3. 画像認識を応用したヒューマンインタフェース

3-2. PC向けユーザインタフェース

「ハンドジェスチャリモコン」

岡田 隆三^{†1}, 風間 久^{†2}, 鈴木 孝子^{†3}

キーワード ハンドジェスチャ, ユーザインタフェース, 画像認識, 手検出, 追跡

1. ま え が き

近年, PCは技術者や一部の先進的ユーザが使用する高級品から, 一般的なユーザが日々の生活の中で日常的に使用するツールへと急速に変貌を遂げている. それに伴って, ユーザインタフェースも, 慣れは必要だがすばやく正確な操作が行えるキーボードやマウスだけでなく, 操作方法が直感的でわかりやすい画面を直接触るタッチパネルタイプのものが普及するなど変革がなされている.

一般ユーザの日常のPCの用途として, メールや文書作成の他に, 映像や音楽などのメディアを楽しむことも多い.

このようなAV用途の場合, PCの前に座ってプログラム開発や文書作成の作業を行うのに向いたキーボードやマウスではなく, 少し離れたところから気軽にメディアを操作して視聴できるようなユーザインタフェースが望まれる. 通常, 遠隔操作にはリモコンが用いられ, メディア視聴向けのPCにはリモコンが付属している場合もあるが, 東芝ではリモコンすら必要のないデバイスフリーの操作により, さらに手軽な操作が実現できると考え, カメラでユーザの手の動き, すなわち, ハンドジェスチャを認識しPCを操作する「ハンドジェスチャリモコン」を開発し, AVノートPC "Qosmio™"に搭載した.

Fist

Thumb-up

Palm

(a) 認識する手形状

(b) V1の操作パネル

(c) V2の操作パネル

図1 操作方法とGUI

†1 株式会社東芝 研究開発センター

†2 株式会社東芝 セミコンダクター社 システムLSI事業部

†3 株式会社東芝 デジタルプロダクツ&ネットワーク社 PC開発センター

"User Interface for PC "Hand Gesture Remote Controller"" by Ryuzo Okada (Corporate R&D Center, Toshiba Corporation, Kawasaki), Hisashi Kazama (System LSI Division, Semiconductor Company, Toshiba Corporation, Kawasaki) and Takako Suzuki (PC Development Center, Digital Products & Network Company, Toshiba Corporation, Ome)

ハンドジェスチャリモコンを実現するには、カメラで撮影された映像から高い精度でユーザの手の動きを認識する高度な画像認識手法の開発と、高速処理によりユーザにストレスを与えない操作感を実現するという二つの課題がある。東芝では、独自のジェスチャ認識手法を開発し、メディア処理プロセッサSpursEngine™向けに最適化して実装することにより、これらの課題を解決した。本稿では、これらの技術を利用した新しいPCのインタフェースについて述べる。

2. ハンドジェスチャリモコンの構成

本章では、ハンドジェスチャリモコンの全体の構成について述べる。まず、操作方法やGUIなどのユーザインタフェースについて述べ、次にハードウェア構成と処理の概要について述べる。

2.1 ユーザインタフェース

ハンドジェスチャリモコンでは、PCに搭載されているカメラでユーザを撮影し、得られるビデオ映像をリアルタイム画像認識処理することによって、図1(a)のFist, Thumb-up, Palmの3種類の手形状を認識する。毎フレーム得られる画像内のFist手形状の位置をマウスポインタの位置(図1(b))や、アイコンのフォーカス位置(図1(c))とし、Thumb-up手形状を検出した際にクリック動作とする。ユーザは、Fist手形状の状態でカーソルやフォーカスを移動し、その場で親指を立ててThumb-up手形状をとるとクリックを行うことができる。Fist手形状は人にとって楽な姿勢であり、また個人差も少ないため、多用するカーソル移動に割当てた。Palm手形状に関しては位置に関係なく、検出された際にAVアプリケーションの一時停止、その他のアプリケーションではクリック動作の切替え(左クリック、右クリック、ダブルクリック)を行う。

以上のように基本はマウス操作の代替となっているが、よく使われるコマンドについては別途パネルを用意し、簡単にコマンドを実行できるようにした。ハンドジェスチャリモコンのバージョン1のパネル(図1(b))には、再生、一時停止などのメディア操作系のコマンドの他、音量、サスペンドなどのホットキーに割当てられているコマンド、アプリケーションを起動するためのボタンを用意した。バージョン2では、マウスカーソルを直接移動させるモードの他に、Fist手形状の移動に応じてアプリケーションやコマンドを示すアイコンのフォーカスを移動させることにより、マウスカーソルをアイコンに合わせる細かい操作を不要にして、操作性を向上させた(図1(c))。

