

LK-011

## パースペクティブウィンドウにおける鑑賞者の位置検出システム Face tracking system for "Perspective Window"

津田貴生  
Takao Tsuda

深谷崇史  
Takashi Fukaya

柴田正啓  
Masahiro Shibata

### 1. はじめに

鑑賞者の位置や動きに合わせて、表示画面がインタラクティブに変化するディスプレイシステムは、CAVE[1]を始めとする大規模システムから、自由視点テレビ[2]や視線検出によるインタラクティブな画面表示システム[3]などの小規模システムまで多数の研究が行われている。我々は、個人で鑑賞するテレビや PC などへの表示を目的に、鑑賞者の位置に応じた補正を行いながら、映像表示を行える"パースペクティブウィンドウ"を提案している[4]。

本稿では、パースペクティブウィンドウの概要とコンパクトで制約の少ない自由な鑑賞を可能とする鑑賞者の位置検出システムについて述べる。

### 2. パースペクティブウィンドウ

通常、映像は鑑賞者がモニターを正面から見ることを前提に制作されているため、斜めから映像を見る場合、鑑賞者は無意識のうちに映像に透視変換を掛けて補正を行っている。そこで我々は、鑑賞者の位置に応じた透視変換をあらかじめ映像に加えることによって、どの位置から見ても正しいパースペクティブで見ることが出来る"パースペクティブウィンドウ"を開発した。

鑑賞している様子を図1に示す。鑑賞者は、モニターを通してモニター背後にある実空間を見ることが出来る。描かれる映像は、実空間とまったく同じ空間配置を持つ仮想空間から生成され、実際の鑑賞者の位置と仮想カメラの位置を一致させてレンダリングを行う。しかし鑑賞者はモニターの正面から見ていないため、図2のようにレンダリング映像に対して、モニターのアスペクト比にあわせた長方形へ透視変換を行い、表示する。このことにより、鑑賞者は、正しいパースペクティブで鑑賞することができる。図1の鑑賞者の位置から見た映像を図3に示す。鑑賞者にはモニター背後の壁と表示映像がつながっているように見え

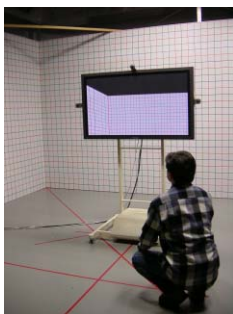


図1 パースペクティブウィンドウの概

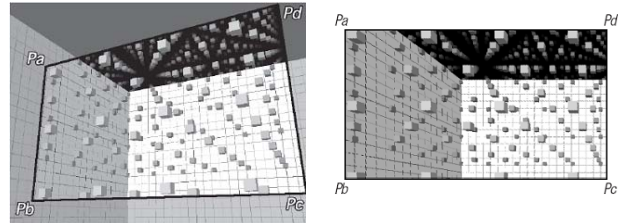


図2 左：仮想カメラから見たレンダリング映像  
右：モニターへの表示映像

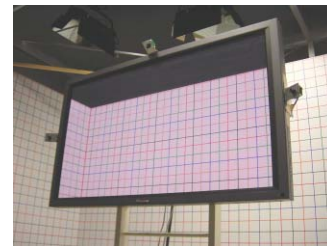


図3 鑑賞者の位置から見た映

るため、本システムは、鑑賞者にモニターを"窓"として実空間を見ている印象を与えることができる。そのため3次元CGで描かれた仮想空間の個人鑑賞用モニターとしての利用やHMD(Head Mounted Display)に代わるAR(Augmented Reality)のツールとしての可能性が考えられる。

