社団法人 情報処理学会 研究報告
IPSJ SIG Technical Report

2006－HI－119 (15)
2006／7／7

おはじきインタフェース: ハイスピードカメラを用いた指を弾くジェスチャの認識

佐藤俊樹† 福地健太郎† 小池英樹†

この論文では，指で物を弾いて飛ばす動作を用いたゲーム等に用いるための力の入力インタフェースを提案する．本システムでは指で弾く動作をハイスピードカメラと高いフレームレートの画像認識を用いて認識する事で腕の角度や弾かれた指の速さを求め，力の強さと2次元的な方向を入力する手法として利用する．本システムを用いる事で，ユーザはスポーツ等で用いられる打つ・弾く動作を行う際の力加減と方向の入力を直感的に行う事が出来る．このシステムを用いて作成された3D ゴルフアプリケーションでは，より直感的なショットの強弱のコントロールを実現した．また「おはじき」という遊戯本来の遊び方をコンピュータ上で仮想的に再現するシステムとしてテーブル上に投影された仮想的なおはじきを指で弾き飛ばす事の出来るテーブルトップおはじきゲームシステムを作成した．このシステムでは，ユーザはテーブル上の仮想的なおはじきを，あたかもそこに存在しているかのように狙いをつけ指の力で弾き飛ばす事が出来る．

OHAJIKI Interface: Flicking Gesture Recognition with a High-Speed Camera

Toshiki Sato,† Kentaro Fukuchi† and Hideki Koike†

This paper described a new interaction technique that recognizes a finger flicking gesture to control power adjustment for a sports game, such as golf swing or batting on a baseball. Our system measures speed of the finger motion and direction of the gesture using a high-speed camera and a high frame rate image processing techniques. By using the system, users can adjust a power and an angle intuitively. We developed a 3D golf game using this interaction technique to provide an intuitive golf swing input, and a tabletop virtual ohajiki game which enables a user to flick a virtual marble projected on a table with his finger.

1. はじめに

これまでゴルフ等のゲームに用いられてきた力の入力手法としては，ゲームパッドや特殊な入力装置を用いた手法があった．しかしゲームパッドを用いた入力手法はゴルフ本来の動作である「物を打って遠くに飛ばす」動作からはかけ離れているため，より現実の動作に近い動きでの入力が好まれる傾向にあるゲームのユーザインタフェースとしては良い手法とはいえない．また特殊な入力装置を用いた手法も，設備が大掛かりなものになってしまうといった問題点があった．

ところで指を弾く動作は，我々にとってとても身近な動作である．平たいテーブル上に置かれたコインや小石を指で弾いて飛ばして遊んだ経験は誰にでもあると思われる．この動作は日本でも古くから「おはじき」等の遊戯の中で親しまれてきた．この指を弾いて物を飛ばす動作は，野球でバットでボールを打つ動作やゴルフでショットを打つの動作と「物を弾き飛ばす」という面で同じ性質を持っていると考えられる．そこで本研究では「指を弾く動作」をゲーム等で利用可能な力の入力インタフェースとして利用する手法を提案する．

2. おはじきインタフェース

2.1 「指を弾く」という動作

指を弾く動作とは人差し指や中指等の指を用いて小さな物体を弾き飛ばす動作であり，テーブルの上の小石やコインを弾いて遊んだり埃を弾き飛ばしたり等と日常的に行う動作でもある（図1）．日本でもこの動作は「おはじき」という遊戯で古くから親しまれてきた．この動作は指にかける力の強さによって弾き飛ばす強さを調節する事が出来，また飛ばす物に対して手を置く位置や角度を変える事で，物を飛ばす方向を指定する事も出来る．

† 電気通信大学大学院情報システム学研究科
Graduate School of Information Systems
The University of Electro-Communications

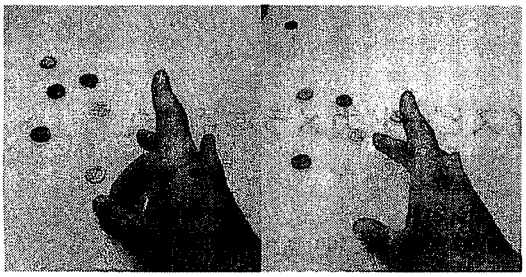

図 1　おはじきの動作はまず人差し指や中指を親指に引っ掛け，力をかける．次に親指にかけた指を外す事で指を弾く．

2.2　力の入力インタフェースとしての利用

今回着目した「指を弾く動作」は，力の強さと力の方向の 2 種類の要素を同時に取得することが出来る．これらの要素は，それぞれ指を弾く時の強さと，狙いを定めた方向から取得することが出来る．まず「指を弾く動作」は，実際に指に力を入れることで弾く強さを調節する動作である．特に「ゴルフ」や「バッティング」等の「ボールを打ち出す」動作があるスポーツに用いると，実際の力を用いて打ち出す強さを制御出来る点に加え，「物を弾き飛ばす動作」という点で共通の感覚があるためより直感的に操作出来ると考える．

