

# 指先イメージセンサによる視覚障害者のためのタッチパネル操作支援システムの開発

研究代表者

山 脇 彰

九州工業大学大学院工学研究院 助教

## 1 はじめに

電子機器のインタフェースとして、タッチパネルの採用が広がっている。しかしながら、健常者とは異なり、指先の触感が操作に重要な視覚障害者にとって、表面が完全にフラットなタッチパネルでは電子機器の操作が困難であり、健常者と同様の恩恵を受けられない。

そこで、我々は、2016年にはタッチパネル全体の90%を占めると予想されている静電容量型タッチパネル[1]に対して、視覚障害者が訓練なしに自然な動作で操作でき、かつ、既存タッチパネル装置の物理的な変更もしなくて済むような操作支援システムの開発を目指している。

提案システムでは、ユーザが中指にカメラを、耳に音声案内用イヤホンを装着する。装置側では、タッチパネルに表示されたアイコンをカラーバーコードにする。ユーザが指先カメラによってタッチパネルをなぞると、カメラが撮影したカラーバーコードが解読され、音声案内がなされる。ユーザはその音声案内に従ってアイコンを人差し指によって押す。

指先カメラでタッチパネルをなぞる際に違和感を与えないため、指先カメラとタッチパネルは近接させる必要がある。そうでないと、カメラを付けた中指とタッチパネルを押す人差し指間に大きなギャップを要してしまう。ただし、カメラとタッチパネルを近接させると撮影したカラーバーコードは、ぼやけて色が混ざった状態で画像化される。

そのため、提案システムの実現に向けて、タッチパネルと近接したカメラの取得画像からコードを認識する処理フローを開発し、バーコードの構成法、認識精度、実行時間を評価検討する。さらに、提案システムの小型化や低消費電力化および高性能化を同時に達成するためにLSIでの実現を目指して提案処理フローのハードウェア化を行い、ハードウェアの回路規模や動作速度、性能を評価する。

以降、2章で関連研究について述べて本研究の意義を明らかにし、3章でシステムの概要を示す。その後、4章で処理フロー、5章でパイプラインハードウェアを概説し、6章で実験結果について述べる。最後に7章でむすぶ。

## 2 関連研究

視覚障害者に対して、タッチパネルの操作を支援するシステムは従来から様々な提案が行われている。

音の違いでユーザに画像の特徴や指のルートを知らせる研究にEdgeSonic[2]とTimbremap[3]がある。しかしながら、タッチパネルには、様々なアイコンが表示され、そのいずれかを指で触れて操作することが一般的であり、音の違いだけで多数のアイコンを識別することは困難と考えられる。

指の動作軌跡を利用した研究もある[4][5]。ただ、様々な操作ごとに指の動作軌跡や順番、および、タイミングを覚える必要があり、それらを正しく装置に認識させるにはある程度の訓練を要する。また、多数のアイコンから所望の一つを選択するような操作には適していない。

タッチパネルにアクチュエータを加え、その動作によって指先に感覚を伝える研究もある。例えば、パネルとしてアレイ状に多数のピンを配置した触覚ディスプレイ[6]や、タッチパネル表面に静電気を起こし、静電気の強さを変えることによって指先に質感を与える提案[7]である。しかしながら、特殊な触覚ディスプレイが必要であり、採用するには既存の装置を置き換える必要がある。さらに、指先に表面上の質感、例えば、ざらざらや凹凸感などを与えるだけであり、多数のアイコンから押すべきアイコンのみを正しく選択するような支援には向いていない。

上記とは異なり、文献[8]では、多数のアイコンから押すべきアイコンのみを訓練せず選択でき、かつ、既存装置を物理的に変更しなくて済む支援システムが研究開発されている。このシステムでは、ユーザはステレオカメラとヘッドフォンを装着する。そして、ステレオカメラによってタッチパネル上の指の位置とアイ

