

# 既存 GUI アプリケーションによる没入 3 次元環境の補強

立山 義祐<sup>†</sup> 小木 哲朗<sup>†,††</sup> 廣瀬 通孝<sup>††</sup>

CAVE に代表される没入型多面ディスプレイの広視野角の立体視映像により、実在感の高い 3 次元形状を持つ仮想物体の提示が可能となった。それと同時に 3 次元位置姿勢情報をもとにう入力装置による、仮想 3 次元物体の操作も可能となった。この没入 3 次元環境を使って、主に 2 次元ディスプレイ・マウス・キーボードで構成される既存の 2 次元環境では実現が困難だった 3 次元アプリケーションの研究開発が行われている。一方我々は、2 次元環境で使用可能な GUI アプリケーションを、手放し難い道具として日常的に活用している。3 次元アプリケーションとともに、既存 GUI アプリケーションが操作可能になれば、没入 3 次元環境の実用性は、より高まると考えられる。そこで我々は、没入 3 次元アプリケーションとともに既存 GUI アプリケーションが操作可能なシステム *pwm* を試作した。この *pwm* 実装では没入型多面ディスプレイで通常用いられる装置のみの使用を目指した。さらに、この *pwm* 実装での 2 次元 GUI 操作の操作性の検証のため、2 次元ポインティングにかかる時間を 5 人の被験者で計測した。2 次元環境のマウスに比べ 55% 程度の操作性低下が見られるものの、操作機能の実現を確認できた。また、3 次元アプリケーションを活用する作業が *pwm* によって支援される例として、ゲノム情報分析ツールとの統合を行い、本システムの有用性を確認した。

## Reinforcement of an Immersive 3D Environment with Existing GUI Applications

YOSHISUKE TATEYAMA,<sup>†</sup> TETSURO OGI<sup>†,††</sup> and MICHITAKA HIROSE<sup>††</sup>

Immersive multiscreen displays like CAVE create a sense of realism for 3D objects. The display user can also manipulate objects with a 6 DoF controller. Specific 3D applications have been developed for the immersive 3D environment that cannot work in a “normal” 2D GUI environment, which usually consists of a 2D display, a mouse and a keyboard. However, the 2D GUI environment has numerous useful applications that we have come to consider as indispensable tools. If a user in an immersive 3D environment could operate existing 2D GUI applications, then the immersive 3D environment would be more useful. This paper describes a prototype system named the *pwm* that enables a user to operate 2D GUI applications in an immersive 3D environment. We implement it without any additional equipment not usually found in an immersive 3D environment. We compared the performance of five users when pointing at the *pwm* with a normal 2D GUI environment. It takes 55% more time for a user to complete a pointing task than in a 2D environment, but it is functional. We also show an example in which a user operated the *pwm* and a genome analysis application simultaneously, demonstrating that the *pwm* is a useful ancillary tool for 3D applications.

### 1. はじめに

CAVE<sup>1)</sup>, CABIN<sup>2),3)</sup>, COSMOS<sup>4)</sup>( 図 1 )などの没入型多面ディスプレイによって、より臨場感の高い仮想 3 次元空間を体感できるようになった。それと同時に 3 次元位置姿勢情報をもとにう入力装置による、仮想 3 次元物体の操作も可能となった。このような没

入 3 次元環境により、主に 2 次元ディスプレイ・マウス・キーボードで構成される既存の 2 次元環境では、提示が困難だった 3 次元アプリケーションの研究開発がさかんに行われている。一方、既存 2 次元ウィンドウシステムには数多くの有用なアプリケーションがすでに利用可能であり、我々にとって手放し難い道具として、日常的に活用されている。そのため、仮想 3 次元空間で、3 次元アプリケーションを利用しつつ、既存の GUI アプリケーションもそのまま利用できれば、没入 3 次元環境の実用性も、よりいっそう高められると考えられる。

仮想 3 次元空間で 2 次元ウィンドウを使用すると

<sup>†</sup> 通信・放送機構

Telecommunications Advancement Organization of Japan

<sup>††</sup> 東京大学

The University of Tokyo



図 1 没入型 6 面ディスプレイ: COSMOS

Fig. 1 An immersive six-screen display: COSMOS.



図 2 COSMOS での pwm の使用風景

Fig. 2 A user operating 2D GUI applications with the pwm in the COSMOS.

いう考え方は、Fisher らによる Virtual Environment Display System<sup>5)</sup> において、想定される応用例として示されていた。具体的なシステムの実装としては、X Window System のウィンドウを対象にした Dykstra のシステム<sup>6)</sup> や、Microsoft Windows のウィンドウを対象にした Task Gallery<sup>7)</sup> が実現されている。しかし、これらのシステムではディスプレイ、入力装置ともに、従来の 2 次元環境が使われており、没入 3 次元環境での入力装置の利用についての考慮はなされていない。また、Feiner らはシースルーヘッドマウントディスプレイでウィンドウを空中に表示する実験システム<sup>8)</sup> を試作しているが、これは現実空間を利用した Augmented Reality を目指したシステムであり、仮想 3 次元空間を用いた 3 次元アプリケーションとの融合についての検討はなされていない。

没入型多面ディスプレイは広視野角で立体感のある仮想 3 次元空間（以下没入 3 次元空間と呼ぶ）を提示することができる。現在、研究開発がさかに行われている、没入 3 次元空間を使った 3 次元アプリケー

ションに、2 次元環境を統合することができれば、さらに実用性の高いアプリケーション作業環境の実現が期待できる。そこで、我々は、没入 3 次元空間の中で、既存の GUI アプリケーションを利用可能にするシステム pwm (Plate Window Manager) を作成した（図 2）。以下、本論文では pwm の意義、実装技術、性能、およびその応用例について論じる。特に、実現した pwm システムの操作性の指標の 1 つとして、没入 3 次元空間の中での 2 次元 GUI の操作性に関しては実験による検討を行った。

## 2. pwm の設計指針

### 2.1 3 次元アプリケーションとの共存

pwm はユーザに没入 3 次元空間での既存 GUI アプリケーションを操作可能とするシステムである。没入型多面ディスプレイには、

- 立体視、
  - 広視野角、
  - ユーザの視点位置、方向に基づく高速映像生成、
- という特徴があり、ユーザの視界をほぼ、あるいは完全に立体映像で覆うので、あたかも仮想世界に没入したような感覚が得られる。この没入感を活用した、さまざまな 3 次元アプリケーションの研究開発が進められている。我々が目標としている技術はそれらと既存 GUI アプリケーションが共存可能な技術である。よって、pwm の導入によって、この没入 3 次元アプリケーションの使用形態を変えたり妨げたりすることは極力避けなければならない、なるべく既存の使用形態のままで使えるようにすることが望ましい。すなわち、ユーザは立った状態であることを前提にし、ユーザのスクリーン内を動き回る行動をなるべく妨げず、映像を隠す物体は極力排除しなければならない。