また実際にゴルフクラブやバットを振回すより体力的にも疲れにくく，また安全でもあり，また多くの人が慣れ親しんでいる動作であるため特別な練習を必要としない．誰でも簡単に行う事が出来るというメリットがある．次に「狙いを付ける」という動作は方向を指示させるためには最も直感的な動作であると言え，ユーザは狙いをつける事で，力の方向を直感的に指示することが出来ると考える．

以上のように，「指を弾く動作」は力の強弱の調節と力の方向の指定を同時に，かつ直感的に行う事が出来る点で優れていると言える．

3.　ハイスピードカメラを用いたビジョンベースでの認識

人間の指は細くて小さく，また大きさや長さにも個人差がある．従って指に対し，位置や加速を検出するような複数のセンサを取り付ける事は難しい．

そのため本研究では画像認識を用いたビジョンベースの認識手法で指の動きを捉える事で指の速さを測定する．また指を弾く時の手，腕の位置や，弾かれるオブジェクトとの位置関係をカメラ画像から得る事で方向を取得する．

しかし，ビジョンベースの入力手法はユーザの動きを制限しないといった利点があるがカメラのスキャンレートの低さや遅延の影響で，専用のセンサーや特殊な入力装置を用いた手法と比べて認識精度や認識速度で劣るという問題があり，高速に動く指の動きの認識が難しいと言う欠点があった．そこで本研究では高速な動きを捕らえる事の出来るハイスピードカメラと，最小限の処理を行う事で高速化した画像処理を用いて指の動きを認識する手法を開発した．

4.　システム構成

本研究では，おはじきという動作を認識するインタフェースとして 2 種類の異なるシステムを作成した．まず一つ目のシステムは単純にユーザの指を弾く動作を見ることで，指の速さとその方向を測定し力の入力インタフェースとして用いる事を可能にする「おはじきインタフェース」である．このシステムを用いて得られた指の速さと方向は，例えばゴルフゲーム等の力の強弱の入力に用いる事が出来る．

もう一方のシステムは実際の「おはじき」という遊戯を仮想的に再現するシステムである．このシステムではテーブル上に配置された仮想的な玉を指で弾いて飛ばす事を可能にするシステムである．ユーザはテーブル上に投影された仮想的なおはじきを，あたかも実際のおはじきがそこにあるかのように指で弾く事で飛ばす事が出来，テーブル上で仮想的なおはじきゲームを体験する事が出来る．

この 2 つのシステムでは手の認識手法やシステム構成が異なるため，まず本論分では単純に指を弾いた時の指の速さと方向を測定する前者のシステム「おはじきインタフェース」について説明し，次にテーブル上で運用するアプリケーションとして後者のシステムを用いた「テーブルトップおはじきゲームシステム」を説明する．

5.　おはじきインタフェース

このシステムでは，ユーザの手の動きからユーザが何処を狙っているのかを判断し，また指を弾いた時の指の動きからユーザが弾いた指の力の強さを測定する事で力の強弱と力の 2 次元的な方向を測定するシステムである．

5.1　ハードウェア構成

本システムのハードウェア構成は図 2 のようになっている．カメラからの入力画像は画像処理用 PC 上で動作する認識システムで処理され，認識結果はネットワークを通してアプリケーションが動作するアプリケーション用 PC に送られる．

ユーザはテーブルの上に手を置き，テーブル上にある物体を指で弾く動作を行う事で入力を行う．カメラはユーザの手を見下ろす角度でテーブルの真上に固定され，テーブル上の手を撮影する．