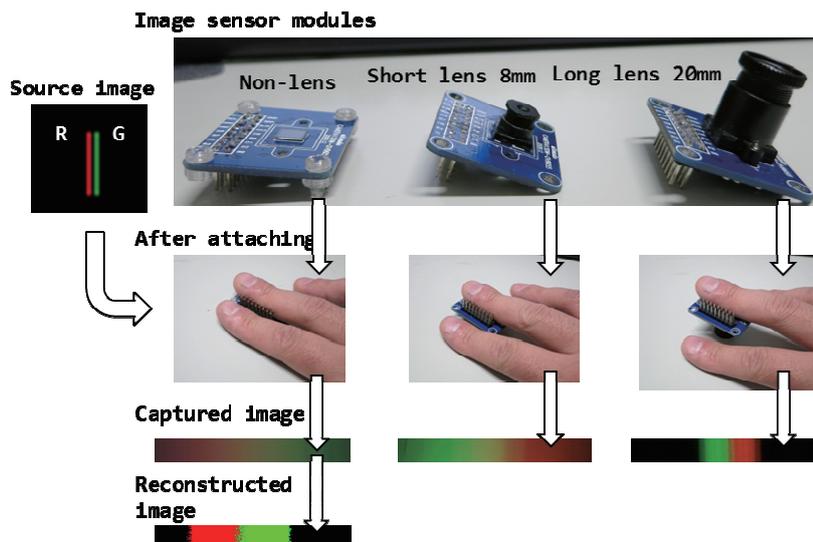


図 1 指先カメラによる撮影画像

コンの位置と種類を認識する(文献[8]ではアイコンは数字のみ). 認識した指先の位置と, アイコンの種類と位置にしたがって, 指が指しているアイコンの種類を音声によって知らせる. ただし, このシステムを使うにあたっては, ステレオカメラがタッチパネルと指全体を正しく映していることが前提条件であり, 視覚障害者にとってその確認は困難である.

それらの従来研究に対して, 我々は, タッチパネルの大部分を占める静電容量型を想定した新しいウェアラブルなタッチパネル操作支援システムを提案している[9]. 提案システムは, 文献[8]にシステムとは異なり, 視覚障害者にとって敷居の高い前提条件が存在しない. 既存のタッチパネル装置では, アイコンをカラーバーコードに変更すればよく, 既存装置を置き換える必要も無い. さらに, 何の訓練もなく, 指先でタッチパネルをなぞるといふ自然な動作によって, 多数のアイコンから押すべきアイコンのみを選べる.

ただし, タッチパネルをなぞる感覚をより実際に近づけるためには, 指とタッチパネルの距離を最大限短くする必要がある. その場合, 図1に示すように, 指先カメラで撮影したカラーバーコードの画像は, 色が混ざって非常にぼやけた画像になる. そのような画像からカラーバーコードを認識する処理フローの開発は提案システムの実現にとって重要な課題となり, 本研究の目的の一つである.

また, 小型で高性能かつ低消費電力なウェアラブルシステムの実現にあたって, 上記処理フローのハードウェア化も重要な課題である. さらに, ハードウェアを Application Specific Integrated Circuit (ASIC) で実現できれば, ASIC の量産効果によって提案システムの低価格化も実現しうる. 本研究では, 処理フローに最適化されたパイプラインハードウェアを開発し, その性能や回路規模, および, 動作周波数を明らかにすることも目的の一つである.

### 3 システム概要

図2は提案システムの構成を示す概念図である. カメラ部は中指の下, 制御部は手の甲や手首, イヤホン

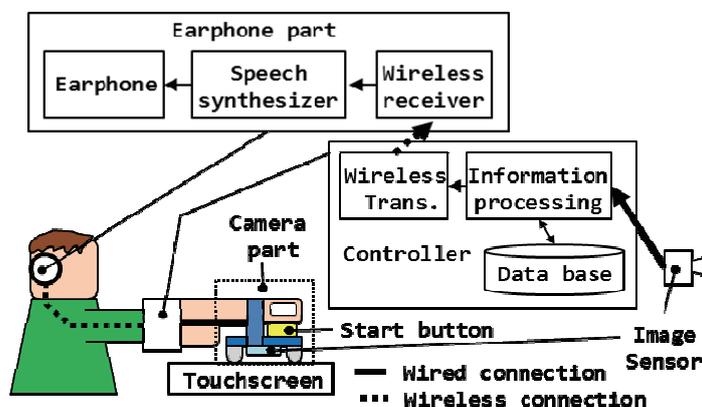


図 2 システム構成の概念図

部は耳に装着される。

カメラ部はイメージセンサとスタートボタンからなり、イメージセンサが直接タッチパネルに触れて画面を傷つけないように、スペーサが挿入される。ユーザは中指でタッチパネルに触り、中指をさらに押し込んでスタートボタンを押す。すると、カメラ部は画像を取得して、制御部にそれを送る。画像はサイズが大きいので有線を介して送信するが、カメラ部と制御部との距離は近く、ウェアラブル性を損ねることはない。

