

# 実時間インタラクションのための3次元手姿勢判定法の提案と評価

山内 康 晋<sup>†</sup> 三原 功 雄<sup>†</sup> 土井 美 和子<sup>†</sup>

本論文では、手の3次元姿勢情報を利用したジェスチャ入力インタフェースの構築法を紹介する。マウスに代表される我々を取り巻く計算機とのインタフェースは、いまだ2次元の域を出ていない。我々は、新しい3次元ジェスチャ入力インタフェースを目指して、距離画像より検出される撮像物体の3次元姿勢情報を利用したインタフェースを構築している。距離画像は、近赤外光の反射光を利用した撮像原理に基づき開発した入力デバイス：モーションプロセッサを用い毎秒50フレームで取得を行っている。3次元姿勢情報の検出には、取得した距離画像から求められる、画像モーメント、ならびに撮像物体表面の法線ベクトルを用い、誤差3度以内の姿勢情報を実時間で検出できることを示す。さらに、人間の手を使った動作入力インタフェースに応用したアプリケーションを紹介し、手の傾きやひねりといった3次元姿勢情報を利用することで、より直感的な情報入力インタフェースが実現できることを示す。

## Proposal and Experiments for Hand's 3D Posture Detection for Real-time Human-computer Interaction

YASUNOBU YAMAUCHI,<sup>†</sup> ISAO MIHARA<sup>†</sup> and MIWAKO DOI<sup>†</sup>

Conventional human interface devices such as a mouse handle only 2D information. Our purpose is to build a 3D interface using a real-time depth image. A new kinetic and 3D image input device Motion Processor can detect the distance from the camera to the object surface in 50 frames/sec. This paper shows an object's 3D posture detection mechanism using image moments and normal vectors of the target object's surface detected by the Motion Processor, and also shows the detection error of posture angle is within 3 degrees through an experiment. We also show some applications using the detected 3D posture information of the hand. These applications show that 3D gesture interface is essential to natural and intuitive communication between human and computers.

### 1. はじめに

近年、コンピュータは処理性能の飛躍的な向上、低価格化などを背景に、オフィスや家庭で広く利用されるようになってきた。また、認識精度の向上にともなう音声認識システムの実装などで、より自然なコンピュータとのコミュニケーションが実現されつつある。さらに、CCDカメラなどに代表される撮像デバイスの実装により、自然画像や動画といったイメージ情報を用いたインタラクションはよりユーザに身近なものになってきた。しかし、いまだ多くの人々にとって、コンピュータは必ずしも直感的に扱えるものになったとはいえない。我々、人間同士のコミュニケーションでは、音声だけでなく、身振りや手振りといったノンバーバル情報が、コミュニケーション意図の伝達に重

要な役割を果たしていると考えられている<sup>1)</sup>。このようなノンバーバル情報を利用することで、より現実に近い直感的で、自然なコンピュータ操作を可能とする機構が切望されている。

従来の画像処理の多くは、2次元画像を対象に、ジェスチャなどの認識に関して様々な研究が行われてきている<sup>1),2)</sup>。撮像画像から手などの対象物を抽出するには、Snake法<sup>3)</sup>などを利用してきたが、多大な時間と計算量を必要とするという問題があった<sup>4),5)</sup>。また、抽出した対象物を解析することで得られる情報は、2次元撮像平面内での位置や形状、あるいは動きであり、空間的にかなり制約を受けたものとなっていた。より自然な感覚での操作インタフェースを構築するためには、3次元空間中での物体位置や形状、そして動きの検出が必要と考えられる。撮像物体の3次元情報を取得するための方法としては、モーションキャプチャを利用する方法やカメラを複数台用いる方法、そしてレンジファインダを利用する方法がある。それぞれ、装

<sup>†</sup> 株式会社東芝研究開発センターマルチメディアラボラトリー  
Multimedia Laboratory, Corporate Research and Development Center, TOSHIBA

着デバイスが必要であったり、背景画像に影響を受けたり、撮像自身がリアルタイムでなかったりするなどの問題があった。

本論文では、リアルタイムに対象物体の切り出しが行えるデバイス「モーションプロセッサ」<sup>6)</sup>により、3次元操作インタフェースの構築を行う。モーションプロセッサは、光（近赤外光）の反射の原理を用いて、背景画像を取り除くことで、対象物の画像のみを取り出す新規画像処理技術を採用した画像情報入力デバイスである。反射光量は、物体とカメラの距離によって変化するので、物体の3次元的な形状情報を取得することができる。そこで、モーションプロセッサにより切り出された物体の3次元形状情報を基に、3次元空間内での撮像物体の動きや、姿勢検出手法を示す。さらに、人の手を対象に検出された3次元位置、姿勢情報を用いることで、仮想空間内の物体操作やインタラクションが行えることを示す。

以下、2章では対象切り出しを行うモーションプロセッサの概要を説明する。3章では、距離画像を対象に特徴量を算出することで、画像内回転姿勢、奥行き方向の傾き姿勢といった3次元姿勢情報の検出手法について説明し、4章では、手の姿勢情報を利用した仮想物体操作、およびインタラクションを行うアプリケーション応用について述べる。

## 2. モーションプロセッサの概要

図1は開発したモーションプロセッサの外観であり、手を対象に撮像した画像を図2に示す。中央のレンズの周囲に、近赤外光を照射するLEDを配してあ



図1 モーションプロセッサ外観  
Fig. 1 Outlook of motion processor.