5.2　認識の状態の定義

指を弾く動作でも，実際に指を弾くごく短い瞬間の動き以外の動作は，手を置く位置や角度を決めたり，狙いを定めたりする動作であり，これらの動作は比較

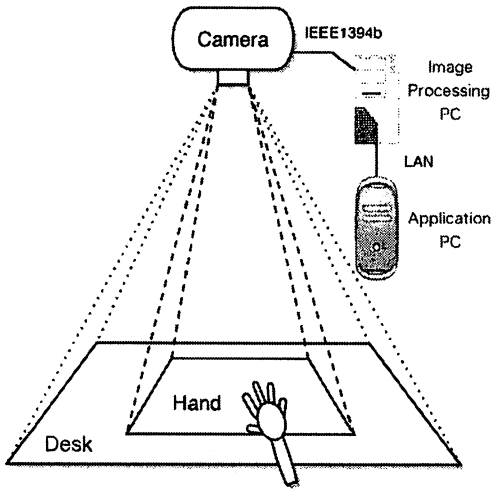

図 2 本システムは IEEE1394b 接続ハイスピードカメラ（PGR 社 DragonflyExpress），画像処理用の PC（Pentium4 3.2Ghz,1GB RAM），アプリケーション用 PC（同上）を用いて図のように構成される．ハイスピードカメラ DragonflyExpress は 640 × 480（8bit グレイスケール）でスキャンレート最大 200fps の撮影が可能で，さらに撮影範囲を絞る事で 320 × 240 で最大 350fps，160 × 120 で最大 420fps 等での撮影が可能である．また撮影した画像は IEEE1394b を通してリアルタイムに PC に転送する事が出来る．

的ゆっくりとした動きであると言える．そのため指を弾く動作で実際に高速な認識が必要なのは，指が弾かれた際のごく短い間の時間だけであるといえる．また本システムで用いているカメラは，撮影範囲を狭める事でより高いスキャンレートでの撮影が可能になる．

そこで本システムは，次のようないくつかの認識処理の段階を定義し，その時のユーザの動作に最適な認識処理が行えるようにした．これにより高速な動作の認識を行う必要がある場合には必要最低限の処理を最大限のスピードで実行する事が出来，また高速な認識を行う必要の無い動作の時は高速な認識を行わない代わりに手の位置の取得や他の認識システムとの組み合わせる等の補助的な認識を行う事も可能になった．

5.3 認識処理の詳細

認識処理は 4 つの状態で構成される．まず 1 つ目の処理は構えの検出処理である．これはテーブル上に手を置いたユーザが，指を弾こうと「手を構えた状態」を検出する処理である．2 つ目の状態は探索ウインドウの設定処理である．これはユーザが指を弾いた際に，指の速さを検出するための指が移動する領域を求める処理である．3 つ目は指先位置の検出である．この処理は探索ウインドウ内でユーザの指がどのように動いたのかを高速に検出する処理である．最後の 4 つ目の処理は指の速さの計算である．指先検出で検出した指の動きから，指の速さを計算する．

以下は，各処理ごとの詳しい処理の詳細を説明する．

(1) 構えの検出

まず最初にユーザが指を弾く動作を行おうとしているかどうかを検出する処理を行う．ここでのカメラの撮影範囲はテーブル全体である．

ユーザが指を弾こうとしているかどうかは，ユーザの手が「構え」状態にあるかどうかで判断する．ここでは構えの状態を親指と弾く時に用いる指とで円形の領域が作られる図 3 のような状態と定義し，背景差分と輪郭線検出による領域の分割を用いてこの円形領域を検出する事で構えの検出を行う事にした．

ユーザの手が構えの状態を取ると，これまでテーブル上の広範囲を撮影していたカメラの撮影範囲をユーザの手の周辺領域を撮影するように狭め，次の処理である「探索ウインドウの設定」に移行する．

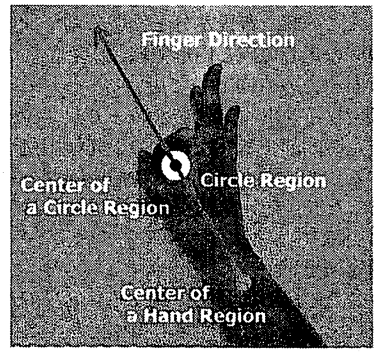

図 3 この図のように円形領域は人差し指，中指を用いて指を弾く際に自然に形作られる空間である．ショットの方向は手領域の重心位置と円形領域の重心位置を用いて決定される．

(2) 指先探索ウインドウの設定

次に，弾かれた指の検出を行うための探索ウインドウの位置を設定する処理を行う．このウインドウは画像処理を行う領域を限定させ，計算コストを軽減させるためのもので，指先の検出処理はこのウインドウ内に対してのみ行う．

