年 10 月 23 日生。1987 年 3 月東京大学大学院工学系研究科修士課程修了。同年 4 月通産省工業技術院機械技術研究所入所。ロボット工学部バイオロボティクス課所属。テレイグジスタンス, 人間の運動制御のモデル化, 神経回路, 非線形最適化等の研究に従事。電子情報通信学会, 計測自動制御学会, 日本ロボット学会等の会員。(日本ロボット学会正会員)



常本 直貴 (Naoki TSUNEMOTO)

1960 年 11 月 21 日生。1988 年 3 月東京大学大学院理学系研究科修士課程修了。同年 4 月通産省工業技術院機械技術研究所入所。物理情報部バイオメカニクス課所属。ヒューマン・インターフェースの研究に従事。高分子学会, 日本生物物理学会等の会員。(日本ロボット学会正会員)



館 暲 (Susumu Tachi)

1946 年 1 月 1 日。1973 年東京大学大学院工学系研究科博士課程修了, 工学博士, 東京大学助手。1975 年通産省工業技術院機械技術研究所入所, バイオロボティクス課長。1979 年より 1980 年, マサチューセッツ工科大学客員研究員。1989 年, 東京大学助教授。1992 年, 東京大学教授。国際計測連合学会 (IMEKO) ロボティクス会議議長。(日本ロボット学会正会員)



井上康之 (Yasuyuki Inoue)

1958 年 4 月 27 日生。1982 年早稲田大学理工学部電気工学科卒業。同年(株)安川電機入社。1989 年から 90 年通産省工業技術院機械技術研究所研修員。現在安川電機つくば研究所在籍, 電動機制御, マニピュレータ制御などの研究に従事。電気学。(日本ロボット学会正会員)