たとえば、仮想 3 次元空間内で GUI アプリケーションを使うことを考えてみると、まず、机や椅子を置く使用形態が考えられる。机や椅子を置くと、ユーザは座ってキーボードやマウスが使用可能になるが、床面にも映像を投影する没入型多面ディスプレイの場合、机を置くと映像を隠してしまうので、せっかくの広い視野角を狭めて没入感を損なうことになるため、通常は避けることが多い。特に、床面の映像提示を背面投影で行っているシステムでは、床面の強度が不足する危険があり、机や椅子の設置は避けなければならない。よって、通常ユーザは立った状態で作業を行う。この状態だと、既存 2 次元環境で通常用いられるマウスは使用できない。

一方、没入型多面ディスプレイでは、ユーザの視点



図 3 仮想 3 次元空間に浮かぶ歪んだ形の 2 次元ウィンドウ板  
Fig. 3 Deformed looking 2D window plates floating in a virtual 3D space.

位置を計測するための 3 次元位置姿勢計測器が用いられており、この 3 次元位置姿勢センサの 1 つを入力装置に組み込むことがよく行われている。没入 3 次元アプリケーションでユーザは、仮想物体に対する操作をこの 3 次元入力装置で行う。そのため、pwm では、没入 3 次元空間における既存 GUI 操作も、この 3 次元入力装置を使ったものとした。

## 2.2 没入 3 次元空間でのウィンドウの利用

没入 3 次元空間中にウィンドウ板が提示されると、観察者の見たい位置や方向は、観察者自身の身体を使った自然な見渡し、覗き込み行動をとるだけでよい。また、ウィンドウ板を正しく長方形として認識することができる。しかし、没入型でないディスプレイではそうはいかない。

まず、視野角が狭い 3 次元ディスプレイであれば、見えているはずの板の一部または全部が見えなくなってしまうことがあり、観察者の見渡し、覗き込み行動を制約してしまう。

また、視点追跡機能がない立体視機能つきディスプレイでは、あらかじめ想定された位置から観察しないと歪んで見えてしまう。よって、視点を移動するには、見渡し、覗き込みではない、不自然な別の操作手段に頼るしかない。観察者がディスプレイとの物理的相対位置を固定するならば、立体視映像により、どの板も正しく長方形と認識することができる。

しかし、立体視機能をも用いないディスプレイでは、観測者に対して傾いた位置にある物体は歪んで見えてしまい、長方形の板も、正しく長方形と認識されないことがある(図 3)。

すなわち、既存 GUI アプリケーションを仮想 3 次元空間で使用する技術の是非を、没入 3 次元空間で評価することは、

- 板のひずみによるユーザの不快感と誤解の危険性が除かれる、
- 視点移動のための不自然で余計な操作が不必要、という点で、有意義である。

## 2.3 3 次元環境利用者にとっての利点

我々は普段、計算機を道具として使って仕事をする際、2 次元環境で操作している。ウェブ閲覧をはじめ、2 次元グラフ表示、ファイル操作、画像編集など、我々が日常的に使うアプリケーションは、一通り揃っている。

3 次元環境で、3 次元アプリケーションのみしか使用できないとなると、道具としての使用が可能なくらいにアプリケーションの質と数が揃うまでに時間がかかることが容易に予想できる。2 次元環境で利用可能である数多くの有用なアプリケーションを 3 次元アプリケーションとともに使うことができれば、その 3 次元アプリケーションの製作者の想定以上のタスクが遂行可能となり、非定型的タスクにも対応できる応用範囲の広い環境となることが期待できる。

さらに、初心者にとっても 2 次元環境は、使用方法の解説書の充実や、教えを乞うことのできる経験者が周りにいるなど、すでに蓄積されている使用方法のノウハウが利用できるという利点がある。さらに、熟練者にとっても、使い慣れた既存 2 次元アプリケーションを使うことができるという利点がある。たとえば、データ分析アプリケーションで、さまざまなパラメータで 2 次元折線グラフを作成する必要があるとする。まず、3 次元アプリケーションにそのような機能がなければ、2 次元環境に戻って操作しなければならないが、3 次元アプリケーションにグラフ作成ツールあるいは機能があったとしても、機能不足やその使用方法でトラブルとなるケースもあるだろう。3 次元環境で既存 GUI アプリケーションが使うことができれば、得意な既存 GUI アプリケーション版グラフ作成ツールで、よりスムーズにタスクを達成することが可能になる。

## 2.4 3 次元アプリケーション開発者にとっての利点

pwm の実現は 3 次元アプリケーション開発者にとってもいくつかの利点が考えられる。第 1 に、要求されているタスクを達成できる GUI アプリケーションがすでにある場合は、それをわざわざ 3 次元環境用に新たに実装し直す必要はない。また、タスク達成のために新たに 3 次元アプリケーションを実装しなければならない場合にも、そのタスクのうち既存 GUI アプリケーションによって実現できる部分は、それらを流用することによって達成できるので、実装の労力を削減できる。

さらに、タスク達成が可能な既存 GUI アプリケーションがないとしても、ユーザとの対話部分で、既存の 2 次元 GUI 部品を利用することによって、より容易に達成できるサブタスクもあるだろう。その際、アプリケーション開発環境も開発し慣れたものが使用できる。すなわち、使い慣れた言語、デバッグ環境、実績がある洗練された GUI ライブラリ、GUI 配置を容易にするインタフェースビルダなどの開発ツールが使用可能となる。

このように、既存 2 次元 GUI の流用により、3 次元アプリケーション開発者は、2 次元 GUI では実現できない、没入 3 次元環境ならではのアプリケーションの開発に、より専念できるようになると考えられる。