制御部はカメラ画像からカラーバーコードを認識し、ユーザが次にすべきことを音声案内によって知らせる。ユーザがアイコン（カラーバーコード）を押すと、アイコンが特定の色に変化し、制御部はボタンが押されたことを知り、アイコンの押下に応じて遷移する状態を保持する。状態遷移図や、状態の遷移条件、および、それらに対応した画面階層やアイコン、および、音声データなどがデータベースに記録されている。制御部は、アイコンの押下時に、そのデータベースを検索し、適切な音声案内を取得する。

制御部が選択した音声案内はイヤホン部に送られる。制御部が送信する音声案内データは単なるテキストデータであり、イヤホン部によって音声に合成される。テキストデータのサイズは小さいので、制御部からイヤホン部へは無線通信によって音声案内データが送られる。つまり、制御部とイヤホン部は、制御部とカメラ部と比較してより離されて配置されるが、無線によってウェアラブル性を損なわずに接続できる。

#### 4 処理フロー

図3は、開発したカラーバーコード認識処理のフローである。

タッチパネル平面のX-Y軸に対して、イメージセンサの撮像面のX-Y軸が傾いていた場合、取得画像中のカラーバーコードも傾く。そのような傾きが認識率に与える影響を低減するために、処理の簡単化を考慮して、単純にイメージセンサから得られた画像をある高さで切り抜く。

ぼやけた画像では、幅の異なる白黒帯の列によって識別する従来のバーコードは使用できない。そこで、カラーバーコードは、抵抗のカラーコードのように、固定長の色帯が横に複数本並んだ単純な構成とし、領域ごとの色の違いのみによってコードを識別する。また、取得画像は色が混ざった状態なので、帯の色には中間色を使用せず、R、G、または、Bのみとする。さらに、色帯ごとの分離を良くし、認識できるカラーバーコードの数を増やすために、色帯間に背景色（黒）の隙間を挿入する。

画像全体に色が広がった状態では、カラーバーコードの開始端と終了端がわからない。そこで、図3(1)において、閾値以下のR、G、Bを0（背景色、黒）にし、色の薄く広がった領域を除去する。その結果、開始端と終了端を表す黒色領域が画像の両端に現れる。

カラーバーコードは、基本的に画像内の横方向に色の異なる領域を抽出して検出される。色の違いを見つけるために、図3(2)において、RGB色空間からHSV色空間に変換する。そして、図3(3)において、色相の値に応じて、混ざった色をR、G、および、Bへマッピングし、RGB色空間に戻す。

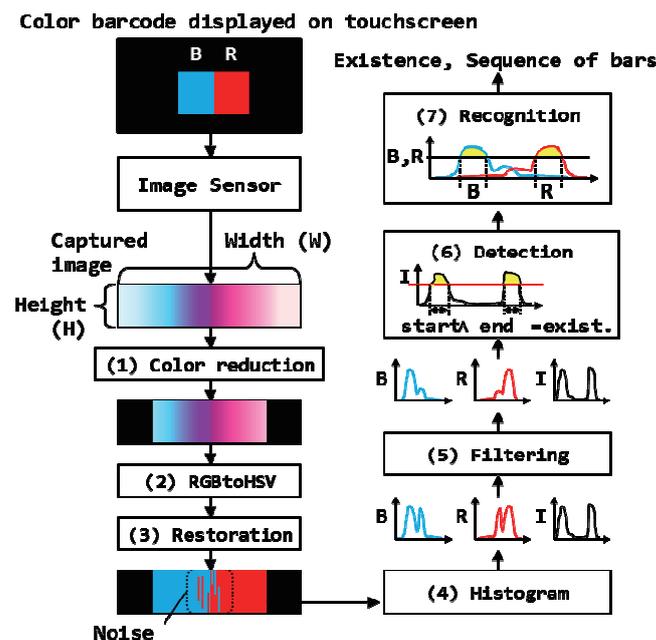


図3 処理フロー

その際、彩度が小さい、つまり、背景やグレーなどの画素に対して、値を強制的に0にする。その結果、R、G、および、Bが明確に分離された画像が復元される。ただし、色帯の境界に両側の帯の色がランダムに入り組んだノイズが発生する。

色帯の検出にはクラスタリングなどを含んだ複雑な処理が実施されるが[10]、処理の簡単化のために、ヒストグラムと閾値を使った単純な方法を採用する。図3(4)において、R、G、B、および、背景色(I)に対するヒストグラムを、復元画像を横方向にスキャンしながら生成する。そして、図3(5)において、色帯の境界に発生したノイズを除去する。