図2 モーションプロセッサ撮像画像  
Fig. 2 Capture image of motion processor.

り、レンズの奥に受光部のカスタム・センサチップがある。撮像画像を見ると、手の部分のみが検出されており、背景にある物体が撮像されていないことが分かる。これは、モーションプロセッサの動作原理に基づいている。

図3にモーションプロセッサの動作原理を示す。LEDから発光された近赤外光が物体で反射され、センサ部分でその反射光を受光するようになっている。反射光の強度は、その伝播距離の2乗に反比例して減衰するため、撮像物体の背景からの切り出しが可能になっている。つまり、背景から反射してくる光は微弱であり、モーションプロセッサでは検出されない特性を利用し、撮像物体の背景からの切り出しを行っている。また、モーションプロセッサ撮像画像中、画素値が高い部分（色の濃い部分）ほどモーションプロセッサから撮像物体までの距離が近いことを表している。反射光の光強度を  $I$ 、物体までの距離を  $L$  とすると、式(1)のような関係式で表すことができる ( $k$  は定数)。

$$I = k/L^2 \quad (1)$$

このように反射光強度と距離との間に相関関係があるため、反射光強度から、対象物の奥行き距離情報を得ることができる。厳密には、反射光の強度は、対象物体の表面属性（反射率といった反射特性や色）などに依存するが、手のように全体を通して均一と見なせる対象であれば、反射光の強度をそのまま奥行き距離情報に対応させることができる。また、撮像系に起因

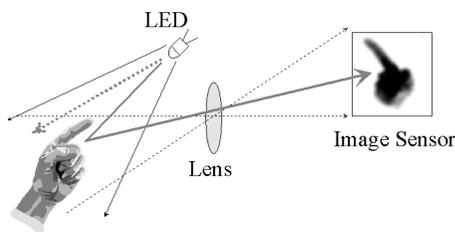


図3 モーションプロセッサ撮像原理  
Fig. 3 Principle of motion processor.

表1 モーションプロセッサ仕様  
Table 1 Specification of motion processor.

解像度	128 × 128 pixels
奥行き深度	256 階調 ( 8 bit )
動作レート	50 frames/sec
撮像レンジ	25-100 cm
水平画角	80 度
サイズ	W75 × H75 × D78 [mm]
対応システム	PC/AT 互換機
OS	Windows 95/98
接続方法	専用ボードで PCI 接続

する撮像素値の非線形性に関しては、修正モデルを仮定し、補正することで、3次元空間内、同一平面上で適正な奥行き距離値を取得することができる<sup>7)</sup>。

試作したモーションプロセッサの性能を表1に示す。

以上のように、モーションプロセッサは、背景部分の除去と同時に、対象物体表面の3次元形状情報を距離画像としてリアルタイムに取得することの可能なデバイスであり、手や身体を使ったジェスチャ入力に適したデバイスである。

また、モーションプロセッサの特性を生かしたアプリケーションの作成を行ううえで、高速な画像処理ライブラリを含んだソフトウェア開発キット(モーションプロセッサ SDK)も並行して開発してきている<sup>8),9)</sup>。

### 3. 距離画像を用いた3次元姿勢情報の検出

我々は今まで、モーションプロセッサを利用し、人間の手を使った動作入力によるマルチメディア・アプリケーションを作成してきた<sup>9)</sup>。ここでは、手の形状パターンや重心位置、あるいはそれらの変化成分を検出し、アプリケーションに反映させることで、ユーザの意図やテンポに応じたインタフェースの作成を可能としてきた。

ところが、人間の手の持つ情報量を考えた場合、位置や形状パターンといった画像レベルの特徴情報だけでなく、傾きや回転といった3次元的な姿勢情報も重要な特徴量と考えられる。特に、微妙なニュアンスの入力といったアナログ的な情報入力場面やより高次の情報入力が必要な場面では、形状パターンといったシンボルレベルに落ちた情報や、画像内重心位置といった撮像エリアなどの制約を受ける情報量で扱うより、手首の“ひねり”や“傾き”といった姿勢情報を用いた方が直感的に操作できることが期待できる。

このような対象物体の3次元姿勢情報を検出するための手法には、モーションキャプチャを利用する方法がある。この手法はシステムが大規模なうえ、接触デバイスを身体に装着する必要がある。家庭における日常的な利用形態を考えた場合、非接触デバイスであることが望ましい。複数カメラを用い三角測量の要領で3次元形状を計測する手法も存在する。この手法では、非接触に撮像物体を計測することができるが、背景画像の影響を避けるため、黒や青といった背景色設定を行う必要があった。また、距離画像を取得するシステムとしては、レンジファインダを用いる手法が存在する。CADデータに使えるだけの精緻な3次元モデルの取得が可能だが、計測時間が長く実時間インタラクションシステムの構築は難しかった。

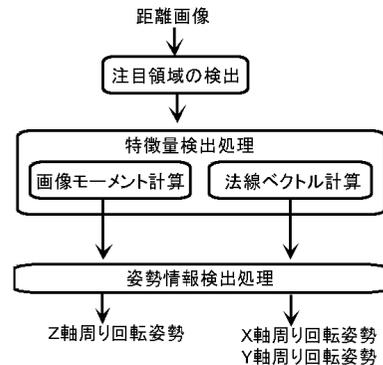


図4 3次元姿勢情報検出処理の流れ  
Fig. 4 Processing flow of 3D posture detection.