ウインドウの位置はユーザの手の角度によって設置する場所が異なるため，適切な位置に設定するために，次のような手順でウインドウの位置を決定する．

1) 円形領域と手領域の重心の位置を基準とした位置に別の小領域（図 4 の点線の領域）を設ける．

2) 小領域を手の方向に対して少しずつ移動させ，小領域が手領域と接触した位置を求める．

3) この位置を基準として，指先探索ウインドウを設定する（図 4 の実線の領域）．以上の処理を毎フレーム行う事で，探索ウインドウの位置を決める．

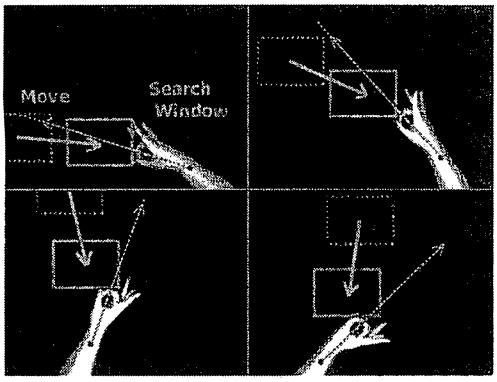

図 4　探索ウインドウは弾かれた指のみが領域内を通過する位置に設定されなければならない．小領域は初期位置（点線の領域）から数ピクセルずつ手の方向に向かって移動させ，手に接触する直前の位置まで移動した所で停止させる．

またこの時，背景差分と領域分割で抽出した手領域の重心位置と円形領域の中心位置とを結ぶ方向を力の方向として用いる．

探索ウインドウの位置が求まり，ユーザが手を一定時間静止させたら次の指先検出処理へ移行する．ここで手がカメラの撮影範囲外に出てしまったり，構え状態が崩れた場合は最初の状態に戻る．

(3)　指先検出

カメラの撮影範囲を探索ウインドウ内だけに切り替え，そこを通過する指に対して高速に指先位置の検出を行う（図 5）．指先の検出には背景差分と円形のテンプレート画像を用いたテンプレートマッチングで行い，この処理は指がウインドウ内に入ってから，再び外に出るまで毎フレーム行い，その都度指先位置を記録していく．指が領域外に出たら記録をやめ，速度の計算処理に移行する．

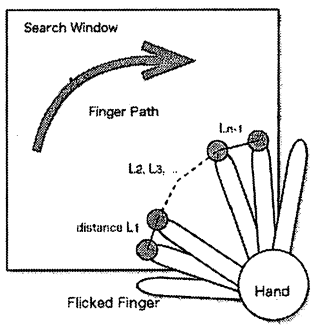

図 5　領域内に入った指は再び領域外に出ないといけない．また領域の境界上にある指は記録されない．

(4)　指の速度の計算

領域内を移動した指の位置の記録を用いて，次の式により指の速さを推定する．

$$V = \frac{\sum_{i=1}^{n-1} L_i}{T} \qquad (1)$$

ウインドウ内で最初に指が検出されてから最後に検出されるまでの時間を T[msec]，検出された n 番目と n+1 番目の指の間の距離を L_n [pixel] とすると指の速さ V [pixel/msec] は次の式 (1) のように求める．計算が終わると，速度や角度といった認識結果をアプリケーション PC に転送し，カメラの撮影範囲を切り替え最初の状態に戻る．

6. 3D ゴルフアプリケーション

本研究では「おはじきインタフェース」で検出した指の速さを利用した簡単なアプリケーションとして，仮想空間内のボールを指を使って弾く事の出来る 3D ゴルフゲームアプリケーションを作成した（図 6）．このアプリケーションは，3D 空間内で簡単なゴルフを行うゲームであり，プレイヤーは 3D 空間内で，ゴールの旗のある目的地に向けて指を弾く事でボールを飛ばす事が出来，いかに少ない回数でゴールの旗の近くにボールを寄せるかを競う．

6.1　システム構成

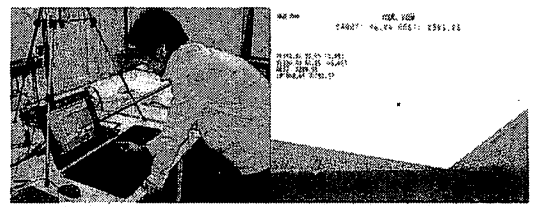

図 6　3D ゴルフゲームアプリケーション．今回のアプリケーションは 3 次元のゴルフゲームであるため，別に表示用のディスプレイを要する．またテーブル全体を使う必要はないため，写真のような小さな認識領域があれば良い．

ゴルフではボールが旗から遠く離れている場合は力強いショットが要求され，ボールが旗に近い位置にある場合は力のコントロールが必要な正確なショットが要求される．

このアプリケーションでは単純に指の速さだけを用い，ボールを飛ばす方角・角度の設定は手の位置を用いて設定する．つまりユーザはテーブル上で手を上下，左右方向に移動させることで角度，方角を決定する．そして指を弾く事で設定した方向に向けてショットを打つことが出来る．ショットの強さはプレイヤーの指を弾く強さによって変化する．

