ハンドジェスチャ インタフェース技術

Hand Gesture Interface Technology

坂本　圭　■ SAKAMOTO Kei
大竹　敏史　■ OOTAKE Toshifumi
池　司　■ IKE Tsukasa
藤田　将洋　■ FUJITA Masahiro

パソコン（PC）上で，手の形状や動作（以下，ハンドジェスチャと記す）を認識し機器を制御するには，高性能な画像認識を実現する高度なアルゴリズムと，ユーザーにストレスを与えない高速処理技術が必要である。

東芝は，これまでのキーボードやマウス，リモコンとは異なり，カメラと画像認識技術を応用してデバイスフリーで操作できる，ハンドジェスチャ インタフェース技術を開発した。当社独自のハンドジェスチャ認識アルゴリズムを最適化し，メディアストリーミング処理プロセッサSpursEngine$_{TM}$とホストプロセッサの協調処理を行うことで，PCの操作性を確保したまま快適な応答性を実現する新しいヒューマンインタフェースを提供できる。

In order to control equipment by hand gesture image recognition on a PC, both high-performance algorithm for image recognition and stress-free high-speed data processing technology are required.

Toshiba has developed a new hand gesture interface using images captured by a video camera and an image recognition technology, which eliminates the need for interface devices such as a keyboard, mouse, or remote controller. We have been able to successfully achieve comfortable operation for users by implementing an optimized hand gesture recognition algorithm on our SpursEngine$_{TM}$ media processor, which supports the smooth handling of image recognition and processing by cooperative processing with the PC's host processor.

1　まえがき

ユーザーのハンドジェスチャを認識して機器を制御する，ハンドジェスチャ インタフェース技術をPC上で実現するためには，高い画像認識性能を達成する高度なアルゴリズムの開発と，操作者であるユーザーにストレスを与えない高速処理という二つの課題がある。

東芝は，独自のハンドジェスチャ認識アルゴリズムを新たに開発し，そのアルゴリズムをSpursEngine$_{TM}$に最適化することでこれらの課題を解決した。ここでは，この技術を用いて当社のAVノートPC Qosmio$_{TM}$ G50シリーズに搭載した，新しいPCのヒューマンインタフェースの概要と特長について述べる。

2　ハンドジェスチャ インタフェースの概要

2.1　手形状と画像認識処理

今回開発したハンドジェスチャ インタフェースでは，**図1**(a)に示す三つの手形状を基本として画像認識処理を行う。

基本的な動作は，マウス操作をもとに次のように定義している。PC搭載カメラのフレーム内で，①の手形状（FIST）の位置をカーソルポインタの位置とし，②の手形状（THUMB UP）が検出されるとクリック動作となる。ユーザーは，FISTの状態でクリックしたい目的の場所へカーソルを移動し，その場で親指を立ててTHUMB UPを認識させるとクリック動作が実行される。FISTは一番自然な手形状であり，人にとって楽な姿勢であるとともに個人差が少ないため，カーソル移動に割り当てた。③の手形状（PALM）では位置に関係なく，検出された際にAVアプリケーション実行時は一時停止，その他の場合はクリック動作の切替え（左クリック，右クリック，及びダブルクリック）を行う。

(a) 基本となる手形状

(b) コマンド，アプリケーション起動用パネル

図1. 手形状と操作パネル — ジェスチャ認識を行う三つの手形状のほか，よく使うコマンドやアプリケーションを実行するための操作パネルを別に用意している。

Basic hand objects and operation panel

以上のように，マウスによるカーソル操作の代替となっているが，ユーザーがよく使うコマンドについては別に操作パネルを用意して，簡単にコマンドを実行できるようにした（図1(b)）。パネルには，再生や一時停止などのメディア系のコマンド，音量やサスペンドなどホットキーに割り当てられているコマンド，及びアプリケーションを起動するためのボタンを用意した。