以上のように従来システムでは難しかった3次元姿勢情報を用いたインタラクション・システムの構築を、実時間に距離画像の取得できるモーションプロセッサにより実現することを考える。先に述べた人の手の“ひねり”具合を計測するためには、その変化量に応じた特徴量が検出できればよいことになる。

ここでは開いた状態の手からその手の3次元的な姿勢情報の検出を行うことにする。手が開いた状態では、人間の手の表面は1枚の平らな板と見なすことができ、この板の3次元空間内における3軸周りの姿勢回転量を手の3次元姿勢情報として定義することができる。モーションプロセッサの撮像軸に対して撮像方向(モーションプロセッサから見て奥行き方向)にZ軸、撮像面に対して水平方向をX軸、垂直方向をY軸と見なせば、撮像物体表面の3次元姿勢は、3軸周りの回転姿勢で定義することができる。そこで、撮像画像中に現れる対象物体の画像分布からZ軸周りの回転姿勢、対象物体を構成する面の奥行き方向傾き具合からX/Y軸周りの回転姿勢を検出することにする。前者の画像分布は画像モーメント、後者の奥行き方向傾き具合は撮像物体表面の法線ベクトルといった特徴量を計測すればよい。図4は本システムにおける3次元姿勢情報検出処理の流れを表したものである。以下では、各ステップを順に説明していく。

#### 3.1 注目領域の検出

画像中から特定の領域を検出する場合、従来は色情報や時間差分情報などを用いていた<sup>10)</sup>。しかし、これらの手法では、背景物体など撮像環境に強く依存したり、静止物体を検出できなかつたりするなど、安定性に関する問題があった。モーションプロセッサを用いれば、距離情報から背景画像を除去することが可能であるだけでなく、上半身の前にある手など奥行き距離で比較的近くにある複数物体を区別することも可能



図 5 ROI による手領域の抽出  
Fig. 5 Hand extraction using ROI.

である<sup>11)</sup>。ここで検出した複数の画像領域を、それぞれ注目領域 (Region of Interest: ROI) と設定し、以後の認識処理をこの ROI 内に限定することで、効率的な姿勢検出処理を可能としている。この領域検出処理を応用すれば、両手を区別して個別の ROI として切り出し、それぞれの手の姿勢情報を判定することも可能である。ROI による領域切り出し画面の例を図 5 に示す。左手を囲む矩形領域が、ROI として設定された領域である。

### 3.2 奥行き軸に対する回転姿勢の検出

モーションプロセッサ撮像画像における撮像物体の画像内回転量を検出することは、3次元空間中における撮像物体のモーションプロセッサから見て奥行き方向、つまり Z 軸周りの回転姿勢を検出することになる。画像内回転量の検出には、撮像物体の平行移動、大きさに依存しない特徴量である画像モーメントを用いることにする。画像モーメントは、画像中の画素分布から撮像物体の大まかな姿勢検出に用いられる一般的な手法であり、撮像物体の細かな形状の差も吸収できるため、手のような撮像対象に対しても、適用可能である<sup>12),13)</sup>。

画像モーメントは、以下のように求めることができる。I(x, y) を画素 (x, y) の画素値として、2 次までの画像モーメント量 M を以下に示す。

$$M_{00} = \sum_x \sum_y I(x, y) \tag{2}$$

$$M_{10} = \sum_x \sum_y xI(x, y) \tag{3}$$

$$M_{01} = \sum_x \sum_y yI(x, y) \tag{4}$$

$$M_{11} = \sum_x \sum_y xyI(x, y) \tag{5}$$

$$M_{20} = \sum_x \sum_y x^2I(x, y) \tag{6}$$

$$M_{02} = \sum_x \sum_y y^2I(x, y) \tag{7}$$

これらのモーメント量から画像重心 (x<sub>c</sub>, y<sub>c</sub>)、Z 軸

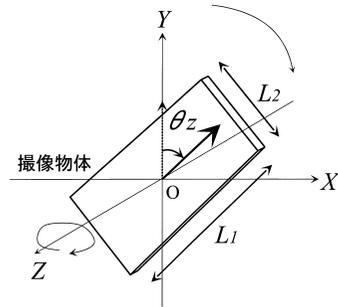


図 6 Z 軸周り回転姿勢の概念図  
Fig. 6 Object's rotational posture along Z-axis.

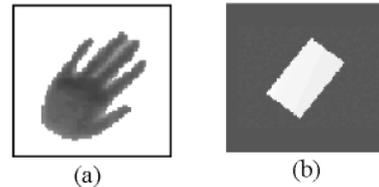


図 7 手の回転姿勢検出例。(a) 撮像例、(b) オブジェクト変形例  
Fig. 7 Hand's rotational posture detection example;  
(a) capture image, (b) transformed object.