### 3. 鑑賞者位置検出システム

表示画面がインタラクティブに変化するディスプレイシステムの位置検出システムは、通常HMDなどのセンサーの装着が必要であったり、顔または目の正確な位置検出を行うために、モニターの前に座って鑑賞するなどの移動範囲を限定したシステムがほとんどであった。パースペクティブウィンドウにおける位置検出システムは、センサーなどの装着を必要とせず、3m x 3m x 2m(前後、左右、上下)の範囲で自由に鑑賞位置を変えられ、画像処理による顔の位置検出を実装しているのが特長である。以下、顔の位置取得のための画像処理アルゴリズムと検出用カメラの設置位置や数などのシステム設計について述べる。

#### 3.1 位置検出の処理アルゴリズム

検出用カメラは、システムの小規模化と移動範囲の広域化という2つの相反する要素を考慮に入れて、モニターの上部と左右に3台設置した。カメラは、少ない台数で広い範囲の鑑賞者を検出するために焦点距離2.5mm(水平画角115度)のものを使用した。鑑賞者の顔の3次元位置は、カメラ映像内のつま先の位置から算出される床面上の立ち位置と、顔の位置から算出される高さにより得られる。処理の流れを図4に示す。上方カメラの映像を用いて、あらかじめ取得した背景映像と現在の映像の差を取り、しきい

値処理を行う．得られた鑑賞者のシルエット画像内のモニターに一番近い画素をつま先の位置とし，モニターからの前後方向の位置とする．また左右方向は，つま先位置から両足を含む程度の領域を設定し，その領域の重心位置を左右方向の位置とした．

次にこの鑑賞者の立ち位置に応じて，左右カメラのどちらか 1 台を選択し，その画面内で顔があると推定される領域内で顔検出処理を行う．処理手法は，つま先検出と同じく，背景差分処理により抽出された映像内での画面の最上部を鑑賞者の頭頂部とし，これを用いて，さらに探索領域を限定して，顔の検出を行なう．これは，できる限り探索範囲を小さくし高速化を図ることと，顔が下を向いたり，横を向いても，可能性の高い領域を探索できるためである．顔の検出処理は，H.Schneiderman らが開発した顔検出技術を応用した．この手法は，約 2000 枚の顔の学習画像から得られる統計的な顔領域の Wavelet 係数を基にした顔検出手法である．目と口が 20 画素以上あり， $\pm 30$  度以内の顔の向きが検出対象である．これにより，顔の高さを算出する．

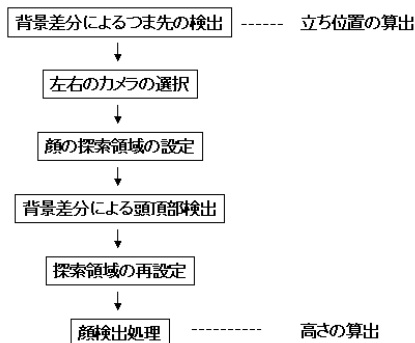
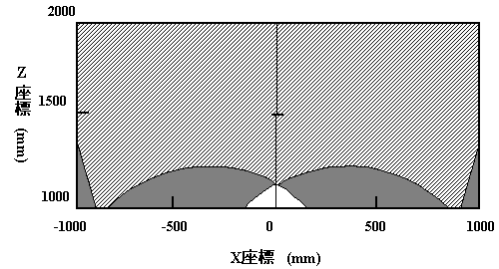


図 4 位置検出処理の流れ

### 3.2 カメラの設置位置

左右カメラの設置位置は，位置検出に用いる顔検出手法により制限される．使用した顔検出手法は，正面を 0 度として左右 30 度までしか顔として検出しないため，カメラに映る鑑賞者の顔の向きが  $\pm 30$  度以内の鑑賞範囲に限定される．カメラの高さは，鑑賞者が画面の中心を見ることを想定し，モニターの中心の左右の画面端に設置した．傾き角度は，上記の  $\pm 30$  度以内を維持でき，広い移動範囲をカバーするため，鉛直方向からモニター側に 40 度傾けて設置した．