2.2 画像認識の処理フロー

ハンドジェスチャ インタフェースは，**図2**に示すように，大きく分けてホストプロセッサとSpursEngine™の処理で構成されている。計算量の多い画像認識はSpursEngine™で処理し，そのほかのカメラやユーザーインタフェース（UI）の処理はホストプロセッサで行っている。処理の流れは次のとおりである。カメラで撮影された画像データは，ホストプロセッサからSpursEngine™へ転送される。SpursEngine™ではハンドジェスチャ認識処理が行われ，認識結果がホストプロセッサへ戻される。認識結果はホストプロセッサで扱いやすいようにデータが加工され，アプリケーションで利用される。このように，計算量の多いハンドジェスチャ認識処理をSpursEngine™で行うためホストプロセッサの負荷が小さく，同時に実行するアプリケーションへの影響を少なくできる。

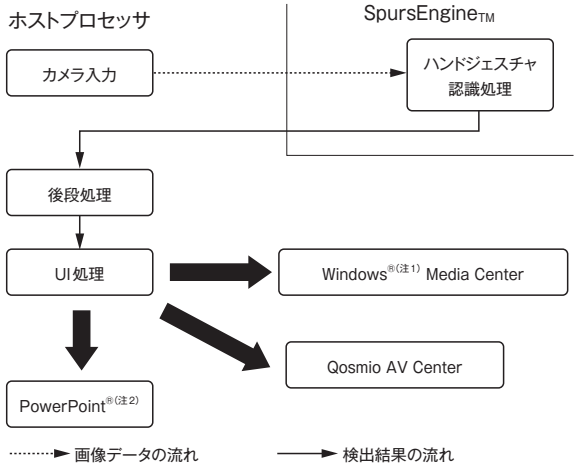

図2. ハンドジェスチャ インタフェースのブロック図 — 画像認識処理はSpursEngine™で処理し，そのほかのカメラやUIの処理はホストプロセッサ側で行っている。

Block diagram of hand gesture interface

3 ハンドジェスチャ認識アルゴリズム

3.1 画像からの手の認識

図2のハンドジェスチャの認識では，PCのディスプレイの上部中央に搭載されたカメラを用いて，1/30 s間隔で撮影した画像からユーザーの手を検出し，その形状及び位置が求められる。

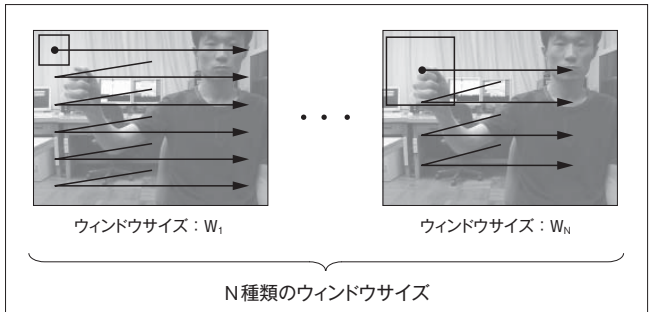

図3. 手の検出処理 — 各ウィンドウ内で検出処理を行い，特定の手形状の有無を識別する。様々なサイズの検出ウィンドウを用い，ユーザーの個人差やカメラからユーザーまでの距離によるサイズの変化に対応している。

Hand detection operation

手の検出では**図3**に示すように，カメラ画像全体を検出ウィンドウで走査し，検出ウィンドウ内の画像に対してそれが手であるかどうかを識別する。画像上の手のサイズはユーザーの個人差及びカメラからの距離で変わるため，様々なサイズの検出ウィンドウを用い，画像上の手のサイズによらず検出を可能にしている。

3.2 手の識別処理

手の識別処理では，手だけでなく背景にある物体も含まれる検出ウィンドウ内の画像について，その輝度値パターンを用いて検出対象の手かどうかを識別する。開発したジェスチャインタフェースは，不特定ユーザーによる一般家庭での利用を想定している。ユーザーの個人差による手形状の違いに加え，部屋の明るさや背景にある物体の違いなどによって，同じ手形状であっても検出ウィンドウ内の輝度値パターンは様々である。そこで，当社が考案した環境変化に強い顔識別技術[1]を手の識別処理に応用することによって，あらゆる環境下で様々なユーザーの手を適切に識別できるようにした。