### 2.5 UI 環境の将来展望

既存 2 次元 GUI には、ウィジェットやコントロールと呼ばれる GUI 部品があり、テキスト入力ボックス、ボタン、メニュー、スライダなどの部品が標準的な対話手段として使われる。このような GUI 部品を、飛び出し表示、引込み表示、枠付き表示などの 3 次元形状にして、仮想 3 次元空間で利用可能にする研究も行われており<sup>9)</sup>、いずれ、このような、3 次元 GUI 部品とも呼ぶべきソフトウェア部品が普及する時代が来るだろう。2 次元環境ではありえなかった 3 次元ならではの操作技術が開発され、普及段階を迎えたととしても、既存の GUI 部品技術の必要性がただちに薄れるとは考えにくい。そして、これらの GUI 部品は、本質的には 2 次元で操作できるものなので、pwm の既存 2 次元 GUI 流用技術で十分にその機能を実現できる。

一方、3D widgets の研究<sup>10)</sup> など、3 次元環境ならではの部品化の試みもなされており、そのうち、2 次元環境ではありえなかった 3 次元環境ならではのインタラクション手法が研究開発され、それらが普及する時代も来るだろう。没入型多面ディスプレイなどの高品位 3 次元提示技術、そしてこれらの 3 次元操作環境技術の進歩により、3 次元あるいはそれ以上の次元の情報を大量に、しかも日常的に扱う時代がこれから到来するかもしれない。しかし、有史以来、我々人類がやりとりしてきた情報は、文書や、絵、写真など、2 次元で表現できるものが大半であった。今日のように 3 次元以上の情報を容易に記録、流通できる時代になっても、それらが我々に提示される段階になると、画像、グラフ、プレーンテキスト、電子化文書、ウェブページなど、形は違うが、基本的には 2 次元で表現できるものが大半である。これら 2 次元での情報提示の必要性が、将来、皆無になることはまず考えられない。これら 2 次元情報を 3 次元空間に提示・操作する

とき、pwm のような既存 2 次元技術流用手法は有用であると考えられる。

## 3. pwm の構成方法

### 3.1 3 次元ディスプレイサーバ

X Window System<sup>11)</sup> のようなクライアント・サーバ型ウィンドウシステム(図 4)では、クライアントプログラムが表示装置や入力装置に対して直接働きかけを行うことはなく、表示はディスプレイサーバに依頼するし、ユーザからの入力情報をサーバ経由で受け取る。この仕組みを利用して、2 次元ウィンドウシステムアプリケーションを、プログラムの変更なしに、没入型多面ディスプレイなどの 3 次元ディスプレイへ表示し、かつ操作可能にするディスプレイサーバ(図 5)を構成することは可能である。これを 3 次元ディスプレイサーバと呼ぶことにする。

その実現方法には、内部にサーバ機能のすべてを含む方法と、外部の描画内容を取得するプロキシシステムによる実現方法がある。サーバ機能内蔵実装では、ウィンドウ内容の更新を容易に知ることができるが、実装が困難である。また、2 次元環境用の 2 次元システムに仕様変更が発生すれば、その対応も容易ではない。一方プロキシシステムは、描画機能は外部に任せ、その描画内容を定期的に取得し、3 次元空間に提示するシステムで、3 次元環境ユーザから見たとき、あたかも、ディスプレイサーバのように見えるシステムである。ウィンドウ内容の更新を知ることが困難であるが、仕組みが単純で実装が容易であるし、また、クライアント・サーバ型の構成になっていない他のウィンドウシステムへの応用も考えられる、柔軟性の高い実装方法である。pwm での 3 次元ディスプレイサーバの実現方法は、このプロキシシステムによるものとした。

### 3.2 実装

既存 2 次元ウィンドウシステムにおけるウィンドウを仮想立体空間で操作するためのシステムのプロトタイプシステムとして、X Window System を没入型多面ディスプレイで使用するためのシステム pwm を作成した(図 6)。今回の実装では、X サーバとして X Consortium が配布した X Window System, Version 11 Release 6.4 のサンプル実装の中に含まれていた Xvfb(X virtual frame buffer)を使用している。X クライアント(たとえば Netscape Communicator など)は Xvfb に対して、通常の X サーバと同様のアクセスを行う。ユーザに直接表示されることのない Xvfb のフレームバッファに描画されたウィンドウ画像は、pwm によって読み取られ、没入 3 次元空間内に

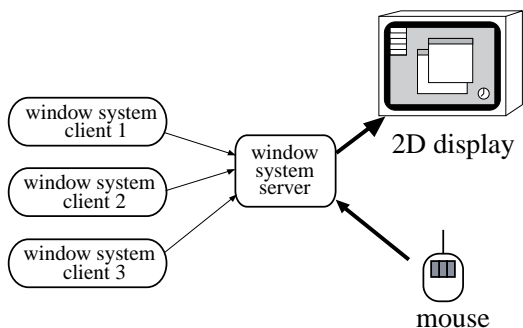


図 4 クライアント・サーバ型ウィンドウシステム  
Fig. 4 A 2D window system employing a client-server architecture.

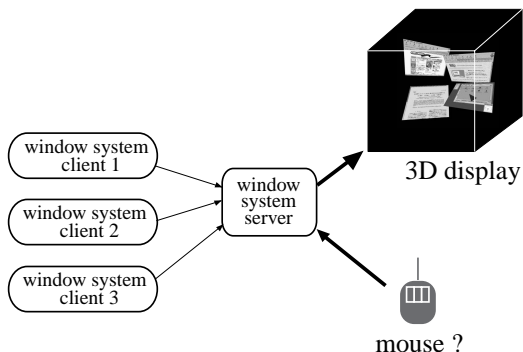


図 5 既存アプリケーションが利用可能な 3 次元ウィンドウシステム  
Fig. 5 A 3D window display server can provide utilities for existing applications.

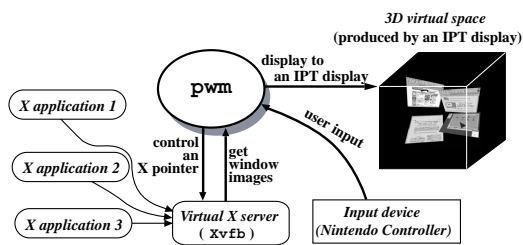


図 6 pwm 構成図  
Fig. 6 Current pwm implementation: a proxy display system.