周り回転量  $\theta_z$ 、対象物体を最も近似する四辺形物体の長軸  $L_1$ 、短軸  $L_2$  は以下のように求められる。

$$x_c = \frac{M_{10}}{M_{00}} \tag{8}$$

$$y_c = \frac{M_{01}}{M_{00}} \tag{9}$$

$$\theta_z = \frac{\arctan(b, (a - c))}{2} \tag{10}$$

$$L_1 = \sqrt{6(a + c + \sqrt{b^2 + (a - c)^2})} \tag{11}$$

$$L_2 = \sqrt{6(a + c - \sqrt{b^2 + (a - c)^2})} \tag{12}$$

ただし、

$$a = \frac{M_{20}}{M_{00}} - x_c^2 \tag{13}$$

$$b = 2 \left( \frac{M_{11}}{M_{00}} - x_c y_c \right) \tag{14}$$

$$c = \frac{M_{02}}{M_{00}} - y_c^2 \tag{15}$$

ここで、画像重心は撮像物体の位置、回転量は回転姿勢、 $L_1$ 、 $L_2$  は撮像物体の中心軸に対する画像分布の縦横比を表している。

以上で述べた Z 軸周りの回転姿勢量の概念図を図 6 に示す。図 7 は、手の撮像画像から、画像モーメントを用いて手の回転姿勢  $\theta_z$  を検出し、長軸の長さ  $L_1$ 、短軸の長さ  $L_2$  の四辺形オブジェクトを検出された角度  $\theta_z$  だけ Z 軸周りに回転させた例である。

### 3.3 奥行き方向傾き姿勢量の検出

1枚の画像から3次元的な姿勢情報を検出する手法は、大きく2つに分けられる。

1つ目は、あらかじめ検出したい姿勢の対象物体を撮像し、その撮像画像をテンプレート画像として登録しておくことで、撮像画像とのマッチングをとる手法である<sup>5),14)</sup>。手の奥行き方向傾き姿勢量の検出を考えた場合、カメラに対する手の傾き姿勢を変えて撮像した画像をテンプレートとして複数登録しておき、撮像画像とのパターンマッチングを行うことで、手の姿勢を検出することができる。また、あらかじめデータグローブなどのセンサでテンプレート画像における姿勢量を計測しておけば、手の姿勢量も同時に取得することができる。しかし、この手法では、どの程度のテンプレート数を用意すれば精度良く姿勢検出が行えるか見極めが難しいというに、テンプレート数が増えると、検出までの適合度の計算に多大な時間がかかってしまう。

一方、撮像物体を表現する3次元モデルを定義し、撮像画像にあてはめることで、モデルのパラメータを検出するという手法も存在する<sup>15),16)</sup>。手を対象にした傾き姿勢の場合、指先、指のつけね、指の関節といった特徴点の位置や角度を検出することになる。モーションプロセッサを使えば、各特徴点における奥行き距離値が取得できるため、手のひらを構成する3点を抽出し、姿勢を検出することができる。しかし、手を大きく回転させたり、指が曲がっていたりした場合、手前の指が奥の指を遮るなどで撮像されない領域が発生してしまい、対応点の探索が困難になってしまう。

先に述べたとおり、ここでは撮像対象として開いた状態の手を仮定している。そこで、1枚の板と見なせる手表面の3次元的な姿勢情報の検出法について以下に述べる。

手の表面を構成する板の傾き姿勢の検出には、撮像対象表面の勾配情報から生成される法線ベクトルの分布を利用する。まず、モーションプロセッサで撮像した距離画像を三次元空間上にマッピングする。画素 $(i, j)$ の画素値を $I(i, j)$ とした場合、各画素を $(i, j, I(i, j))$ で表現される3次元空間中の1頂点に対応づけることにする。次に、隣り合う画素、つまり隣り合う頂点で構成される各四辺形パッチの法線ベクトルを求める。まず、隣り合う3次元頂点情報からパッチを構成する各頂点の法線ベクトルを算出した後、それらを平均化することで、四辺形パッチの法線ベクトルを算出する(図8)。

そして、撮像物体を構成する全四辺形パッチにおい

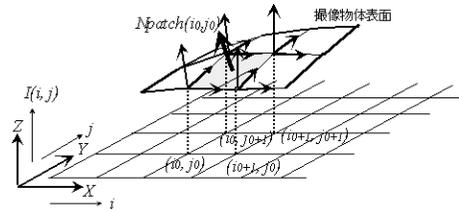


図8 距離画像の3次元表現と法線ベクトル  
Fig. 8 3D representation of depth image and normal vector of object surface.