この技術では，手形状ごとに対応する識別器を用いて検出ウィンドウ内の画像を分析することで手の識別処理を行う。

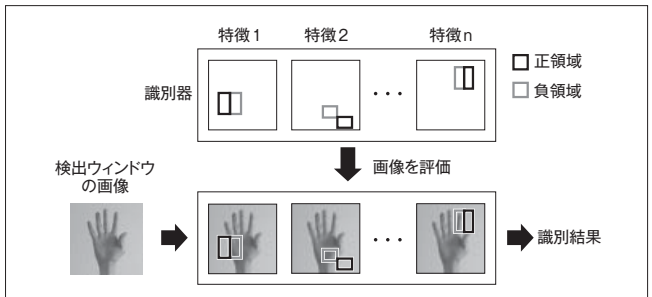

図4. 識別器による手の識別処理 — 学習によって選択された多数の特徴を使って，検出ウィンドウ内の画像が検出対象の手かどうかを識別する。

Hand discrimination using classifiers

（注1），（注2） Windows，PowerPointは，米国Microsoft Corporationの米国及びその他の国における商標又は登録商標。

識別器は，**図4**に示すような多数の特徴から構成され，各特徴は正と負の2種類の矩形（くけい）領域を持つ。識別処理では，これら2種類の矩形領域内の平均輝度値を算出したうえで，その差が特徴ごとに定義されたしきい値を超えているかどうかによって類似度を算出する。これら類似度の総和が識別基準値を超えていれば，検出対象の手と判断する。

なお，識別器を構成する際，様々な環境で撮影した年齢や，性別，人種が異なるサンプル画像を用いて識別に有効な特徴を選択することで，ユーザーの個人差や認識環境によらず安定した識別処理を実現している。

3.3　手の追跡処理

カーソルの移動やコマンド入力は認識された手の形や位置に基づいて行われるため，快適な操作性を実現するには手を高速で認識する必要がある。しかし，カメラ画像全体から手を検出する場合，計算量が非常に大きく，処理にかなりの時間を要する。一方，カメラの撮影間隔である1/30 s程度の時間では，ユーザーの手の位置はそれほど大きく変化しない。そこで，直前のカメラ画像における検出結果を利用して検出範囲及び検出サイズを制限することで，一度検出した手を高速に追跡することを可能にした（**図5**）。

まず，追跡対象の手を検出した際，その位置及び大きさを記憶するとともに，領域内の画像をテンプレート画像として保存する（図5①）。

次の時刻の画像では，初めに，直前の画像での検出領域を基準として手の探索範囲を決定し，探索範囲内からテンプレート画像に類似した画像を探索することによって手のだいたいの位置を推定する（図5②）。具体的には，探索範囲内に対してテンプレート画像と同サイズのウィンドウを走査し，ウィンドウ内の画像とテンプレート画像の輝度パターンの違いが最小になる位置を求める処理（テンプレートマッチング）により，手の位置を推定する。

更に，手の推定位置周辺に限定して手の識別処理を行うことで最終的な手領域を求めるとともに，得られた領域内の画像でテンプレート画像を更新する（図5③）。テンプレートマッチングでは，手の角度変化や認識環境などの影響によって検出位置にずれが生じるため，③のように手の推定位置を中心に識別処理対象範囲を設定し，範囲内で検出ウィンドウの位置を変化させながら3.2節で述べた手の識別処理を行う。

これらの処理結果を用いて最終的な手の位置を決定することで，ユーザーの個人差や認識環境によらない，より正確な手領域が得られる。また，識別の際に検出ウィンドウの大きさを若干変化させ，追跡中のユーザーとカメラ間の距離変化にも対応している。