表示される。また、ユーザがポインタ操作を行うと、それによって、pwm が X Test Extension 機能を利用して X サーバのマウスポインタを制御する。

没入型多面ディスプレイとしては、6 面の没入型ディスプレイ COSMOS を使用し、計算機としては SGI 社製の Onyx2 InfiniteReality (16 CPU, グラフィックパイプ 6 本, メモリ 4 GB, OS: IRIX 6.5) を 1 台使用した。また、入力装置として、任天堂社製家庭用ゲーム機 Nintendo 64 のコントローラ (以下コント

ローラと呼ぶ) を使用した。これには Polhemus 社製 Ultratrak のレシーバ (センサ) 1 つが内蔵されており、6 自由度の 3 次元位置姿勢を検出可能である。板のウィンドウの表示は、ウィンドウ上のピクセルが 1024 ピクセルで 1 m の大きさになるよう表示した。

ソフトウェアは C++ 言語で記述されており、CABIN で OpenGL を利用するために使用されていた glCABIN library (COSMOS 用に改造) を使用した。Ultratrak は 120 Hz 以上の頻度で測定を行った。また、システムの描画更新速度は、800 × 600 ピクセルのウィンドウ板が 10 枚表示された状態で、平均 35 Hz 以上であった。

また、現在のシステムの実装では、実際に操作可能な板は 1 つであり、残りの板はスナップショット、すなわち、ユーザが指示したときに、そのときの操作可能な板の表示内容をそのまま保持し続ける板である。リアルタイムに操作可能な板が 1 つあれば、3 次元アプリケーションとの連携の確認は可能であると考えた。

また、文字列入力機能の導入については、実際の利用に際して、ぜひ実現しなければならない機能であるが、現時点では検討中の段階であり、普通のキーボードや携帯端末用、ウェアラブル計算機用キーボードなど、さまざまな種類のキーボードでの実験を予定している。

### 3.3 pwm での操作方法

pwm でユーザは、

- 操作対象板の選択,
  - 板の位置姿勢制御操作,
  - 2次元 GUI 操作 (ポインタの移動, クリック),
- の 3 種類の制御が必要である。これらの制御はユーザが思った位置姿勢に限りなく近い位置姿勢に、より少ない労力で操作できる必要がある。今回の実装では、1 個の 3 次元位置姿勢センサ付きコントローラにより、これらすべての操作方法を実現した。

まず、操作対象となるべき板が複数ある場合には、コントローラの左右ボタンで選択する。また、板の位置姿勢制御操作と 2 次元 GUI 操作は別のモードに分け、L ボタンを押すと板の位置姿勢制御操作モード、R ボタンを押すと 2 次元 GUI 操作モードとなるようにした。

板の位置姿勢制御操作は、コントローラの位置姿勢により制御される。この際、6 自由度の相対位置による位置制御手法 (以下、6 自由度ドラッグと呼ぶ) を採用した。これは、コントローラがどのような方向に向いていようと、ドラッグボタンを押すことにより、そのときに板が向いていた位置と姿勢から位置姿勢制



図 7 6 自由度ドラッグ

Fig. 7 A 6 DoF drag technique.

御が始まり，ドラッグボタンを押している間は，コントローラの位置と姿勢の変化量を，新しい板の位置姿勢に反映させ，ドラッグボタンを離すと，板の制御はそこで止まり，その板は再びドラッグ操作がない限りそこにとどまり続けるというものである（図 7）．6 自由度ドラッグでは，絶対位置姿勢制御と違って，操作開始時にコントローラを無理な位置や方向に合わせる必要がなく，操作目標へ制御する際に，コントローラを無理な位置や方向へ向ける必要もない．物理的 3 次元世界の物体操作モデルに比較的近く，ドラッグボタンの押し離し操作に慣れれば，初めてこのシステムに触る人でも，自分の思った位置姿勢に板を置くことができる．

以下に現在実装されている 6 自由度ドラッグ処理を示す．まず，物体の姿勢を四元数<sup>(12),(13)</sup>で表すことにする．ドラッグ開始時刻を  $t_0$  とする．時刻  $t$  のコントローラの位置姿勢を  $(\mathbf{p}_c(t), q_c(t))$  とし，対象ウィンドウ板の位置姿勢を  $(\mathbf{p}_w(t), q_w(t))$  とする．ドラッグ中の対象ウィンドウ板の位置姿勢は，以下の式で表すことができる：

$$\begin{aligned} \mathbf{p}_w(t) &= \mathbf{p}_w(t_0) + \alpha(\mathbf{p}_c(t) - \mathbf{p}_c(t_0)) \\ q_w(t) &= q_c(t)q_c^{-1}(t_0)q_w(t_0) \end{aligned}$$

$\alpha$  は加速係数で現在  $\alpha = 5.0$  としている．この値は，通常の腕の長さのユーザが，なるべく 1 回の操作で無理なく望みの位置へ板を動かすことができる値として設定した．たとえば，肩からコントローラまでの長さを 40 cm とし，1 m 四方の板が 2 m の距離にある状況を想定する．ユーザから 2 m の距離とは，1 m 四方程度の板を円筒状に並べるとすると，視野 180 度に板が 1 列 6 枚並ぶ程度の距離である．正面にある板を，ユーザとの距離を変えずにユーザを中心としてユーザの周りを移動させるのに必要な操作は，半径 40 cm の円弧を描くことで実現される．また，板を 50 cm まで引き寄せるには，30 cm の操作量で済む．また，10 cm 移動させるための操作量は 2 cm であり，1 m 四

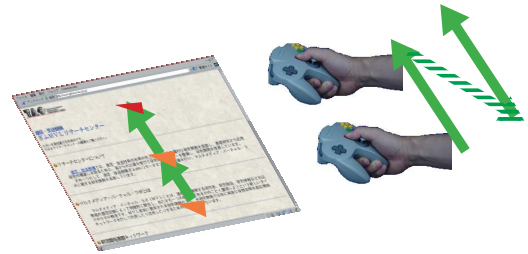


図 8 3 次元空間位置による 2 次元ポイントドラッグ

Fig. 8 A two dimensional drag technique in a 3D space.