フィルタによって滑らかになったヒストグラムに対し、ヒストグラムに存在するピークを活用して色帯の組を検出する。図3(6)において、背景色のヒストグラムから、画像両端にある程度の幅を持ったピークが存在すれば、画像内にカラーバーが存在すると判断する。そして、図3(7)において、各色のヒストグラムを横方向にスキャンしながら、ある程度の幅を持ったピークが現れたら色帯としてマークする。もし、そのピークの幅が大きい場合、同一色の色帯が複数個並んでいると判断する。

## 5 パイプラインハードウェア

前章の処理フローは、処理の高速化やハードウェア化を考慮して固定小数点化される。つまり、本処理フローのプログラムはそのままハードウェア化できる。

ただし、プログラムを直接ハードウェア化すると、各処理が逐次を実施されることになり、ハードウェアの高速性を最大限引き出せない。一般的に、ハードウェアの高速性を引き出すには、プログラムに内在する並列性を抽出し、抽出された部分が同時に動くように設計する必要がある。その典型がパイプライン化であり、それによってハードウェアの性能を最大限引き出すことを試みる。

並列性の抽出にあたっては、当該プログラムに存在するデータ依存を解析する必要がある。当該プログラムには、概ね、ヒストグラム前後の処理間で、ヒストグラム上に解消できないデータ依存がある。つまり、前処理がヒストグラムを生成し終えるまで、後処理は実行を開始できない。よって、前処理と後処理は逐次に実行させる必要がある。ただし、前後の処理内は、各々、パイプライン化できる。

図4が本処理に最適化されたパイプラインハードウェアの概要である。図4に示すように、前処理は、ステージ1～ステージ28(S1～S28)にかけて、29段のパイプライン構成であり、後処理は、ステージ29～ステージ37(S29～S37)にかけて、9段のパイプライン構成である。灰色の実線がパイプラインステージの境界を意味し、各境界上にパイプラインレジスタが置かれる。これによって、前処理と後処理には1クロックご

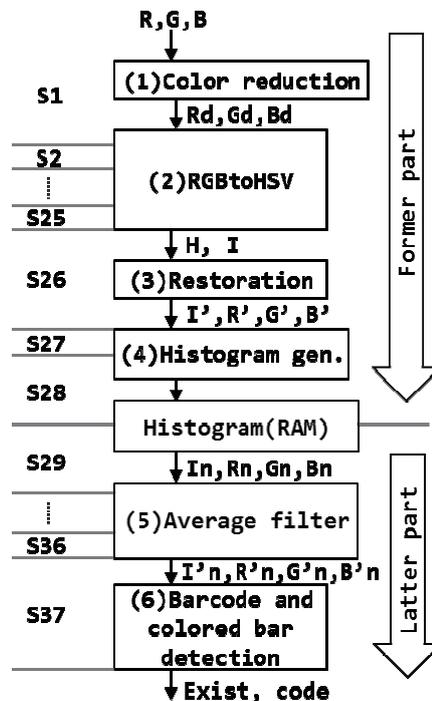


図4 パイプラインハードウェアの概

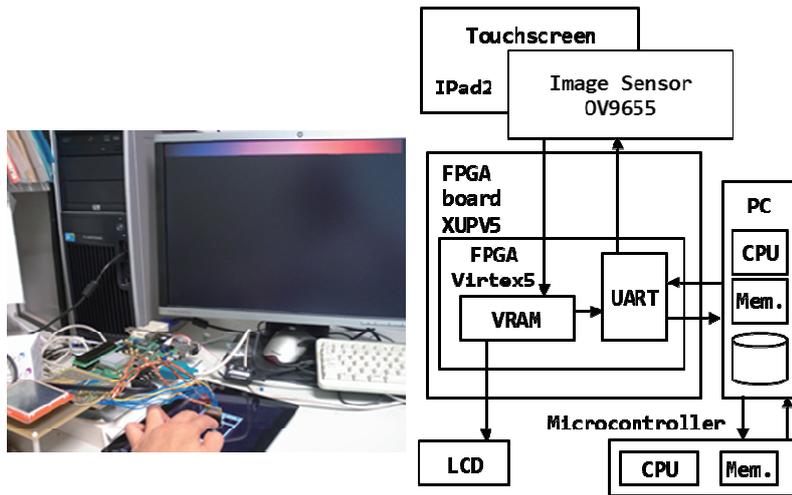


図 5 実験環境

とにデータを入力できる。そして、パイプライン段数と同じクロック数のレイテンシ後には、パイプラインが詰まった状態において、1クロックごとに出力結果が生成される。

イメージセンサは1フレームをラスタスキャンしながら、RGBからなる画素を図4のハードウェアへ入力していく。入力画素は流れ作業的にパイプラインステージ上を進んで行く。