2.2 処理フロー

ハンドジェスチャリモコンが利用するハードウェアは、CPUやメモリといった一般的なPCのハードウェアの他に、ユーザを撮影するためのカメラと、画像認識処理を実行するためのメディア処理プロセッサSpursEngine™である。

図2 処理フロー

これらのハードウェアを用いたハンドジェスチャリモコンの処理フローを図2に示す。大きく分けて、PCのCPUであるホストプロセッサでの処理と、SpursEngine™での処理に分かれる。計算量の多いジェスチャ認識処理については、SpursEngine™で処理してリアルタイム性を確保し、その他のUSBカメラからの画像入力処理、UIの処理については、ホストプロセッサで行う。

処理の流れとしては、ホストプロセッサでUSBカメラから画像をキャプチャし、SpursEngine™へ画像データを転送する。SpursEngine™では、3章で述べるジェスチャ認識処理を行い、認識結果をホストプロセッサへ転送して各アプリケーションで利用する。このように、計算量の多い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少なく、実行中のアプリケーションへの影響を少なくすることができる。

3. ハンドジェスチャ認識手法

ハンドジェスチャの認識は、ディスプレイ上部中央に搭載されたカメラを用いて1/30秒間隔で撮影した画像からユーザの手を検出し、その形状および位置を求めることによって行われる。

手の検出は、図3のように、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スライディングウィンドウ法により行う。ここで、画像上での手のサイズはユーザの個人差やカメラからの距離によって変わるため、さまざまなサイズの検出ウィンドウを用いる。さらに、画像全体から毎フレーム手を検出するのは処理コストが大きいため、一度手を検出すると次のフレームの画像からは、検出ウィンドウの位置およびサイズを前のフレームの検出結果周辺に制限する追跡処理を行うことにより、計算量を削減する。以下では、手の識別処理と追跡処理について詳しく述べる。

図3 手検出手法

3.1 手の識別処理

手の識別処理では、検出ウィンドウ内の画像について、その輝度パターンを用いてそれが検出対象の手形状かどうかを識別する。しかし、ハンドジェスチャリモコンでは、不特定ユーザによる一般家庭での利用を想定しているため、ユーザ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等により、同じ手形状であっても検出ウィンドウ内の輝度パターンはさまざまである。そこで、東芝が開発した高速で高性能なパターン認識手法¹⁾を手の識別処理に応用することにより、さまざまな環境下でユーザの手を適切に識別する。

識別のルールは、事前に収集しておいた多数の特定手形状 (Fist等) 画像と手以外のサンプル画像から学習する。個人差や照明変動などの変動は吸収し、両者の違いを際立たせるような効率的で正確な識別ルールをいかに見いだすかがポイントとなる。小さい計算量で正確に特定の手形状と手以外の画像を見分ける識別器を作るために、検出ウィンドウ内の局所領域から画像特徴を抽出する。図3のような、隣接する白い長方形と黒い長方形の組をHaar Wavelet特徴と呼び、白い部分と黒い部分でどちらの領域が明るい(暗い)かを画像特徴とする。明るさの大小関係しか見ないので、照明条件の変動や手の個人差に影響を受けにくい性質が得られる。このHaar Wavelet特徴を複数組合せて共起性を利用することで、大幅に性能が向上する。例えば、図3のように3組の特徴を配置すると、①AはBより明るい、②CはDより明るい、③EはFより明るい、という輝度パ

ーンの構造的な特徴が備わっているかどうかを評価できる。手以外の任意の画像には、このような共起性を満たす輝度パターンは少ないため、効率よく特定の手形状か否かを判定できる。

このような画像特徴の種類は、各長方形の位置や大きさ、組合せ方によって多数存在するが、どの画像特徴を識別に用いるかは、Boosting学習アルゴリズムを用いて決定する。具体的には、さまざまに位置や大きさを変えたHaar Wavelet特徴を生成し、その特徴のさまざまな組合せの中から、まず学習サンプルを識別する確率をもっとも高い組を選ぶ。選んだ組合せだけでは正しく識別できないサンプルも残るので、次に識別できなかったサンプルを識別できるように別の組を選ぶ。これを繰り返すことで、識別誤りの少ない強力な識別器を構成することができる。なお、このような学習を行う際、さまざまな環境で撮影したさまざまな年齢・性別・人種のサンプル画像を用い、識別に有効な特徴を選択することにより、ユーザの個人差や認識環境の影響を受けにくい安定な識別器を実現している。

3.2 手の追跡処理

ポインタの移動やコマンド入力、認識された手形状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ためにはカメラの撮影間隔である1/30秒毎に手を認識することが望ましい。画像全体から手を検出する場合、計算量が大きく処理時間を要するため、1/30秒程度の時間ではユーザの手的位置はそれほど大きく変化しないと仮定し、直前のフレームにおける検出結果を利用して検出ウィンドウの位置およびサイズを制限して、一度検出した手を高速に追跡する。