方程度の大きさの板に対する比較的細かい配置調整操作が可能である．

また，2 次元 GUI 操作のうち，ポインタの位置制御も，コントローラの相対位置制御に基づくものとした（図 8）．現在の実装では，あたかもマウスパッドが板と平行な面にあるかのようなポインタ位置制御を行うようになっている．以下に現在実装されている 2 次元ポインタの操作処理を示す．板座標系におけるポインタの位置を  $\mathbf{p}_p(t)$ ，位置  $\mathbf{p}$  への座標系移動変換行列を  $\mathbf{T}(\mathbf{p})$ ，四元数  $q$  で表される回転への座標系変換行列を  $\mathbf{R}(q)$  とする．ユーザがドラッグしているときの  $\mathbf{p}_p(t)$  は以下の式で表すことができる：

$$\begin{aligned} \mathbf{p}_p(t) &= \mathbf{p}_p(t_0) \\ &+ \beta \cdot \mathbf{T}^{-1}(\mathbf{p}_w(t)) \mathbf{R}^{-1}(q_w(t)) \mathbf{d}_c(t) \end{aligned}$$

（ただし， $\mathbf{d}_c(t) = \mathbf{p}_c(t) - \mathbf{p}_c(t_0)$  とする）

$\beta$  は加速係数で，現在の実装では， $\beta = 1,500.0$  (pixel/m) と設定されている．この値は，ドラッグボタンを離すことなくウィンドウの端々までポインタを動かすことができ，しかも，小さめの GUI ボタンを押すことができるくらいの精密制御が可能という基準で，実際に試験操作を行いながら定めた値である．なお，この  $\beta = 1,500.0$  (pixel/m) という値は，板は 1,024 pixel/m で表示されており，800 pixel 移動のための操作量は 50 cm ほどとなる値である．また 1 pixel あたり 0.67 mm の操作量であり，170 pixel/inch のマウスの 0.15 mm よりは大きな値に設定されている．

#### 4. pwm でのアプリケーションの操作性

従来の没入 3 次元環境では，ユーザが 3 次元アプリケーションを使用している最中に，既存 GUI アプリケーションを使う必要が生じたとき，ユーザはわざわざいったん没入 3 次元空間から出て，2 次元環境で作業を行わなければならないかった．この環境変更にかかるコストはかなり大きく，

- 物理的な変更

ユーザが 3 次元環境から出るとき，液晶シャッタ

眼鏡など、3次元環境のために装着していた装備を外さなければならず、2次元環境へ移行するとき、もし3次元環境と2次元環境の入出力装置が異なる場所に設置してある場合、移動に労力と時間がかかる。

- 環境を変化させるためにかかる計算機の処理時間  
3次元環境と2次元環境の入出力装置が同一システムである場合も、ソフトウェア的切替えに必要な処理時間は不可避である。
- ユーザの視界の急激な変化による心理的負担  
ユーザの視界が没入3次元空間から急に平面になる。この見栄えの劇的な変化はユーザを少なからず驚かせる。これを軽減するために、漸時的に明暗をつけて、できるだけこの驚きを少なくする手法もあるが、これには少なからず時間を消費する。また、見栄えの急激な変化に対応するための、頭の中のモデル変更に必要な労力を完全に払拭することはできない。

があると考えられる。最初の2つは、現時点での問題点であり、これからの技術の革新により解決が期待できる。しかし、3番目のモデル変更は環境変化を行う限り、ユーザの慣れでしか解決できない。pwmは既存2次元アプリケーションを没入3次元アプリケーションとともに操作可能にすることを目的としたシステムで、この環境変更のコストを省くことが期待できる。

また、2次元環境での作業を考えてみると、画面を読んだり、考えたり、あるいはアプリケーションの応答を待ったりするなど、操作以外にも時間を消費しており、通常、純粋に操作のみに費やす時間の割合は、それほど大きいわけではない。

そのため、操作性に関して、特に大きな負担があるのでないならば、没入3次元空間で3次元アプリケーションを操作するユーザは、pwmに有用性を見出し、没入3次元空間に入ったまま既存GUIアプリケーションを操作する方法を選択することが期待できる。しかし、pwmでの操作が極端に困難だとすると、ユーザは環境変更のための労力と時間を費してでも、2次元環境へ移動せざるをえない。よって、今回試作したpwmで、2次元GUI操作性がどれほど実現されているのか、調べておく必要がある。この操作性の指標の1つとして、ユーザに2次元ポインティング操作を行わせ、その操作時間を測定した。これにより、ユーザに、ある2次元上の領域をクリックする必要が生じた瞬間から、どれくらいの時間でそれが達成できるかの指標となりうる。また、その結果がどのくらいの操作性を意味するのかを知るために、2次元環境で同様の測定を

行い、基準とした。

pwm実装での操作性の検討についてはフィッツ則を用いた。フィッツ則では、動作開始点から目標までの距離を $A$ 、目標の許容範囲を $W$ としたとき、 $A$ と $W$ によって計算されるID (Index of Difficulty) によって、MT (Movement Time) が、 $MT = a + b \times ID$  ( $a, b$  は定数) で表される。測定の結果得られるID、MTの組に回帰直線をあてはめ、得られる直線を操作性の指標とした。ここでのIDとしては、さまざまな式が提案されているが、我々は今回、測定結果にそれらをあてはめて検討し、それらの中で最もよくあてはまったIDである、MacKenzieによって提案された、 $ID = \log_2(A/W + 1)$  を用いた<sup>14)</sup>。

今回の測定ではpwmと2次元環境という異なる2つの環境で、同一アプリケーションの同一GUI操作についての時間測定を行う。フィッツ則では、ある操作タスク達成にかかる時間の予想式が示される。ユーザの手から、そのアプリケーションへ操作が伝わる前に、pwmあるいは2次元環境を介しているので、この異なる環境が異なる操作性を示せば、それらは異なる操作タスクとして、異なる予想式が得られると予想できる。それらの予想式により操作タスクの難易度が示されるが、実際には同一アプリケーションなので、環境の違い、すなわち環境の操作性が示されると考えた。

基準とする2次元環境の測定には、IBM社製ノートパソコン Thinkpad 235 (CPU: MMX Pentium 233 MHz, RAM: 96 MB) にマウスを接続したものを使用した。OSはLinuxで、XサーバはXFree86バージョン3.3.6を使用した。使用したマウスで、操作量に対する画面上のポインタ移動量は約170 pixel/inchであった。マウスの速度設定は、加速係数1とした。Xサーバは、操作速度がある閾値以上のとき、この加速係数を乗じて、ポインタの移動速度を速くする機能を持つ。しかし、この機能により、人間の操作量とポインタの移動量は非線型となる。フィッツ則は、元々物理的な距離で確認された法則なので、人間の物理的な指示量とポインタの移動量は比例関係である必要がある。この加速機能によって、フィッツ則の前提が成立しなくなる可能性があるため、これを無効とした。

#### 4.1 実験

測定に使用したアプリケーションは、最も単純なモグラタキゲームである(図9)。通常の2次元環境(図10)と、pwmによる3次元環境での2次元GUI操作(図11)で、まったく同一のプログラムで操作時間を測定した。測定プログラムでは、

(1) ウィンドウ上のランダムな位置に、ある半径の

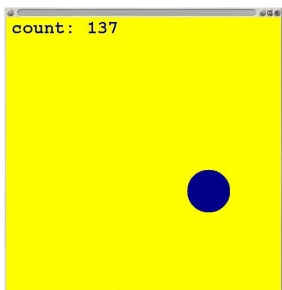


図 9 実験で用いた計測システム

Fig. 9 A window image of a picking experiment application.