## 6 実験および考察

### 6-1 実験環境

図5に実験環境の概要を示す。使用したCMOSイメージセンサはOmniVision社のOV9655であり、解像度は1280×1024、フレームレートは15fpsである。また、ホワイトバランスの調整や色ゲインの調整などの自動機能はすべてオフにした。OV9655からは取得画像のRGBデータが出力される。R, G, Bは、各々、5bit, 6bit, 5bitであり、1画素は合計して16bitである。

OV9655はXilinx社のXUPV5ボードに接続され、OV9655の出力画像はXilinx社のFPGA(Virtex5)を経由して、FPGA内のVRAMに格納される。FPGAの内蔵メモリの容量から、OV9655からの出力画像(1280×1024)のうち、真ん中の1024×64のみがVRAMに保存される。実験では、カラーバーコードの認識処理に対する基礎実験が目的であるため、VRAMに保存された静止画に対して処理を行う。VRAMはパーソナルコンピュータによって読み出され、1024×64の取得画像がファイルに保存される。

パーソナルコンピュータのメモリは4GB、プロセッサはAMD Athlon II X2 B26、動作周波数は3.2MHzである。OSは、Windows 7 enterprise 32ビットである。

カラーバーコードの認識処理はCプログラムとして実装され、cygwin上のgcc4.5.3によってコンパイルされる。最適化オプションは-O3である。

カラーバーコードはアップル社のiPad2上に表示され、OV9655をiPad2に接触させながらその画像が取得される。色帯は1ビット幅とし、iPad2が自動的に画像処理を施さないようにカラーバーコードは表示される。なお、色帯の数が4本以上になるとOV9655では全体を収めきれなかったため、最大3本までとする。したがって、実験では、一度に、最大39種類のカラーバーコードを使用できる。

### 6-2 カラーバーコードの構成

色帯の並びによっては、正しく認識できないカラーバーコードがあるかもしれない。また、色帯の間に隙間を設けると色帯がより分離されやすくなるかもしれないが、カラーバーコードの幅が広くなり、イメージセンサの取得画像内に収まらなくなることもあり得る。本節では、上記の懸念に対して、2本以上の色帯が並ぶ際に色帯間に隙間(背景色の黒)を設けた場合とそうでない場合とでカラーバーコードの認識を行い、使用可能な色帯の組と、隙間が必要かどうかを明らかにする。

表 1 認識可能な色帯の組

隙間	色帯の組		
	正認識数	誤認識数	総数
有	31	8	39
無	16	23	39

実験では、1024×64 の取得画像に対し、ヒストグラムのフィルタには 64 個のデータに対するメディアンフィルタを使用した。なお、色帯領域における色帯の幅は、実験的に、R で 400 画素、G で 420 画素、B で 400 画素とし、隙間は 1 画素とした。

実験結果を表 1 に示す。実験を通して、隙間を入れた場合でも色帯の数が 3 本以下ならば、CMOS イメージセンサの取得画像内に収まることが確認できた。さらに、隙間を設けると、より多くのカラーバーコードが使えるので、以降は、色帯間に隙間を入れることにする。

隙間ありでも認識できなかった色帯の組は BGB, BRG, BRG, GBG, GRB, GRG, RBR, および、RGR であった。これは、真ん中の色帯や、右端の色帯がその他の色帯に埋もれてしまったことによる。

結果として、本実験環境では 31 種類のカラーバーコードを使用できることが分かった。カラーバーコードで表現できるボタンの最大数は、画面階層を  $n$  とすると、 $31^n$  である。例えば、画面階層が 3 の場合、最大ボタン数は 29791 個である。今後、試作システムを開発するにあたっては十分な数と考える。

### 6-3 認識率

ユーザの指がタッチパネルに表示されたカラーバーコードに対して傾いている場合、画像の高さによっては、その傾きがカラーバーコードの認識処理に悪影響を及ぼし得る。より具体的には、取得画像の両端にある無効領域が、傾いたカラーバーコードによってつぶされてしまい、カラーバーコードの存在が検出されなくなる。傾きを考慮して補正しながら認識すれば、認識精度は上がると思われるが、その処理に時間を費やしてしまう。また、ヒストグラムのフィルタリングに使用するフィルタの種類や回数によっても精度や処理時間が異なる。

指の傾きが与える画像への影響は、画像の高さを小さくすることによって回避可能である。ただし、少ない行だけの特性でカラーバーコードを識別しようとするため、多くの行を用いた平均的な特性を利用する場合と比較してインパルス的な要素に弱くなり、認識精度が低くなると予測される。フィルタの種類や回数などを変更して対処し得るが、実行時間との兼ね合いになる。