て、上記で求めた法線ベクトルを平均化することで、撮像物体表面を代表する法線ベクトル成分を求める。2章で述べたモーションプロセッサの対象物体撮像原理より、各頂点の $z$ 値が0の四辺形パッチは背景と見なせるため、この段階で除外できる。四辺形パッチ $patch(i, j)$ における法線ベクトルを $N_{patch}$ 、撮像物体を構成する全パッチ数を $C_{patch}$ とすると、撮像物体の表面を構成する法線ベクトル $N_{surface}$ は以下のよう求められることができる。

$$N_{surface} = \left( \sum_{patch \subset Object} N_{patch} \right) / C_{patch} \quad (16)$$

最後に、撮像物体表面を代表する法線ベクトルから、 $X$ 軸、および $Y$ 軸周りの姿勢回転量を求める。姿勢回転量は撮像物体表面が正面を向いた姿勢、つまり法線ベクトルが $Z$ 軸単位ベクトル $U_z$ である姿勢からの $X$ 軸/ $Y$ 軸周りの回転変移量として求めることができる。法線ベクトル $N_{surface}$ の $XZ$ 平面への射影ベクトルを $N_{surfaceXZ}$ 、 $YZ$ 平面への射影ベクトルを $N_{surfaceYZ}$ とすると、 $X$ 軸周りの姿勢回転量 $\theta_x$ および $Y$ 軸周りの姿勢回転量 $\theta_y$ は以下のように求めることができる。

$$\theta_x = \arccos \left( \frac{N_{surfaceYZ} \cdot U_y}{|N_{surfaceYZ}|} \right) \quad (17)$$

$$\theta_y = \arccos \left( \frac{N_{surfaceXZ} \cdot U_x}{|N_{surfaceXZ}|} \right) \quad (18)$$

奥行き方向傾き姿勢量の概念図を図9に示す。図10は、手の撮像画像を対象に、法線ベクトルを用いた奥行き方向傾き姿勢量である $X$ 軸/ $Y$ 軸周りの回転量を検出し、四辺形オブジェクトに対し3次元回転処理を施したものを示している。手の“ひねり”や“かざし”といった姿勢が検出できていることが分かる。

### 3.4 評価実験と考察

3.3節で述べた手法の実効性を評価するため、奥行き方向の傾き角度の検出実験を行う。ここでは、反射率の様な薄い板(B5サイズ:W25×H19[cm])を

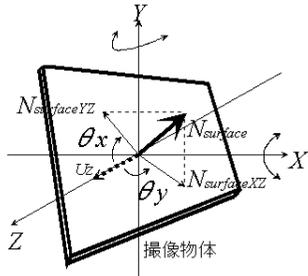


図 9 奥行き方向傾き姿勢の概念図

Fig. 9 Object's gradient posture along Z-axis.

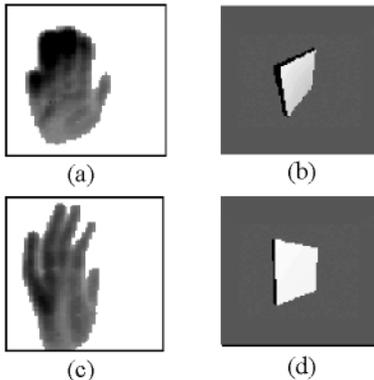


図 10 手の奥行き方向傾き姿勢検出例

- (a) 撮像例 1, (b) オブジェクト変形例 1,
- (c) 撮像例 2, (d) オブジェクト変形例 2

Fig. 10 Hand's gradient posture detection example;

- (a) capture image 1, (b) transformed object 1,
- (c) capture image 2, (d) transformed object 2.

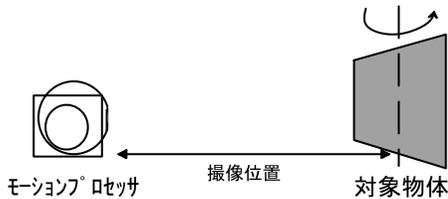


図 11 奥行き方向傾き姿勢検出実験の構成

Fig. 11 Experiment of depth gradient detection.

参照物体とした姿勢検出実験を行った。モーションプロセッサから 30 cm の距離に配置した対象物体を撮像し、撮像画像から対象物体の奥行き方向傾き角度の検出を行う(図 11)。実験では、板の断面が影響を与えない 0 度から 80 度までの範囲で 5 度刻みに Y 軸周りに回転させたときの姿勢検出を行う。

実験結果を図 12 に示す。横軸が板の傾き角度、縦軸が検出角度を表している。実験では、修正を加えていない撮像画像と、2 章で述べた画素値と距離値との間に存在する 2 乗減衰特性を補正し、線形特性に変換

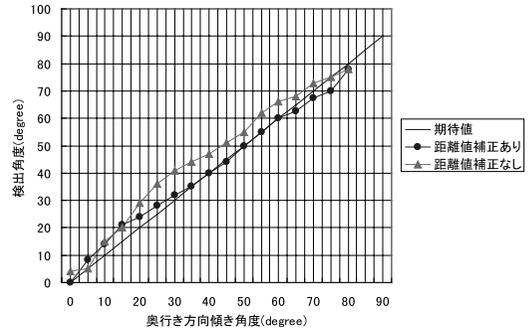


図 12 奥行き方向傾き姿勢の検出実験結果 (撮像位置 30 cm)

Fig. 12 Experimental result of depth gradient detection (target position is 30 cm ahead).