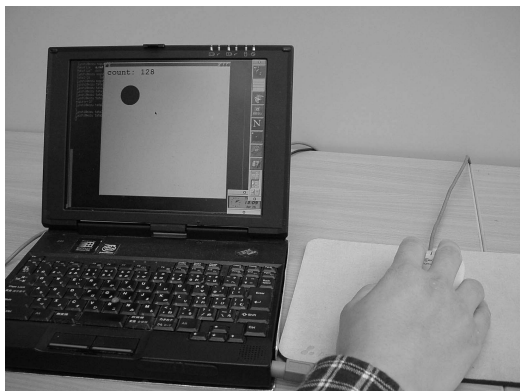


図 10 2次元環境での実験の様子

Fig. 10 A picking experiment at a normal 2D environment.

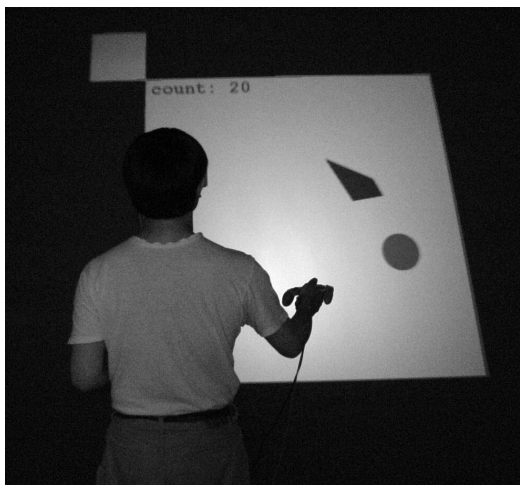


図 11 COSMOS での実験の様子

Fig. 11 A picking experiment at COSMOS.

円を表示する,

- (2) ユーザは円内にポインタを合わせてクリックする,
- (3) クリックを検出したら即座に次の円を表示する, という試行を繰り返した.

表 1 2次元環境と pwm での各被験者の動作時間

Table 1 Measured motion time comparing the pwm with a 2D environment which consists of a 2D display and a mouse.

被験者 (環境)	操作時間	成功数	試行数	成功率
A (pwm)	$767.9 + 195.8 \times ID$	690	779	88.5%
A (2次元)	$471.4 + 115.9 \times ID$	389	415	93.7%
B (pwm)	$988.5 + 124.8 \times ID$	487	497	98.0%
B (2次元)	$444.3 + 182.3 \times ID$	289	299	96.7%
C (pwm)	$485.2 + 239.4 \times ID$	763	991	77.0%
C (2次元)	$465.8 + 107.7 \times ID$	336	416	80.8%
D (pwm)	$639.7 + 226.5 \times ID$	3318	3635	91.3%
D (2次元)	$463.2 + 159.3 \times ID$	490	502	97.6%
E (pwm)	$842.2 + 224.0 \times ID$	681	720	94.6%
E (2次元)	$325.7 + 203.2 \times ID$	782	790	98.9%

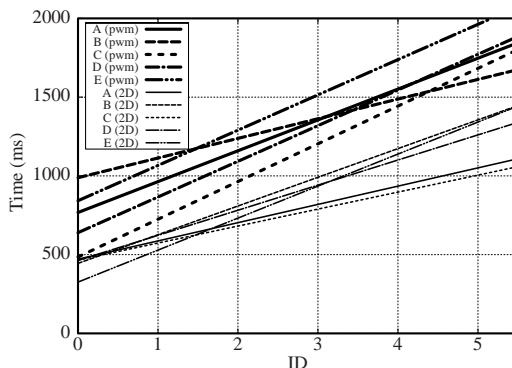


図 12 被験者ごとの回帰直線結果

Fig. 12 Regression lines of each subject.

ポインタの初期位置から円の中心までの画面上の距離 (単位はピクセル) と、ユーザがポインタを動かしてはじめてから、クリックするまでの時間を記録した。表示される円の位置は、その全部が表示されるよう、また、次の円が現在のポインタ位置を含まないように位置決めされている。また、1回の実験中には円の半径は変えず、3回以上の実験で3種類の半径で測定を行った。被験者は5人で1回の実験で少なくとも100回のクリックを行うようにした。モグラたたきのウィンドウの大きさは  $512 \times 512$  ピクセル、円の半径は10, 25, 40ピクセルとした。

4.2 実験結果

表 1 は、得られた測定結果を被験者ごとにまとめたもので、図 12 は、被験者ごとにあてはめた回帰直線を図に表したものである。これらによると、被験者ごとには個人差があるものの、各被験者とも、基準となる2次元環境に比べ、pwmでは余計に時間がかかっている傾向が見られた。

このとき、距離 A は 17 から 627 ピクセルの範囲内であり、ID は 0.9 から 5.01 の範囲であった。また、今回の実験では、ドラッグボタン離し、すなわち、マ



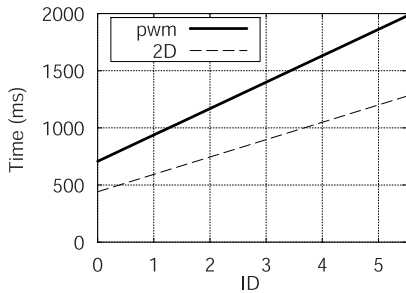


図 13 2 次元環境と比較したときの pwm の 2 次元 GUI 操作の回帰直線

Fig. 13 Regression lines of pwm and 2D environment.

表 2 pwm と 2 次元環境との比較: ID に具体的な値を入れたとき  
Table 2 Comparing the pwm with a 2D environment at ID = 1 or ID = 5.