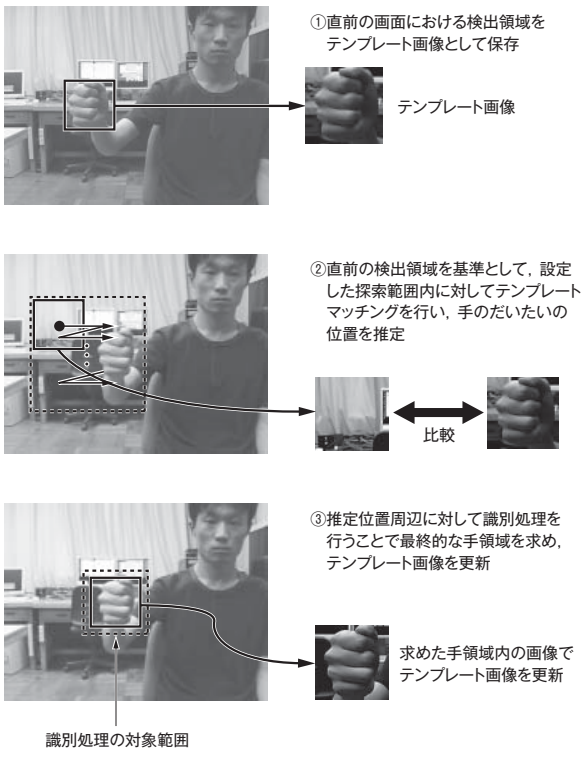

図5. 手の追跡処理 — 直前の検出結果を用いて推定した手の位置に限定した識別処理を行うことで，手領域を高速かつ高精度に求める。

Hand tracking operation

4　SpursEngine™ への適用と高速化

ここでは，3章で述べたハンドジェスチャ認識アルゴリズムをどのようにSpursEngine™へ適用し，リアルタイム応答性を実現するためにSpursEngine™への実装でどのような高速化を行ったかについて述べる。

4.1　ハンドジェスチャ認識処理の構成

SpursEngine™で行う認識処理は，次の三つの処理から構成される（**図6**）。

(1)　ハンドジェスチャ認識を行うカメラ入力画像をホストプロセッサから SpursEngine™へ転送する。

(2)　転送された画像をSpursEngine™に搭載されている演

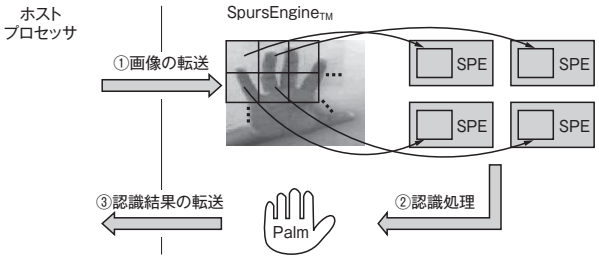

図6. ハンドジェスチャ認識処理 — SpursEngine™での認識処理は，カメラ画像の転送，SPEによる認識処理，及びホストプロセッサへの認識結果の転送の三つの処理から構成される。

Overview of hand gesture recognition in SpursEngine™

算プロセッサ（SPE：Synergistic Processor Element）で認識処理を行う。

(3) 認識結果の情報をホストプロセッサへ転送する。

(2)における処理の高速化を実現ために採用した並列化と最適化，及び処理全体を効率化するために導入した非同期処理について以下に述べる。

4.2 画面領域の並列処理

4.1節の(2)で行う認識処理アルゴリズムは，画面領域で並列化しやすい。SpursEngine$_{TM}$での実装にあたっては，入力画像を図6に示すような画面領域に分割し，複数のSPEでそれぞれ分割された画面領域を処理するようにしている。

4.3 認識アルゴリズムの最適化

リアルタイム性を実現するために，SPE へ移植した認識アルゴリズムに様々な最適化を施した。ここではそのなかから，処理の大部分を占める3.2節で述べた識別処理アルゴリズムを最適化するのに導入した，特別な最適化技法について述べる。

手の識別処理の演算は，図4に示すように多数の特徴を評価して判定を行う仕組みになっている。ここで，利用する識別器の構成は手形状ごとにあらかじめ決まっており，形状の識別処理実行時には不変である。また，識別器の構成が決まれば，演算のパスも入力画像に依存せずあらかじめ一意に決まる。この性質を利用して，辞書データからあらかじめ最適なSPEコードを生成しておき，実行時はそのSPEコードを実行することで同等の処理を実現することができる（**図7**）。この変換を手作業で行うのは非常に難しいので，今回独自の自動最適化コンパイラを作成して変換処理を行った。また，単純に変換するだけでなく SIMD（Single Instruction Multiple Data）の最適化を行うことで，より効率的で高速なコードを出力するようにしている。

以上の並列化及び最適化により，SPE の演算パワーを余すことなく引き出すことに成功した（**図8**）。図8(a)は，全画面探索や追跡処理のフレームも含んだ実際の利用条件下における1フレームの平均処理時間を示し，図8(b)は，負荷の大きい全画面探索時だけを考慮したときの1フレームの平均時間を示している。SpursEngine$_{TM}$での認識処理はIntel®(注3)版の実装

```
for (i=0;i < n; i++){
  rect = i番目の特徴領域の画素を取得
  sum += evaluate(rect)
}
```