7. テーブルトップおはじきゲームシステム

次に本研究ではこのシステムをテーブル上での運用

に発展させ，プロジェクタによりテーブル上に投影された仮想的なおはじきを指で弾く事が出来るテーブルトップおはじきゲームシステムを作成した．

7.1 テーブル上に投影された仮想的な玉を弾く

おはじきという遊戯は，本来，床やテーブル上に置かれたおはじきやボールを指で弾き飛ばして遊ぶ遊戯である．そこで本研究ではおはじきインタフェースを用いて，おはじき本来の遊び方であるテーブル上のおはじきを弾いて飛ばす事を再現し，テーブルトップゲームアプリケーションを作成した．

このアプリケーションでは3Dゴルフとは違い，ユーザは実際にテーブル上にプロジェクタによって投影された仮想的なおはじきの玉を，あたかも現実にそこにあるように自分の力で弾く事が出来る．この際，現実におはじきを行う場合と全く同様に手の置いた場所や指を弾く強さに応じておはじきを飛ばす角度，強さを調節する事が出来る．

7.2 関連研究

これまでの研究において，テーブル上の仮想的なボール（この場合パック）を弾いて飛ばすインタラクティブシステムとして，大島らのAR^{2}Hockey1)が挙げられる．AR^{2}Hockeyは赤外線LEDの埋め込まれたマレットを用いてテーブル上に広がる仮想空間内にあるパックを弾く事が出来，これにより仮想的なエアホッケーの対戦を行う事が出来るシステムである．プレイヤーはHMDを通してテーブル上を見る事で仮想空間上にあるパックを見る事が出来，カメラによって位置が取得出来る赤外線LEDが埋め込まれたマレットを使い，このパックを弾く事が出来る．

本研究のテーブルトップおはじきゲームシステムもこのAR^{2}Hockeyのような仮想的に映し出されたおはじきを現実の動作を使って弾くことの出来るインタラクティブシステムである．

7.3 ハードウェア構成

本システムのハードウェア構成は図7のようになる．画像処理プログラムが動作する画像処理用PCと，アプリケーションが動作するアプリケーション用PCからなるシステム構成は「おはじきインタフェース」と同じである．

手を認識するハイスピードカメラとアプリケーションの映像を投影するプロジェクタはそれぞれのPCに接続され3Dゴルフと違い，カメラの撮影領域とアプリケーションの画面表示領域が同一のテーブル上になる．

カメラはテーブル上方の天井に取り付けられ，カメラの視界はテーブル全面をカバーする．この時のカメラの撮影解像度は640 × 480である．

プロジェクタも同様にテーブル上方に固定され，テーブル全面に映像を投影する．テーブルにはユーザの手の他に，プロジェクタによって投影されたおはじきの画像やスコア等のテキスト情報が描画される．

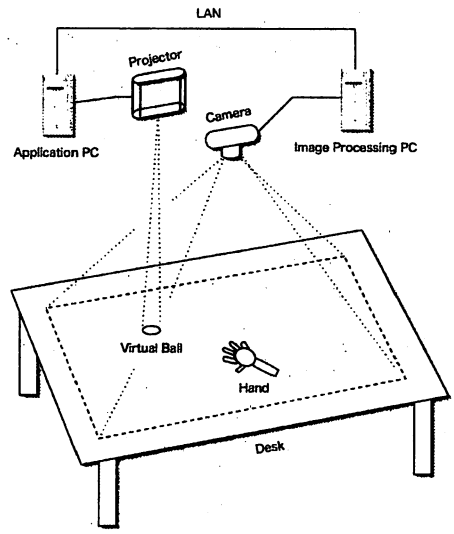

図7 ハードウェア構成．テーブルの色は背景差分を行う際に照明によって出来る手の影が影響しないように黒色にした．

7.4 テーブルトップでの認識処理

本システムではユーザが指を弾く際にはテーブル上に投影されたおはじきを狙って弾く事になる．従って指を弾く時はユーザ側がおはじきに手を近づけ，投影されたおはじきの上を指が通過するように指を弾く事が予想出来る．また方向については図8のようにユーザの構えた手の位置とおはじきの位置によって決まると仮定することが出来るため，ユーザが構えた際の円形領域と，おはじきの重心位置から容易に方向を求める事が出来ると考えられる．

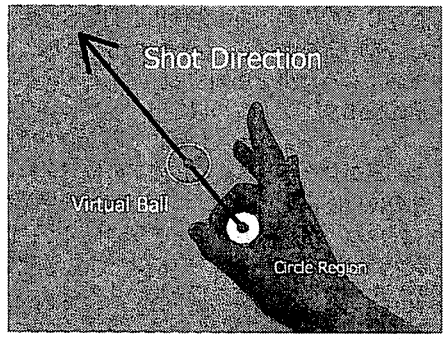

図8 力の方向の取得方法．おはじきの位置と円形領域の位置から求める．

従って手だけを認識対象として指を弾く動きを認識する場合と違い，ユーザの手から指の動く領域を予想し，そこに探索ウインドウを設定するといった処理を行う必要がない．またどの角度から指を弾いたかも考慮する必要が無いため，認識処理を簡略化することが出来る．