	ID = 1	ID = 5
$MT_{pwm}$ (ms)	940	1860
$MT_{2D}$ (ms)	590	1190
$MT_{pwm} - MT_{2D}$ (ms)	350	670
$\frac{MT_{pwm}}{MT_{2D}} \times 100$	159.3	156.3

ウス操作でのマウスを持ち上げる動作にあたる操作がほとんど見られなかった。また、どの被験者からも、円半径が 10 ピクセルの実験は、十分難しいという感想が聞かれた。

また、測定値のサンプル数が最も少ないのは被験者 B であるが、その他の被験者について、それ以上の測定値を切り捨てて、被験者ごとの測定値のサンプル数を一定とし、全被験者の測定結果をあわせ、それらを基に回帰直線をあてはめた。その結果得られた式は、

$$MT_{pwm} = 7.1 \times 10^2 + 2.3 \times 10^2 \times ID \quad (1)$$

$$MT_{2D} = 4.4 \times 10^2 + 1.5 \times 10^2 \times ID \quad (2)$$

であった。図 13 は、これらをグラフにしたものである。また、式 (1)、(2) に具体的な ID 値 (ID = 1, ID = 5) を代入したものを表 2 に示す。今回の pwm の実装では、ユーザがポインティングにかかる時間は、2 次元環境を基準にすると約 55% 増しで、350 ms から 670 ms 増しであるという結果が得られた。

## 5. 考 察

没入型ディスプレイの導入数は、今日徐々に増え続けている。我々は即座に利用可能な技術を目指し、これらの没入型ディスプレイによく使われる位置姿勢計測器である磁気式システムだけで、板の位置姿勢制御操作や 2 次元 GUI ポインタの制御操作ができるシステムの実装を試みた。

2 次元ポインティング性能においては、既存 2 次元環境のマウスに比べて多少時間がかかるという計測結

果を得たが、これによって、2 次元環境へ移行したほうがよいかどうかは、その 2 次元 GUI 上のタスクに大きく依存するところであろう。操作性のみを追及するのであれば、pwm を改造して、より測定誤差の少ない 3 次元位置姿勢計測器を導入したり、力覚提示装置による不要自由度の制約、2 次元 GUI 用にあつらえた入力装置など、新たな装置の追加導入により改善される事例もあるだろう。しかし、そのような新たな装置を導入せずに、3 次元アプリケーションと既存 2 次元 GUI が融合できる我々の手法は、たとえ操作自体に多少より多くの時間がかかったとしても、利用価値があると考えられる。

実験後の簡単なインタビューで、特に違和感を感じることはなく、操作はしやすいとの感想を得た。また、今回の測定では、被験者に 1 回の測定で連続 100 回以上のクリックを要求したが、どの被験者もそれを遂行することができた。また、被験者のうち D だけは長い実験時間を確保し、約 55 分間で連続 1,000 回以上のクリックを行うことができた。操作が、ユーザに過度の心理的、肉体的負担を強いるものであるなら、数多くの連続操作は不可能である。このことから、pwm でのポインティング操作のみに起因する負担は少なく、通常の用途に十分使えるものと考えられる。

今回の pwm の実装では、2 次元画面更新を 1 秒間隔のポーリングで行っており、また、クリック後はその反応画面を得るのは、必ず 300 ms 待った後である。最悪のタイミングだと、クリックしてから、その更新された画面が表示されるのに 1.3 秒遅れることもありうる。しかし、今回の測定では、測定アプリケーションで 2 次元ポインタの動き始めを検出してから時間計測を開始している。ユーザが、次の円が提示されてからコントローラを動かし始めるとするなら、時間計測開始は画面更新が行われた後であることが期待できる。また、ポインタの現在位置提示は、3 次元映像更新頻度で行われており、操作中、すなわち時間計測中のポインタの位置をユーザはつねに確認できる。よって、この 2 次元画面更新の遅れは、今回の計測に大きな影響を与えていないと考えられる。

## 6. pwm と 3 次元アプリケーションの統合例

没入 3 次元空間を使った 3 次元アプリケーションを操作しながら、同時に既存 2 次元アプリケーションを操作することができる機能の有用性を示す具体的な応用例として、pwm と遺伝子クラスタ情報可視化ツール<sup>15)</sup>との統合を行った(図 14)。

巨大な情報量を持つ遺伝子データは、遺伝子の発現



図 14 ゲノム情報分析アプリケーションの使用と同時に pwm で “the Human genome”<sup>16)</sup> ページを見ているユーザ

Fig. 14 A user using the genome function analysis application and watching “the Human Genome”<sup>16)</sup> Web pages with pwm.

量や既知の機能情報、染色体上の位置など、さまざまな基準でクラスタ化することができる。このゲノム情報分析ツールは、3次元空間上に遺伝子クラスタの比較マップを表示することで、遺伝子情報の分析を行うシステムである。比較マップ上での可視化データからは共通遺伝子の候補などが見出されるが、遺伝子分析を行うには、同時に各遺伝子の性質を考慮に入れた検討が必要になる。そのため、従来は仮想3次元空間での可視化と通常の端末を用いたデータ検索を繰り返しながら作業を進めなければならなかったが、開発したシステムではゲノム情報分析ツールに pwm を組み込むことで、仮想3次元空間上で2次元ウィンドウを表示し、同時にウェブを用いた情報検索を行うことができる。本研究では、このシステムを実際に肝細胞癌と肝芽腫の遺伝子情報解析に適用してもらった。肝細胞癌と肝芽腫は、どちらも肝臓の癌ではあるが、生物学的にはまったく異なる症例である。ここでは、分化度が3段階に異なる肝細胞癌を25サンプルと、分化度が2段階に異なる肝芽腫の8サンプルという2種類の実験セットの階層型遺伝子クラスタについて、比較マップ手法を適用した。この結果、7個の共通な遺伝子の候補が見つかったが、pwmを用いた情報検索と併用した結果から、このうち6個の遺伝子が metallothionein という金属で誘導される遺伝子群であることが特定できた。最終的な特定には、さらに生化学的な調査も必要であるが、pwmを用いることで無数にある調査候補の中から、より可能性の高い遺伝子を効率的に選別することができた。このように pwm では、当該3次元アプリケーションの作業場であった没入3次元空間の中で、そのまま既存2次元アプリケーション(この場

合はウェブブラウザ)の活用フェーズヘシームレスに移行することができ、より効率的なデータ分析を進めることが可能であることが示された。

## 7. 結 論

没入型多面ディスプレイで、既存 GUI アプリケーションを操作可能とするシステムである pwm について述べた。没入3次元環境そのものをより使いシステムにするために、