まず、認識対象の手形状を検出した際、その位置および大きさを記憶するとともに、領域内の画像を参照画像として保存する。次のフレームの画像では、直前のフレームで検出された手領域周辺に手の探索範囲を設定し、参照画像に類似した画像をテンプレートマッチングによって探索して手の位置を推定する。次に、テンプレートマッチングによって求めた手領域の周辺に限定して、前節で述べた手の識別処理を行うことにより最終的な手領域を求めるとともに、得られた手領域内の画像で参照画像を更新する。これによって、テンプレートマッチングで問題となる、参照画像が徐々に変化して別の物体を追跡してしまうドリフトの問題を回避している。

4. SpursEngine™によるリアルタイム処理

本章では、ハンドジェスチャリモコンのリアルタイム応答性を確保するために、3章で述べたハンドジェスチャ認識手法をSpursEngine™へどのように実装したかについて述べる。

4.1 SpursEngine™の概要

SpursEngine™は、東芝、IBM、ソニーが共同開発した

図4 SpursEngine™による高速処理

Cell Broadband Engine™アーキテクチャに基づくメディア処理プロセッサで、四つのSynergistic Processor Element (SPE) と呼ばれるプロセッサコアの他に、MPEG-2やH.264エンコーダ、デコーダを搭載している(図4(a))。四つのSPEは、並列に動作し、各SPEは4並列の32ビットSIMD演算を実行することができる。ハンドジェスチャリモコンでは、このようなSpursEngine™の並列演算性能を利用して、高速な画像認識処理を実現している。ハンドジェスチャリモコンで用いているジェスチャ認識処理において、ほとんどの処理時間を占めているのは、3.2節で述べた手の識別処理であるので、以下ではこの識別処理に関する並列化と最適化について述べる。

4.2 最適化手法

手の識別処理では、3章で述べたように検出ウィンドウを設定し、ウィンドウ内の画像について検出対象の手形状かどうかを識別する。各検出ウィンドウに対する処理は独立であるため、画像を複数の領域に分割して各領域に対する手の識別処理を別のSPEに割当てることにより、並列に実行することができる。SpursEngine™への実装では、四つのSPEを利用して4領域を並列処理している。

手の識別処理の演算は多数の画像特徴を評価して判定を行う仕組みになっている(3章参照)。利用する画像特徴量は、手形状ごとにあらかじめ学習時に決まるため、識別処理実行時には演算のパスも入力画像に依存せずに一意に決まる。そこで、学習時に選択した画像特徴から最適なSPEコードをあらかじめ生成し、識別時はそのSPEコードを実

行する。画像特徴の数は数千に及び、この変換を手作業で行うのは困難なため、SIMD最適化も行う自動最適化コンパイラを作成して変換処理を行った。

このような並列化と最適化により、SPEの演算性能を引き出した。図4(b)にSpursEngine™およびIntel CPU実装でのパフォーマンスを示す。「平均」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、「高負荷時」は負荷の高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™での認識処理はIntel CPU版実装に比べて30倍高速に処理することに成功し、リアルタイム(30fpsの処理)での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ている。

5. む す び

SpursEngine™搭載Qosmio™のハンドジェスチャリモコンは、近年のPCの利用環境の変化に伴うユーザインタフェースの変革を先取りし、AV視聴などより手軽なPCの操作を行うための先進的な取り組みである。本稿では、このハンドジェスチャリモコンの概要と、用いられている画像認識手法について述べ、リアルタイム応答性を確保するためのSpursEngine™への実装について述べた。今後はより性能の高いジェスチャ認識手法の開発を進め、さらに使いやすいユーザインタフェースへと発展させていく。

(2010年9月7日受付)

〔文 献〕

- 1) T. Mita, T. Kaneko, B. Stenger and O. Hori: "Discriminative Feature Co-occurrence Selection for Object Detection", IEEE Trans. PAMI, 30, 7, pp.1257-1269 (July 2008)

岡田 隆三 1999年、大阪大大学院工学研究科博士課程修了。同年、(株)東芝入社。2006年、カリフォルニア大学ロサンゼルス校客員研究員。現在、研究開発センターに勤務。コンピュータビジョンに関する研究に従事。博士(工学)。

風間 久 1992年、東京工業大学総合理工学研究科修了。同年、(株)東芝入社。コンピュータビジョン、画像認識に関する研究・開発、マルチメディアSoCソフトウェア開発に従事。

鈴木 孝子 1986年、東海大学工学部光学工学科卒業。同年、(株)東芝入社。音声認識/合成、ハンドジェスチャなどのアプリケーション設計・開発に従事。