7.5 認識の流れ

このシステムではテーブル上のおはじきを制御するアプリケーションプログラムと手の認識を行う認識システムが処理に必要な情報を図10のように互いに交換しながら処理が行われる．この時，認識システムを制御し，ゲームの進行を行うのはアプリケーションプログラム側である．

(1) 認識プログラムによる手の位置の取得
ユーザの手がテーブル上で「構え」の状態にある時，認識システムは常に手の位置情報をアプリケーション側へ送信し続ける．認識システムが構え状態にある手の位置を取得する際のフレームレートは約200fpsである．アプリケーションプログラムは認識システムから送られてきた手の位置を用いて手の動きを常に監視し，手とおはじきの接近を検出する．
おはじきと手が一定距離に近づくと，ユーザがおはじきへ狙いを付ける動作に移行したとみなし，ユーザの狙う動作の検出処理へ移行する．

(2) ユーザの狙う動作の検出
アプリケーションプログラムにおいて手とおはじきとの接近が確認できたら，ユーザがテーブル上の何処に手を置き，おはじきをどの角度から弾こうか考えていると判断し，手の位置とおはじきの位置とを基準として図8のように角度を計算し，ユーザに方向が分かりやすいようにテーブルに矢印を表示する事で示す．
さらにこの時手が一定時間静止した事が確認できれば，ユーザがおはじきを弾く準備が出来たと判断し，アプリケーションプログラムは認識プログラムに現在のおはじきの位置を送信し，おはじきの位置での指先検出を開始させる．またこの時音を鳴らし，ユーザにも指を弾く事が可能になった事を通知する．

(3) 指先検出ウインドウの設定
アプリケーションからおはじきの位置を取得した認識プログラムは，現在の手の位置とおはじきの位置関係を判断し，ユーザがおはじきを指で弾いた場合に探索ウインドウの最適な位置を指が通過するように探索ウインドウの位置や指先検出の方向の設定を行う．

(4) 指先検出
指先検出は「おはじきインタフェース」と同様であり，検出された指の速さはアプリケーションに通知される．このウインドウ位置は，図9のように手の位置に対しておはじきを中心とした対称位置に設定する．
もしユーザが指を弾いた場合に適切に指の位置が取得出来なかった場合は「空振り」したとみなす．また一定時間指が弾かれなかった場合はユーザが狙う角度を変更したとみなし，おはじき検出モードへ戻る．

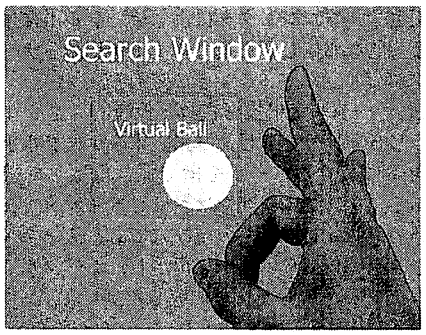

図9 ウインドウの設置位置

(5) おはじきへの力の適用
検出された指の速度（指の速さと手とおはじきの位置から計算した角度）はおはじきへと即座に反映され，おはじきは弾き飛ばされたように移動する．おはじきが弾かれると，処理は初期状態へ戻る．

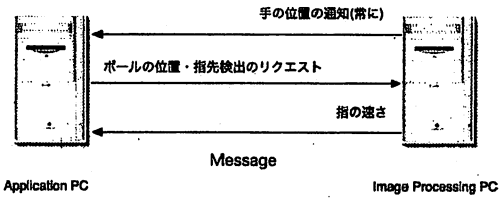

図10 認識プログラムとアプリケーションプログラム間のメッセージ通信

7.6 テーブル上に投影する事での認識への影響

手を認識するために必要な手領域の抽出処理は背景差分を用いて行っている．そのため，若干ではあるがテーブル上に投影された映像が背景差分に影響する事になる．しかし，抽出された領域の面積等で判断する事で手領域とノイズ領域との区別が可能である事から，実際には問題はない．

また指先を検出する際は，一瞬だけおはじきの描画を停止させる．こうする事で，指先探索ウインドウ内のおはじき画像による指先検出への影響は回避出来る．

7.7 カメラキャリブレーションについて

カメラの認識範囲とディスプレイの投影面にはずれが生じていたり，プロジェクタの投影面がカメラからみて歪んでいたりする場合がある．プロジェクタの投影面の歪みはプロジェクタ側である程度の補正が可能であるが，システムを設置する環境によってはカメラの認識範囲とディスプレイの投影範囲のサイズの違いは避けられない場合がある．