- 3次元アプリケーションと同時に操作可能な技術であること、
- ユーザは立った状態で、3次元位置姿勢情報をともなう入力装置で操作すること、
- 既存の没入型多面ディスプレイに特別な装置を必要としない技術であること、

という要件を満たすよう pwm を設計した。没入型多面ディスプレイにおいて、よく使われる位置姿勢計測器である磁気式システムだけで、pwm を実装することにより、既設の没入型多面ディスプレイですぐに利用可能な技術の可能性について明らかにした。

また、現在の pwm 実装での2次元ポインティング操作の操作性の指標の1つとして、2次元ディスプレイとマウスで構成される従来の2次元環境を基準とした指標を示した。その結果、現在のハードウェア構成および pwm の実装における純粋に操作だけにかかる時間は、2次元環境を基準とすると55%増しであることが分かった。そして、その操作性でも、遺伝子クラスタ情報可視化ツールとの統合での使用例を示すことにより、没入3次元アプリケーションを補強するサイドツールとして十分に活用される可能性を示した。

謝辞 遺伝子クラスタ情報可視化ツール作者の加納真氏、COSMOS 維持管理者の山田俊郎氏、ならびに COSMOS を保有する岐阜県科学技術振興センターに感謝いたします。

## 参 考 文 献

- 1) Cruz-Neira, C., Sandin, D.J. and DeFanti, T.A.: Surround-Screen Projection-Based Virtual Reality: The Design and Implementation of the CAVE, *Computer Graphics Proceedings, Annual Conference*, pp.135-142 (1993).
- 2) Hirose, M., Ogi, T., Ishiwata, S. and Yamada, T.: Development and Evaluation of Immersive Multi-screen Display CABIN, *Systems and Computers in Japan, Scripta Technica*, Vol.30, No.1, pp.13-22 (1999).
- 3) 廣瀬通孝, 小木哲朗, 石綿昌平, 山田俊郎: 多

画面型全天周ディスプレイ (CABIN) の開発とその特性評価, 信学誌 D-II, Vol. J81-D-II, No. 5, pp. 888-896 (1998).

- 4) Yamada, T., Hirose, M. and Iida, Y.: Development of Complete Immersive Display: COSMOS, *Proc. VSMM 98*, pp. 522-527 (1998).
- 5) Fisher, S.S., McGreevy, M., Humphries, J. and Robinett, W.: Virtual Environment Display System, *Interactive 3D Graphics* (1986).
- 6) Dykstra, P.: X11 in Virtual Environments: Combining Computer Interaction Methodologies, *Proc. IEEE 1993 Symposium on Research Frontiers in Virtual Reality* (1993).
- 7) Robertson, G., Dantzich, M., Robbins, D., Czerwinski, M., Hinckley, K., Ridsen, K., Thiel, D. and Gorokhovskiy, V.: The Task Gallery: A 3D Window Manager, *Proc. CHI 2000* (2000).
- 8) Feiner, S., MacIntyre, B., Haupt, M. and Solomon, E.: Windows on the World: 2D Windows for 3D Augmented Reality, *UIST '93* (1993).
- 9) Heath, D.: Virtual User Interface (VUI) — A Windowing System for VR, *2nd International Immersive Projection Technology Workshop* (1998).
- 10) Zeleznik, R.C., et al.: An Interactive 3D Toolkit for Constructing 3D Widgets, *SIGGRAPH 93*, pp. 81-84 (1993).
- 11) Scheifler, R.W. and Gettys, J.: The X Window System, *ACM Trans. Graphics*, Vol. 5, No. 2, pp. 79-109 (1986).
- 12) Shoemake, K.: Animating Rotation with Quaternion Curves, *SIGGRAPH '85*, pp. 245-254 (1985).
- 13) Hearn, D. and Baker, M.P.: *Computer Graphics — C version*, Prentice Hall (1997).
- 14) MacKenzie, I.S.: Movement Time Prediction In Human-Computer Interfaces, *Graphics Interface '92*, pp. 140-150 (1992).
- 15) Kano, M., Tsutsumi, S., Nishimura, K., Aburatani, H., Hirota, K. and Hirose, M.: Visualization for Genome Function Analysis Using Immersive Projection Technology, *IEEE VR* (2002).
- 16) National Center for Biotechnology Information: The Human Genome — A guide to online information resources.  
<http://www.ncbi.nlm.nih.gov/genome/guide/human/>

(平成 14 年 4 月 19 日受付)

(平成 14 年 10 月 7 日採録)



立山 義祐 (正会員)

昭和 44 年生。平成 6 年東京大学大学院工学系研究科機械工学専攻修士課程修了。平成 9 年同大学大学院工学系研究科情報工学専攻博士課程単位取得退学。平成 9 年新エネルギー産業技術総合開発機構提案公募研究員。平成 11 年より通信・放送機構ぎふ MVL リサーチセンター研究員、現在に至る。GUI, パーチャルリアリティにおける操作処理に関する研究に従事。ACM, ヒューマンインタフェース学会各会員。



小木 哲朗

昭和 35 年生。昭和 61 年東京大学大学院工学系研究科産業機械工学専攻修士課程修了。同年より(株)三菱総合研究所勤務。平成 6 年東京大学大学院工学系研究科産業機械工学専攻博士課程修了。博士(工学)。平成 8 年同大学大学院工学系研究科助教授。インテリジェント・モデリング・ラボラトリー担当。平成 11 年より通信・放送機構研究員、現在に至る。専門はシステム工学, パーチャルリアリティ, 特に没入型ディスプレイの研究開発に従事。日本機械学会, 計測自動制御学会, 日本バーチャルリアリティ学会, ヒューマンインタフェース学会各会員。



廣瀬 通孝 (正会員)

昭和 29 年生。昭和 52 年東京大学工学部卒業。昭和 57 年同大学大学院工学系研究科博士課程修了, 工学博士。同年同大学工学部産業機械工学科専任講師。昭和 58 年同大学工学部産業機械工学科助教授。平成 11 年同大学先端科学技術研究センター教授, 現在に至る。専門はシステム工学, ヒューマン・インタフェース, パーチャルリアリティ。主な著書に「バーチャル・リアリティ」(産業図書), 「技術はどこまで人間に近づくか」(PHP), 「脳都市の誕生」(PHP) 等がある。日本バーチャルリアリティ学会理事, ヒューマンインタフェース学会評議員。日本機械学会, 計測自動制御学会, IEEE, ACM 等の会員。