等価な処理をSPEコードに自動変換
（SIMD最適化も行う）

```
…
vec_float4 tmp_s1_1  = spu_add (s_1_r1, s_1_r1) ;
vec_float4 tmp_s0_1  = spu_add (s_1_r0, s_1_r2) ;
vec_float4 s1_1      = spu_add (tmp_s1_1, area_1) ;
vec_float4 s0_1      = spu_add (tmp_s0_1, area_1) ;
sum                  = spu_add (sum, s0_1) ;
…
```

図7. 手の識別処理をSPEコードへ展開 — 認識アルゴリズムを実行時に評価するのではなく，あらかじめその演算パスをSPEコードへ変換しておくことで高速化を図る。

Translation of detection operation into equivalent synergistic processor element (SPE) code

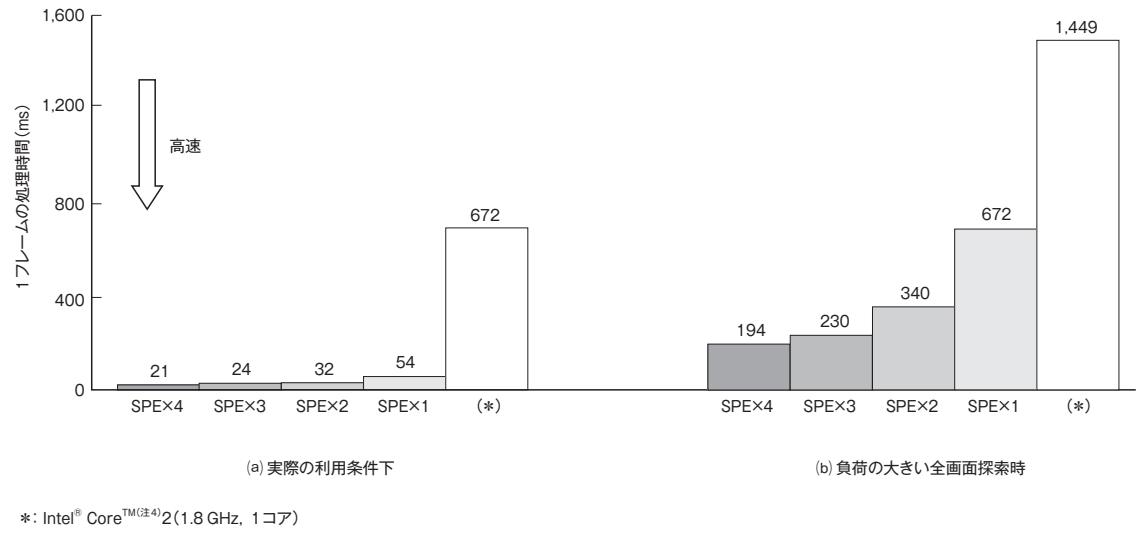

図8. ハンドジェスチャ認識処理時間 — SpursEngine$_{TM}$での画像の認識処理速度は，Intel®版の実装時に比べて約30倍速く，リアルタイムのハンドジェスチャ認識に十分対応でき，SPEの数に比例してパフォーマンスが向上する。