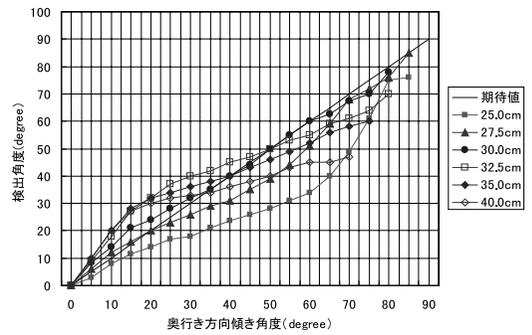


図 13 奥行き方向傾き姿勢の検出実験結果 (撮像位置可変)

Fig. 13 Experimental result of depth gradient detection (target position is variable).

した補正画像とに対して行った。実験の結果、撮像画像に対して補正処理を行うことで、誤差 3 度以内で検出できることが分かった。また、距離値の補正を行わない場合、板の傾き角度が 30 度あたりで検出誤差が最大 10 度程度に膨らむなど全体として大きめな値に出力されることから、距離画像の補正処理は有効な姿勢検出に必要であることが確認できる。

次に、同様な構成で、モーションプロセッサから対象物体までの距離を変えて測定を行った結果を図 13 に示す。

対象物体をモーションプロセッサに近づけた場合(撮像距離 25 cm や 27.5 cm の場合)、傾き角度を大きくする過程で測定値が小さく出る方向にずれの生じていることが分かる。これは、傾き角度が大きくなるにつれて撮像画像中に占める飽和画素の割合が増えるためと考えられる。モーションプロセッサは照射した近赤外光の反射光を撮像するという撮像原理に基づいているため、撮像距離の近い対象物体表面部分では、反射光が強く画素値が飽和してしまう。

一方、対象物体をモーションプロセッサから離れた

場合(撮像距離 35 cm や 40 cm の場合), 対象物体の傾き角度が小さな状態(30 度以下)で検出角度が最大 10 度程度大きめに出るのに対して, 対象物体の傾き角度が大きくなる(50 度以上)につれて, 検出角度が飽和する傾向にあることが分かる. モーションプロセッサから照射された近赤外光の強度は距離の 2 乗に反比例して急速に減衰するため, 撮像素子の単位画素値変化に対する絶対的な奥行き距離の分解能は撮像距離が伸びるにつれて低下している. 対象物体の傾き角度が大きくなっても検出角度が飽和してしまうのはこのためと考えられる. また, 傾き角度が小さな状態で検出角が大きくなることは, 照射する近赤外光の投影面に対する非一様性が原因である. つまり, モーションプロセッサから照射される光強度は光軸(撮像画像中心)から離れるにつれて同垂円上に減衰する傾向にある. 対象物体がモーションプロセッサから離れるにつれて撮像画像中の対象物体が占める画像領域も小さくなっていく. したがって, 少しの傾き角度変化でも撮像画像中の対象物体エッジ周辺に急速な反射光強度の減衰が発生し, 大きな傾き角度検出結果につながったものと考えられる.

また, 対象物体までの距離が 27.5 ~ 30.0 cm の範囲では, 25 度程度までの角度検出が誤差 5 度以内で精度良く行えることも分かる. 同様に対象物体までの距離が 30 ~ 35 cm の範囲では, 25 度 ~ 50 度程度までの角度検出が誤差 5 度以内で行えることから, 撮像物体の位置によって精度良く行える検出角度に違いのあることが分かった.

なお, 上記の評価実験で距離画像の撮像を含めた姿勢検出は毎秒 45 フレーム行っており, 実時間処理性能が確認できた.

#### 4. 3次元手姿勢判定によるインタラクション応用

3章で述べた撮像物体の3次元姿勢情報の検出処理を利用することで, 人間の手のジェスチャ操作によるインタラクション・システムの構築を行う. ここでは仮想物体操作と, パーチャル・ピンポンを紹介する.

##### 4.1 仮想物体操作

図 14 は, モーションプロセッサで撮像したユーザの手姿勢から検出される奥行き方向傾き量に応じて操作対象である CG オブジェクトを3次元回転させるアプリケーションの画面出力である.  $X$  軸, および  $Y$  軸周りの回転量として検出された手の傾き角度を, それぞれ操作対象の  $X$  軸および  $Y$  軸周りの姿勢回転角度にマッピングし, 3次元的回転動作を実現して



図 14 仮想物体操作画面

Fig. 14 Screen shot of virtual object manipulation.

いる. ここでは, ユーザへのフィードバックのため, モーションプロセッサで撮像した手の3次元モデル<sup>17)</sup>も, 画面上に実時間表示している. このことでユーザはマウスやキーボードでなく自分の手形状と操作対象の姿勢を見比べ確認しながら操作を行うことができる.

すでに, インターネット上のバーチャルモールなどでは3次元モデル化された商品サンプルを眺めることのできる3次元閲覧機能付きサイトも存在するが, 現状ではマウスによる操作が主体である. マウスを利用して3次元物体を回転させようとした場合, マウスの縦方向ドラッグあるいは横方向ドラッグで操作を行い, そのドラッグ量で操作対象の3次元回転量を制御している. この場合, マウスのドラッグ操作と操作対象の姿勢変化との関係は相対的であり, 一意に見たい姿勢を定義することができなかった. つまり, マウスの横方向, 縦方向操作の繰返しで見たい姿勢を試行錯誤で探す必要があった. このようにマウスによる2次元操作をCGオブジェクトの3次元姿勢操作に適用した場合, 直感的とはいえないがたい対応関係が内在する可能性があった.