そこで、本節では、画像の高さとフィルタの種類、および、フィルタの回数を変更して、まずは、認識率を評価する。画像の高さが 64 未満の場合、画像を上から下に向かってスキャンし、それぞれのスキャン画像に対して処理を実行した。これは、同じ取得画像でも、スキャンごとに異なった画像に対する評価を実施したことになり、入力したテストパターン画像を増やすためである。テスト画像の数は、64/画像の高さ、であり、使用可能な 31 個のカラーバーコードに対して実施した。なお、フィルタはメディアンフィルタと平滑化フィルタの 2 種類である。

認識率の評価結果を表 2 に示す。結果から、画像の高さが大きいほど、認識率が高くなる傾向が読み取れる。これは、予測した通り、多数の行を使った平均的な特性を利用した方が、インパルス的な要素が取り除かれるためと考えられる。

実験を通して、指の傾きが無理のない範囲内であれば、画像の高さが 8 以下のときに傾きの影響なく認識

表 2 認識率

Image height	Filter	None	Average Filter [degree]				Median Filter [degree]			
			8	16	32	64	8	16	32	64
	1	15.1	48.6	66.9	78.0	74.6	48.2	71.7	92.6	99.4
2	30.9	87.1	96.6	92.6	85.1	73.2	90.6	99.2	100	
4	33.3	80.2	92.6	99.5	96.4	74.0	88.4	98.8	100	
8	57.8	92.4	97.9	100	100	88.5	96.2	99.7	100	
16	86.2	98.9	99.5	100	100	98.4	99.1	100	100	
32	94.7	99.6	100	100	100	99.6	100	100	100	
64	100	100	100	100	100	100	100	100	100	

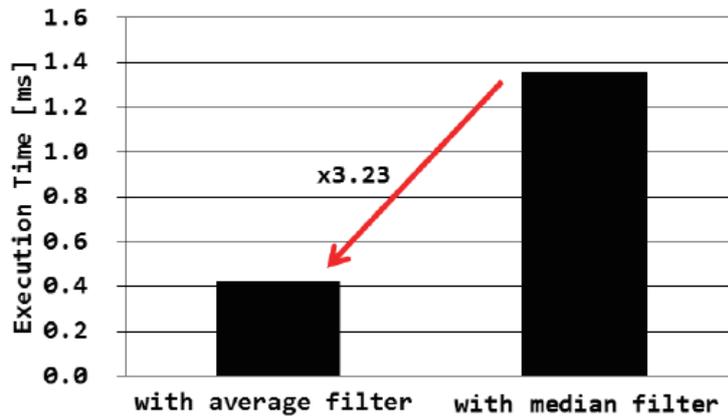


図 6 フィルタの性能

できることも実験的に確認した。つまり、バーコードの傾きを前もって計算して補正しなくても、画像の高さを 8 以下にすれば問題ない。よって、本実験環境において、認識率を 100%にするには、平滑化フィルタの場合、画像の高さは 8 で、次数は 32 以上であれば良い。また、メディアンフィルタの場合、画像の高さは 2 以上で、次数は 64 であれば良い。

#### 6-4 フィルタの性能評価

前節で、精度の面において、平滑化フィルタでは画像の高さが 8 で次数が 32、メディアンフィルタでは画像の高さは 2 で次数が 64 と決めた。それらの処理時間を比較し、どちらのフィルタが適しているか検討する。

図 6 にその結果を示す。プログラムの実行時間を計測するために、`rdtsc` 命令によって取得した性能カウンタの値を用いた。その際、スケジューリングの優先度を最大にしてプログラムを実行させ、それを 100 回繰り返し、得られた 100 個の実行時間から最小のものを選んだ。

図 6 からわかるように、ソーティングを伴うメディアンフィルタの方が平滑化フィルタよりも処理時間が長いことがわかる。その差は約 3.23 倍であった。以上から、本実験環境では、画像の高さは 8、フィルタリングには次数が 32 の平滑化フィルタが適している。

#### 6-5 ハードウェアの性能評価

前節までのソフトウェアによる認識率や性能評価結果から、ハードウェアにおいても、画像の高さは 8 で、次数が 32 の平滑化フィルタを用いる。使用可能な 31 個のカラーバーコードに関して、認識率は、表 2 のソフトウェアによる認識率と同じ結果であった。