今回，鑑賞者の検出には 3 台のカメラを用いたが，原理的には 2 台のカメラで 3 次元座標を算出できるステレオカメラシステムを使用することも可能である．ただパースペクティブウィンドウのような鑑賞システムの場合，カメラの台数を減らすメリットよりも鑑賞者の移動範囲を広げる効果の方がシステムの機能向上になると考え，立ち位置に応じて，左右どちらかのカメラを選択する方法（カメラ選択手法）を採用した．左右 2 台のカメラを用いて，ステレオカメラ処理とカメラ選択手法によって位置検出を行う場合の鑑賞者の移動範囲の比較を図 5 に示す．座標系は，モニターに向かって，左右が X 軸，上下が Y 軸，前後が Z 軸である．表示画面が大きく変化する X 方向に  $\pm 1$  m，Z 方向に 1 m から 2 m の移動範囲内では，カメラを選択して処理する方が，移動できる範囲が約 15% 増える．これは画



面付近での動きに伴う表示映像がダイナミックに変化するパースペクティブウィンドウの特長をより顕著に表現できることを示している．

### 3.3 CG 生成用 PC への出力

上記の処理から鑑賞者の目の 3 次元位置を CG 生成用 PC へ出力する．つま先位置検出から鑑賞者の XZ 平面の 2 次元位置，顔の位置検出から Y 座標（高さ）を得る．リアルタイムの CG 映像を生成するためには，1 秒あたり約 60 枚の映像をレンダリングする必要があるため，等速直線運動をモデル化したカルマンフィルタを掛け，動き予測を行いながら 60Hz のデータとなるように補間を行う．さらにスムージング処理を行ったデータを PC へ出力することにより，滑らかな動きの映像表示を行っている．

## 4. 試作システムによる検証

本検出システムと CG 生成 PC を用いてパースペクティブウィンドウの試作システムを構築した．違和感のない表示画像を生成するために，鑑賞者が動いた時の表示画面の応答性と検出精度の 2 点について，位置検出システムの性能評価を行った．

### 4.1 試作システムの概要

試作システムの概観図を図 6 に示す．ディスプレイの背後に 5 色の格子が描かれた壁を設置し，その壁とまったく同じ壁の CG モデルを仮想空間内に設置する．システムは鑑賞者の位置に応じて，画面を通して見えるであろう映像を画面に映し出す．鑑賞者からは，画面内に映っている格子と画面の外側に見える実際の壁の格子が繋がっているかのように見ることができる．

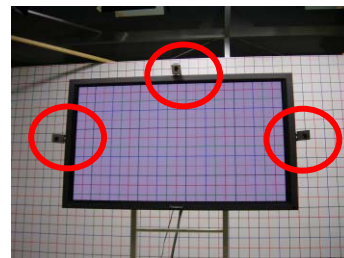


図 6 3 台のカメラを用いた位置検出システム

### 4.2 処理時間における検証

検出システムの処理は，つま先検出，頭頂部検出，顔検出を各 PC を割り当てて，並列に行っている．Pentium4-3GHz での 1 フレーム当たりの処理時間を表 1 に示す．CG 生成 PC へのデータ送信は，前後左右の移動と上下の移動

は別々に行っているため、左右方向の移動に伴う表示までの時間は、約 50msec 程度であるため、鑑賞者は遅れをあまり気にすることはない。上下方向は 160msec を超えるため、速い動きに対しては、遅れを伴って表示された。

表 1 処理時間

処理	処理時間
つま先検出	10 msec
頭頂部検出	20 msec
顔検出	100 msec
CG 生成	33 msec

### 4.3 キャリブレーション方法と検出精度の検証

#### (1) 上方カメラによるつま先位置検出

画面中心の校正とレンズ歪の除去を行ったモニター上部のカメラをピンホールカメラとみなし、画面内の座標から世界座標系における高さが 0 の平面への変換行列を求めることにより、キャリブレーションを行った。各位置に 5 秒間静止した時のつま先の位置の検出結果を表 2 に示す。校正点の 4 点 ((-1000,1000), (-1000,2000), (1000,1000), (1000,2000)) 以外の位置でも X 座標で ±10mm 以内、Z 座標で最大 40mm で測定できることを確認した。