一方, 本アプリケーションのように広げた手姿勢を入力インタフェースとして利用すれば, 手のひらの傾け具合でそれぞれの軸周りの姿勢回転量を一意に指定することができ, 3次元仮想空間内での  $X$  軸および  $Y$  軸周りの回転変位量を気にする必要はない. また, データグローブなどを利用した装着型インタフェースの利用も考えられるが, 3次元姿勢の検出には磁気センサが必要になるなど一般にシステムが大規模になる可能性があった. 撮像系デバイスによる本手法を用いれば, 装着デバイスに拘束されることなく小規模なシステム構成で, “手をかざす”, “ひねる” 感覚での動作入力が可能となり, 自然な形態での3次元操作が実現できる.

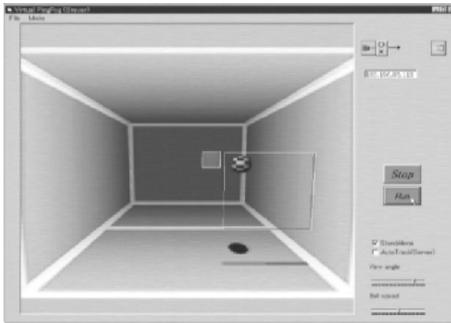


図 15 バーチャルピンポン操作画面

Fig. 15 Screen shot of virtual ping-pong.

#### 4.2 バーチャル・ピンポン

図 15 はユーザの手をラケットと見なし、計算機内に設定された 3 次元コート内で、ネットワーク上離れた相手と仮想ボールの打ち合いを行うピンポンゲーム・アプリケーションの操作画面である。モーションプロセッサにより撮像したユーザの手はラケットとして仮想コート内に 3 次元の板として表示されている。検出された手の撮像画像内重心位置からコート内のラケット位置、奥行き方向傾き姿勢からラケット面の角度を決定している。コート内に配置された仮想ボールは、コートの壁やラケットとぶつくと正反射するようにになっているため、ユーザが手を傾ければラケット面も傾きボールの反射軌道を制御することができるようにになっている。

非接触系デバイスを用いた場合、フィードバックの提供は重要である。ここでは、ユーザが視線を手に向けることなく、ボールの打ち返し動作に集中できるように、手の姿勢をラケットの 3 次元姿勢として実時間表示し、ボールが壁やラケットに衝突した際には、衝突音を効果音として発生させるようにしている。

マウスやゲームパッドなど従来型システムでこのようなインタラクションを実現しようとした場合、画面内ラケット位置の制御に十字キー入力を使うことが一般的である。しかし、ラケット面の制御といった詳細な操作入力を行おうとした場合、ボタンを併用する必要があるなど同時に制御できるパラメータ数には限界があった。さらにボタンの組合せでコマンド入力も可能だが、その組合せを覚える必要があり直感的な操作入力は困難であった。

手姿勢による入力インタフェースを利用すれば、ちょうどユーザが実世界で手を広げてボールに触れるのと同様な感覚でラケット位置とラケット面の傾き情報を同時に入力することができ、ストレスを感じるこ

となく自然で直感的な操作入力が行える。展示会でモーションプロセッサについて知らないユーザに操作を行ってもらったところ、手の位置と画面に表示されたラケット位置との対応関係を把握するまでに数秒かかったものの、慣れると打ち合いを行うことができるようになった。また、操作中、手の位置がモーションプロセッサに対し前後に振れる結果、最適な姿勢検出が保証されないケースも見られた。3.4 節でも述べたとおり、撮像位置と検出精度の間には密接な関係があるため、手の位置が適正な距離にあることをユーザにフィードバックして知らせる枠組みが今後は必要になってくると考えられる。

上記 2 つのインタラクション・アプリケーションでは、ユーザに対して手を開いた状態で操作してもらうように事前に説明を行っている。手を握ったり、手を丸めた状態では、検出結果が意図した手の姿勢を表していると保証できないためである。バーチャル・ピンポンのようなものの場合、ラケットを持ってゲームすることを考えるとラケットを持たず手を広げるという動作は自然な動作形態と考えられる。

なお、これらのアプリケーションはモーションプロセッサ SDK<sup>8)</sup> と Direct3D を用いた 3 次元画像処理ライブラリ<sup>17)</sup> を Visual Basic 上で利用し実装を行っている。モーションプロセッサによる距離画像の取得から、3 次元姿勢の検出・判定処理、アプリケーションデータの更新、画面の更新といった一連のシステム処理を毎秒 26 フレームで行っており、実時間インタラクション・システムとして実現できている(使用システム: PC/AT 互換機, Celeron 500 MHz, 128 MB RAM, Matrox G400)。