提案ハードウェアの性能に関する比較対象として前節までの PC と組み込み機器で使用されるマイコンを使用した。マイコンは、STMicroelectronics 社の STM32F4 であり、プロセッサが ARM 社の Cortex M4 で、動作周波数は 168MHz である。本マイコンは 1MB の ROM と 192KB の RAM を持つ。提案ハードウェアは、メンターグラフィックス社の論理シミュレータ (ModelSim10.1c) 上で動作周波数を 100MHz として実行時間を計測した。また、PC とマイコンは、予め画像がメモリにある状態でプログラムを 10 回実行させ、それらの実行時間のうち最小値を用いる。

図 7 に性能評価の結果を示す。結果から、提案ハードウェア (HW) は、PC とマイコン (MCU) と比較し、大幅な性能向上を達成できた。具体的に、対 PC の性能向上は 2.8 倍～3.6 倍 (平均で 3.3 倍) であり、対マイコンのそれは 109 倍～122 倍 (平均で 116 倍) であった。PC とマイコンに搭載されたプロセッサもパイプライン化されているが、一つのデータを処理するために多数の命令を要する。それに対して、提案ハードウェアでは一つのデータを 1 クロックで処理できるためである。

また、PC、マイコン、および、提案ハードウェアの動作周波数は、それぞれ、3.2GHz、168MHz、および、100MHz である。一般的に CMOS テクノロジーの半導体集積回路 (ロジック) の平均的な動的消費電力 ( $P_{avg}$ ) は、 $P_{avg} = \alpha_T \cdot C_{load} \cdot V_{DD} \cdot f_{CLK}$  とモデル化される [11]。ここで、 $\alpha_T$  はトランジスタのトグル率、 $C_{load}$  はトランジスタ出力の総負荷容量、 $V_{DD}$  は電源電圧、 $f_{CLK}$  は動作周波数である。つまり、動的消費電力は動作周波数に比例する。よって、提案ハードウェアは、動作周波数が低いにも関わらず最短時間で処理を終えており、性能に加えて消費電力にも優れるといえる。

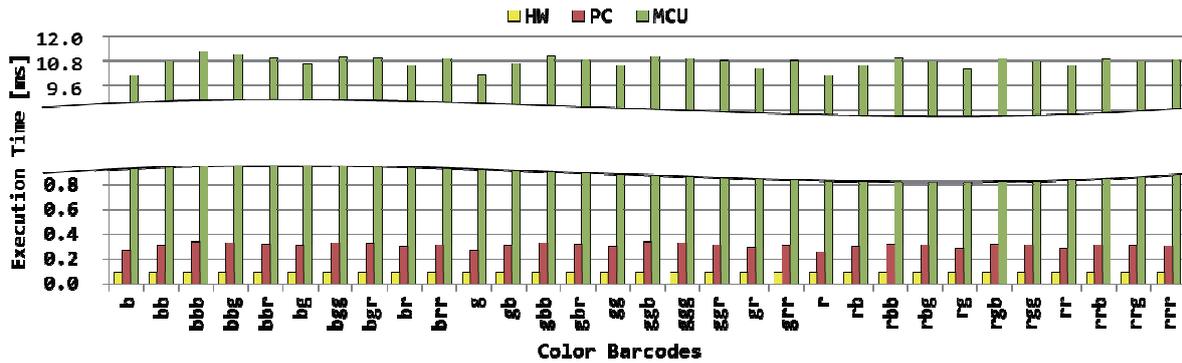


図 7 ハードウェアの性能

### 6-6 ハードウェアの回路規模

提案ハードウェアの回路規模や動作速度に対する一指標として、いくつかの FPGA (Xilinx 社) に実装した結果を表 3 に示す。FPGA への実装ツールに関して、spartan6 には Xilinx 社の ISE14.4 を、その他の FPGA には Xilinx 社の Vivado 2012.4 を使用した。

表 3 の FPGA は FPGA の大分類 (ファミリー) を意味し、CLK は入力可能な最大の動作周波数である。また、FFs, LUTs, BRAMs, DSPs は、FPGA 内で使用された回路資源の数であり、それぞれ、フリップフロップ、真理値表、組み込みメモリ、および、乗算器に対応する。回路資源のカッコ内のパーセンテージは、spartan6 を除いて、各ファミリー内で規模が最小の FPGA における使用率である。廉価版 FPGA である spartan6 に関しては、最小規模の FPGA には実装できなかったため、一つ規模の大きな FPGA を対象とした。ちなみに、Artix7 が今後の廉価版 FPGA である。

結果から、対象とした全 FPGA に実装可能なことがわかる。対象 FPGA が、spartan6 は 2 番目に規模が小さく、それ以外は最小規模の FPGA なので、提案ハードウェアは、表 3 で示したファミリーに関して、大部分の FPGA に実装可能である。