Hand gesture recognition performance results

(注3)，(注4) Intel，Intel Coreは，米国又はその他の国における米国Intel Corporation又は子会社の登録商標又は商標。

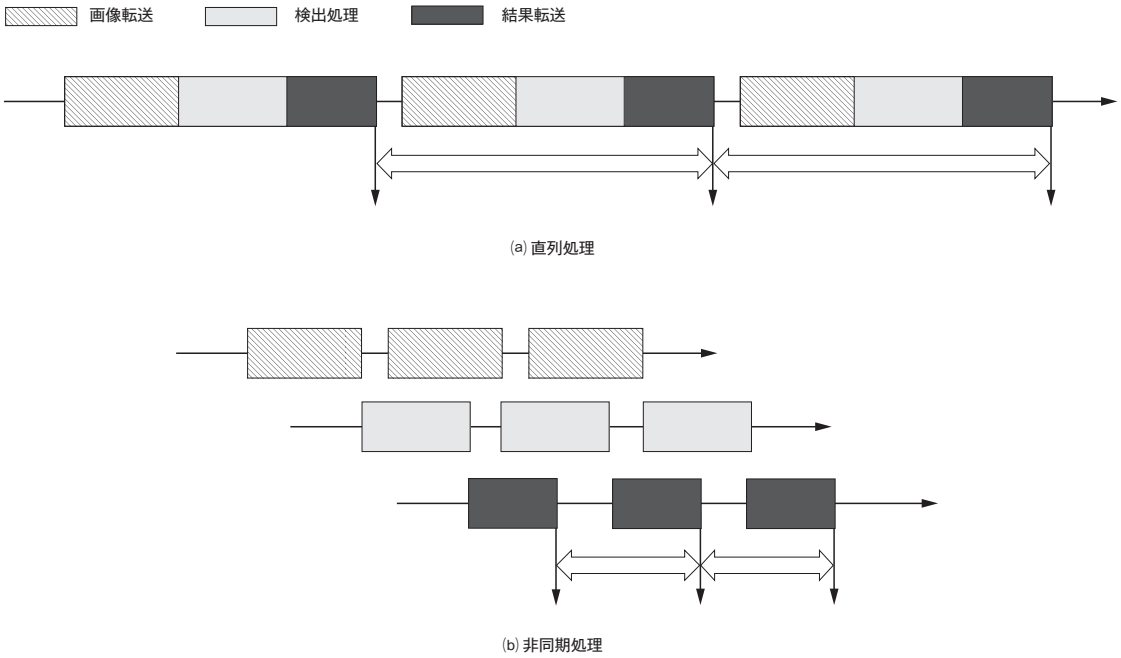

図9. 非同期処理のタイムライン図 — PCI-Expressの通信遅延を隠ぺいするために，処理を直列に行うのではなくて並列非同期に処理するようにしている。
Timeline visualization of asynchronous processing

時に比べて約30倍高速に処理することができ，リアルタイム（30フレーム/s以内の処理）でのハンドジェスチャ認識に十分なパフォーマンスを提供することができる。また，高負荷時は演算リソースをフルに利用する条件であり，並列化の効果でほぼSPEの数に比例するパフォーマンスを達成している。

4.4 データ転送の非同期処理

SpursEngine$_{TM}$はシリアル インタフェースのPCI-Express（Peripheral Component Interconnect-Express）で接続されているため，画像の転送（図6①）や認識結果の転送（図6③）の処理にそれぞれ十数ms程度の転送遅延が発生する。SpursEngine$_{TM}$側の認識処理（図6②）自体が速くても，PCI-Expressを介した通信がボトルネックになり応答性が低下すると問題である。そこで**図9**に示すように，画像転送，検出処理，及び認識結果の転送では，それぞれスレッド化して非同期で行うことで遅延を隠ぺいし，スループットを向上させて，このような遅延が認識処理の応答に影響を与えないようにしている。

5 あとがき

SpursEngine$_{TM}$を搭載したQosmio$_{TM}$の特長を生かせる，ハンドジェスチャソフトウェアの開発に使われている技術について述べた。今後はアルゴリズムの改良や新しいアルゴリズムの探求を進め，更に使いやすいユーザーインタフェースへと発展させていく。

文 献

(1) Mita, T., et al. “Joint Haar-like Features for Face Detection”. Proc. International Conference on Computer Vision. Beijing, China, 2005-10, IEEE. 2005, p.1619－1626.

坂本　圭　SAKAMOTO Kei
PC＆ネットワーク社　PC開発センター　PCソフトウェア設計第一部グループ長。PCソフトウェアの開発業務に従事。
PC Development Center

大竹　敏史　OOTAKE Toshifumi
PC＆ネットワーク社　PC開発センター　PCソフトウェア設計第一部。PCソフトウェアの開発業務に従事。
PC Development Center

池　司　IKE Tsukasa，Ph.D.
研究開発センター　マルチメディアラボラトリー研究主務，工博。画像認識に関する研究・開発に従事。IEEE，電子情報通信学会会員。
Multimedia Lab.

藤田　将洋　FUJITA Masahiro
セミコンダクター社　システムLSI事業部　先端SoC開発センター。Cell派生プロセッサへのソフトウェア最適化業務に従事。
System LSI Div.