そこで認識システムで取得した手の位置をアプリケーション側で用いたり，逆にアプリケーション上で

のおはじきの位置を認識システムで持ちいる場合にはカメラ座標とディスプレイ座標の座標変換を行う必要がある．

この処理は適切に狙った角度を計算したり，指先位置の検出を行うためには重要な作業であり，もしカメラ座標とディスプレイ座標の変換に狂いやずれが生じていた場合は，適切に指が検出できなかったり，狙った角度と違った方向に飛んでいってしまう．本システムではパースペクティブ変換を用いてカメラ座標からディスプレイ座標への座標変換を行っている．

8. 評価と考察

8.1 フレームレートについて

本システムでは画像処理のコストを抑え出来るだけ高速に認識処理が行えるようにカメラの撮影範囲を各認識の場面にあわせて切り替えることで画像処理を行う画像の大きさを抑え高速化を図っている．またカメラは認識領域を限定する事でフレームレートを高めることが出来る．各認識状態での画像処理の適用範囲と認識のフレームレートは表 1 のようになっている．なお認識システムのフレームレートとカメラのスキャンレートは同値じである．またカメラのシャッタースピードは指の動きを捕らえるのに十分な 2.0msec に設定してある．この表を見ると，指先検出以外の処理

表 1 認識状態におけるフレームレート一覧

認識の状態	画像サイズ	フレームレート
構え検出	640 × 480	約 160 fps
ウインドウの設定	320 × 240	約 130 fps
指先位置検出	100 × 80	約 450 fps

でも 100fps 以上のフレームレートで動作可能なことが分かるため手の位置をポインティング等の操作にも十分対応出来ると思われる．

8.2 指の速度の検出精度について

本研究ではユーザの指を弾く動作の強弱を正しく認識出来ているのかを検証するために，簡単なユーザテストを行った．テストは本システムの使用方法を十分に説明し，理解させた本研究室の 4 人の学生に参加してもらった．ユーザにはそれぞれ 5 回指を弾いてもらい，この時「ゆっくり」弾いた場合から「思い切り」弾いた場合までの 5 段階の段階的な強弱をつけて指を弾くように要求し，それぞれの結果を記録した．この時の力のかけ具合はユーザにまかせる事にした．また事前にテンプレート半径の大きさ，円形領域の大きさ等の認識のパラメータはそれぞれのユーザの手に合うように調節した．このテストで得られた結果はグラフの通りである．

今回，ユーザの正確な指の速さを測定することが出来なかったため，ユーザ同士の速さの値の違いについては比較する事が出来なかった．しかしグラフからは

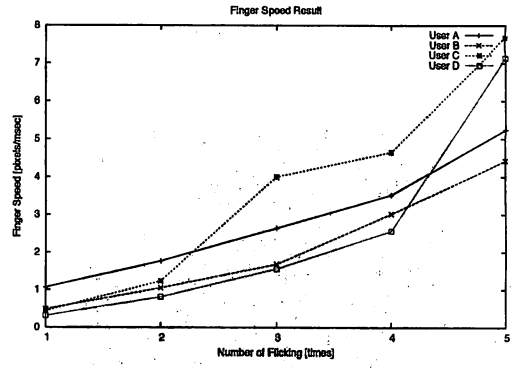

図 11 指の速さの測定結果．横軸は指を弾いた回数，縦軸は指の速さ [pixels/msec] である．

ユーザの指の速さの段階的な強弱が検出出来ている事はわかる．さらにゆっくり弾いた場合や，思い切り強く弾いた場合でも値が検出が出来ている事がわかる．

しかし今回用いた指先の速さの検出手法は，ウインドウを通過する指先の位置を求め，移動距離と経過時間を用いて速さを算出するというごく単純なものである．従って指先検出の精度は指先検出のフレームレートに大きく依存するものであると考えられ，より速く指を弾いた場合は指先の検出回数が低下し，これによって精度も低下する事が考えられる．今回の測定でも，1 回目に（ゆっくり）弾いた場合の検出回数が 20〜40 回なのに対し，3 回目になると 5〜10 回程度に減少し，5 回目に（思い切り）弾いた場合の指先の検出回数は 2〜4 回ほどであった．