表 2 立ち位置の検出結果 (単位はmm)

Z \ X	-1000	0	1000
1000	校正点	(2, 1006)	校正点
1500	(-1000, 1503)	(9, 1522)	(991, 1493)
2000	校正点	(-7, 2039)	校正点

#### (2) 左右カメラによる高さ検出

左右のカメラのキャリブレーションは、上方のカメラと同様に、画面座標から  $Z=z_1$  における世界座標系の XY 平面への変換行列を求めた。しかし鑑賞者は、カメラの光軸方向へも移動するため、カメラをピンホールカメラとみなし、得られた画面座標を変換行列で変換して得られた世界座標系の Y 座標  $y_1$  とカメラから被写体までの距離  $z_2$  から実際の高さ  $y_2$  を式 1 で算出した。

$$y_2 = \frac{z_2}{z_1} y_1 \quad (\text{式 1})$$

高さ方向の検出結果を表 3 に示す。鑑賞者は、立った状態で  $X = -1000, 0, 1000, Z = 1000, 1500, 2000$  の各点に立ち、測定を行った。目の高さは実測で 1670mm であった。どの地点でも、±20mm の範囲内で測定できた。誤差の原因は、立つ位置によって顔の上下の向きが異なることによるものが大きいと考えられる。

表 3 顔の高さの検出結果 (実測値 = 1670 mm)

Z \ X	-1000	0	1000
1000	1685.0	1674.5	1665.5
1500	1691.9	1675.5	1673.0
2000	1663.6	1668.5	1678.0

### 4.4 試作システム評価と考察

鑑賞者位置検出システムの処理結果を図 7 に示す。3 種類の画像処理とも正確に検出できていることがわかる。鑑賞者が座った状態でも、同じように検出することができた。得られた位置データを用いてパースペクティブウィンドウを操作した結果、ほぼモニターと壁の格子がほぼ一致して見ることができた。ただ上半身を前後左右のどちらかに少し動かす必要がある場所もあった。この原因は、顔検出結果は高さの検出にしか使用していないため、体の重心の位置により、左右方向に頭の推定位置がずれが生じるためと思われる。今後、高速なアルゴリズムや予測制御などを加えることによって左右方向の顔の位置を使用することにより、精度の向上が見込めると考えている。

### 5. まとめ

パースペクティブウィンドウにおける鑑賞者の位置検出システムを構築した。試作したパースペクティブウィンドウでは、上下方向の移動に伴う映像表示の遅れがあるものの、左右方向の移動に対しては、ほとんど違和感なく映像表示ができることを確認した。

この鑑賞者位置検出システムは、パースペクティブウィンドウに限らず、どのインタラクティブ表示装置にも応用ができ、現状のシステムでも 3m x 3m まで移動範囲を広げることが可能である。今後、他のインタラクティブな表示装置での応用を試みる予定である。

### 参考文献

- [1] C. Cruz-Neira: "Surround-Screen Projection-Based Virtual Reality: The Design and Implementation of the CAVE", Proc. of SIGGRAPH'93
- [2] 東尊弘ほか「自由視点テレビのための観視者の頭部検出」, 情報処理学会研究報告, 2003-HI-104, p.19-25, 2003
- [3] 飯尾淳ほか, 「顔の位置と頭部位置の 3 次元情報を利用したユーザインタフェースの試作」, 画像センシングシンポジウム SSI2002
- [4] 深谷崇史ほか, 「パースペクティブウィンドウ ~ 出演者の位置に対応した映像表示 ~」, IJADAD, vol.1, p.171-176, 2003
- [5] H. Schneiderman: "A Statistical Method for 3D Object Detection Applied to Faces and Cars", CVPR2000



図 7 鑑賞者検出システムによる画像処理結果 (上: つま先検出, 中: 頭頂部検出, 下: 顔検出)