#### 5. おわりに

本論文では、近赤外光の照射と物体からの反射光の取得という撮像原理に基づくモーションプロセッサによって取得した距離画像から、画像モーメントにより Z 軸周りの回転姿勢量、および撮像物体表面の距離勾配から求められる法線ベクトルより、奥行き方向の傾き姿勢量を算出することで、撮像物体の 3 次元姿勢情報の検出が行えることを示し、奥行き方向の傾き姿勢検出実験からその有効性を確かめた。さらに、この手法を手の 3 次元姿勢情報の検出に適用することで、特殊な姿勢検出デバイスを装着することなく、非接触で仮想物体との実時間インタラクション操作が可能なアプリケーションを開発した。アプリケーションシステムの構築を通し、手の形状をそのまま用いた入力インタフェースは、マウスやキーボードによる操作入力

は不可能な直感的かつ多次元な情報操作を可能とすることを示した。手のひらに限らず、上半身や腕といった身体を構成する各部位の姿勢情報検出による身体的動作入力インタフェースの実現は、計算機との自然なインタラクション・システム構築にとってより重要なものになってくると考えられる。

### 参 考 文 献

- 1) 黒川隆夫：ノンバーバルインターフェース，オーム社 (1994).
- 2) Pavlovic, V.I., Sharma, R. and Huang, T.: Visual Interpretation of Hand Gestures for Human Computer Interaction: A Review, *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol.19, No.7, pp.677-695 (1997).
- 3) Kass, M., Witkin, A. and Terzopoulos, D.: Snakes: Active Contour Models, *International Journal of Computer Vision*, pp.321-331 (1988).
- 4) Cipolla, R., Okamoto, Y. and Kuno, Y.: Qualitative visual interpretation of 3D hand gestures using motion parallax, *Proc. MVA '92*, pp.477-482 (1992).
- 5) 渡辺孝彦, 李 七雨, 谷内田正彦：インタラクティブシステム構築のための動画像からの実時間ジェスチャ認識—仮想指揮システムへの応用，電子情報通信学会論文誌，Vol.J80-DII, No.6, pp.1571-1580 (1997).
- 6) Numazaki, S., et al.: A Kinetic and 3D Image Input Device, *CHI '98*, pp.237-238 (1998).
- 7) 沼崎俊一, 森下 明, 梅木直子, 土井美和子：ハンドアクションを用いた直感的な情報入力装置, *Interaction '99*, pp.177-184 (1999).
- 8) 山内康晋, 三原功雄, 梅木直子, 森下 明, 土井美和子：モーションプロセッサ用ソフトウェア開発キット, 情報処理学会第 58 回全国大会, No.4, pp.155-156 (1999).
- 9) 三原功雄, 森下 明, 梅木直子, 沼崎俊一, 山内康晋, 土井美和子：ハンドアクションを用いた直感的な情報入力装置, ヒューマン・インタフェース・シンポジウム, pp.301-304 (1998).
- 10) 岡 隆一, 西村拓一, 向井利朗：しぐさで伝える, 電子情報通信学会誌, Vol.82, pp.332-339 (1998).
- 11) 三原功雄, 山内康晋, 土井美和子：モーションプロセッサを用いたビジョン型インタフェースの構築, *WISS '99*, pp.33-42 (1999).
- 12) Freeman, W.T., et al.: Computer Vision for Interactive Computer Graphics, *IEEE Computer Graphics and Applications*, pp.33-42 (1998).
- 13) Horn, B.: *Robot Vision*, MIT press, Cambridge, Mass. (1986).
- 14) Darell, T. and Pentland, A.: Space-time gestures, *Proc. IJCAI'93* (1993).
- 15) Rehg, J. and Kanade, T.: Visual Tracking of High DOF Articulated Structures: An Application to Human Hand Tracking, *Computer Vision-ECCV'94*, Vol.801, pp.35-46 (1994).
- 16) 岩井儀雄, 八木康史, 谷内田正彦：単眼動画像からの手の 3 次元運動と位置の推定, 電子情報通信学会論文誌, Vol.J80-DII, No.1, pp.44-55 (1997).
- 17) 山内康晋, 三原功雄, 土井美和子：奥行き距離情報を用いた 3 次元可視化インタフェース, 情報処理学会第 59 回全国大会, No. 特 1, pp.71-72 (1999).

(平成 12 年 10 月 30 日受付)

(平成 13 年 4 月 6 日採録)



山内 康晋 (正会員)

1992 年東京大学工学部電子工学科卒業, 1994 年東京大学大学院工学系研究科電子工学専攻修士課程修了。同年 (株) 東芝に入社。リアルタイム 3 次元 CG 用 LSI, 入力デバイス, 触覚と画像を用いたコミュニケーション機器等 CG/HI に関する研究・開発に従事。



三原 功雄

1995 年東京工業大学工学部情報工学科卒業, 1997 年東京工業大学大学院情報理工学研究科計算工学専攻修士課程修了。同年 (株) 東芝に入社。入力デバイス, ノンバーバルインタフェースに関する研究・開発に従事。ACM, 電子情報通信学会, 人工知能学会各会員。



土井美和子 (正会員)

1979 年東京大学大学院工学系研究科修士課程修了。同年 (株) 東芝に入社。文書処理, CG/VR, 入力デバイス, モバイル機器を中心とした HI 研究・開発に従事。ACM, 電子情報通信学会, 計測自動制御学会各会員。