これを解決するためには，より高速なカメラを用い，より高速に画像処理を行う事でフレームレートを高め，指先の検出回数を増加させるか，もしくは指の動きの数学的なモデルを用いる等の低フレームレートでも指先の移動が推定可能な全く別の手法を用いて指の速度を測定する必要があると考えられる．また，指先ウインドウの設定についても精度に大きく依存するものであると考えられる．指先がウインドウを大きく横切るか，端を小さく横切るかによって指先の精度が変わってくる．この問題を解決するには，手がどの角度でも指先が適切にウインドウを通過するようにウインドウ位置を設定する必要がある．以上のように精度の問題は今後の大きな課題であると言える．

9. 今後の課題と展望

指を弾く動作については今回のテストによって，ユーザにより指を弾く時の動作に様々な癖がある事が分かった．中指ではなく人差し指で弾く人もいれば，手を握るように指を弾く人もいる（図 12）．これは指を弾く動作にスポーツの動作のような標準的な動作が存在せず，ユーザは各々自分の好みの方法で指を弾く動作を

行う事が原因だと思われる．誰にでも扱えるシステムにするには，これらの癖を適切に判断して認識を行う必要があると考える．

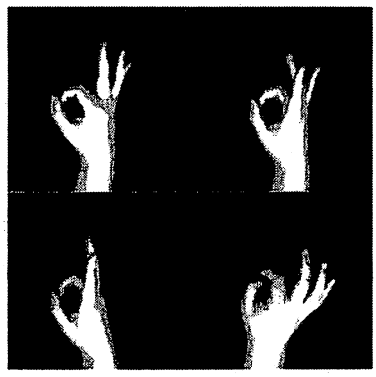

図 12　弾く指の種類での構え方の違い．それぞれ人差し指（左上），中指（右上），薬指（左下），親指（右下）

また今回，指先検出へ以降するタイミングはユーザの手が一定時間静止した事を認識していたが，ユーザがより自然に弾く動作に移行出来るためには，ユーザが指を弾こうとする瞬間の動きを検出出来る事が望ましい．ハイスピードカメラを用いた場合，ユーザの手の僅かな動きでも捉える事が可能であるため，例えば弾く指が親指から離れた瞬間が検出できれば，自然に指先検出へ移行出来ると考える．

今回実装を行ったシステムでは，ユーザはテーブルの一方向から手をいれないといけないといった制限があった．今後はテーブルの左右からも手を入れる事が出来るようにしたい．

また「テーブルトップおはじきゲーム」については，

さらに今回実装したシステムではユーザはただテーブル上にあるおはじきを弾く事だけしか出来なかった．そのため実際にテーブルトップアプリケーションとして運用するにはよりゲーム性を高めるための拡張が必要である．

これには例えば次のような要素の追加が考えられる．

- 複数人での同時参加
  現在のシステムではプレイヤは一人，テーブル上のおはじきは 1 個であると仮定した設計になっている．複数人，複数おはじきの対応を行う事で，テーブル上の対戦プレイが可能になり，よりゲーム性が増すと考えられる．
  複数人対応を行う場合は，ユーザの識別を行う必要がある事に加え，ユーザが同時におはじきを弾こうとした場合に対応するために，認識システムを複数台用意する必要が考えられる．
- テーブル上の実物体との組み合わせ
  テーブル上に置かれたオブジェクト（例えば本など）を認識し，障害物として登録する事でおはじきが障害物に当たって跳ね返るといった要素を加える事を考えている．
  このような要素を加える事で，ユーザはより現実感を得られる考えられる．
- 手領域抽出手法の改善
  今回用いたハイスピードカメラはグレイスケールカメラである．このカメラの場合，手の抽出手法は画像処理のコストを考えれば背景差分以外に選択肢は無かったため，アプリケーションの画像をテーブルに投影する際に認識に影響がでる事が避けられなかった．
  この問題を解決する手法としては，カメラに赤外線フィルタを取り付け，赤外線ライトでテーブル上を照らす事で赤外線での手領域の抽出を行う手法が考えられる．この手法を用いる事で，プロジェクタによる投影の影響を考慮する必要が無くなるため，アプリケーションにおいて認識に縛られない描画が可能になる．
  またカラーのハイスピードカメラを用いるという手法も考えられる．カラーで撮影が出来れば，肌色を使った手領域の抽出が行えるため，投影の影響をある程度避ける事が出来ると考える．
- 「おはじき」のルールの再現
  実際のおはじきのルールを再現する事も考えている．

以上のような改善を行い，今後精度の面での評価や実際にユーザに遊んでもらう事での評価を行いたい．

参 考 文 献

1) T. Ohshima, K. Satoh, H. Yamamoto, and H. Tamura: AR Hockey: ACase Study of Collaborative Augmented Reality, Proc. Int. Conf. PatternRecognition, Vol.2, pp.1226–1229 (1998).