提案ハードウェアを含めても FPGA の容量に余裕があるため、FPGA 上にプロセッサ組み込んだシステム LSI としての実装も可能である。つまり、単一 FPGA で、提案ハードウェアとソフトウェアを組み合わせた組み込みシステムを実現し得る。また、前節の論理シミュレーションで設定した 100MHz 以上の動作周波数で動作可能なこともわかる。

## 7 むすび

指先カメラを用いた視覚障がい者向けタッチパネル操作支援システムでは、タッチパネルに近接したイメージセンサがとらえたぼやけた画像からカラーバーコードを認識する必要がある。本研究では、カラーバーコードを認識する処理フローを開発し、その認識処理の更なる高速化や省電力化を目的として、処理のハードウェア化を実施した。

開発した処理フローは、HSV 色空間上で RGB を分離した画像に対し、無色と各色のヒストグラムを作成する。そして、それぞれのヒストグラムに対するフィルタリングといくつかの閾値によってカラーバーコードの色帯を検出する簡素なフローである。演算はすべて固定小数点化されており、内部はストリーム処理なので、ローエンドなマイコンだけでなく、ハードウェア化にも適している。処理フローをハードウェア化するためにあたって、認識処理に最適な形でパイプライン化されるように設計した。

実験を通して、カラーバーコードの構成法や使用可能な色帯の組を明らかにした。また、フィルタリングで使用するフィルタの種類によって認識率に違いがあり、指の傾きに影響を受けずに認識率を上げるには、

表 3 FPGA への実装実験の結果

FPGA	CLK [MHz]	FFs	LUTs	BRAMs	DSPs
Spartan6	184	3974 (35%)	2401 (42%)	2 (6%)	3 (19%)
Artix7	189	3617 (3%)	1893 (3%)	1 (1%)	3 (1%)
Kintex7	215	3617 (4%)	1894 (5%)	1 (1%)	3 (1%)
Virtex7	240	3617 (1%)	1883 (1%)	1 (0.1%)	3 (0.1%)

最適な画像の高さとフィルタの次数があることが分かった。

提案ハードウェアは、最適なパイプライン設計によって、パーソナルコンピュータに対して約3倍、マイコンに対して約116倍の性能向上を達成した。提案ハードウェアの動作周波数は、パーソナルコンピュータの約3%、マイコンの約70%であり、性能を向上させながら、省電力化も達成しうる。また、提案ハードウェアの回路規模は大きくなく、多くのFPGAやASICの一部に組み込んで使用できる。

今後、図1と等価な試作システムを開発し、実際に動かしながらの認識率の評価や、音声案内との連動を実現する。そして、提案システムの使いやすさを評価するために、操作時間の調査や使用感に関するアンケートなどの被験者実験を行う。

## 【参考文献】

- [1] 富士経済, "タッチパネルの世界市場を調査," 富士経済グループプレスリリース第12031号, 2012.
- [2] T. Yoshida, K.M. Kitani, H. Koike, S. Belongie, and K. Schlei, "Edgesonic: Image feature sonication for the visually impaired," Proc. of the 2nd Augmented Human International Conference, pp.11:1-11:4, 2011.
- [3] J. Su, A. Rosenzweig, A. Goel, E. deLara, and K.N. Truong, "Timbremap: Enabling the visually-impaired to use maps on touch-enabled devices," Proc. of the 12th International Conference on Human Computer Interaction with Mobile Devices and Services, pp.17-26, 2010.
- [4] G. Yfantidis and G. Evreinov, "Adaptive blind interaction technique for touchscreens," Universal Access in the Information Society, vol. 4, no. 4, pp. 328-337, 2006.
- [5] S.K. Kane, J.P. Bigham, and J.O. Wobbrock, "Slide rule: Making mobile touch screens accessible to blind people using multi-touch interaction technique," Proc. of the ACM SIGACCESS Conference on Computers and Accessibility, pp. 73-78, 2008.
- [6] 半田拓也, 坂井忠裕, 御園政光, 森田寿哉, 伊藤崇之, "触覚ディスプレイにおける点字の提示手法," 電子情報通信学会技術研究報告, WIT, 福祉情報工学 107(555), pp. 23-26, 2008.
- [7] O. Bau, I. Poupyrev, A. Israr, and C. Harrison, "Teslatouch: Electro vibration for touch surfaces," Proc. of the 23rd ACM Symp. on User Interface Software and Technology, pp. 283-292, 2010.
- [8] 久野素有, 山下淳, 金子透, "ステレオカメラを用いたタッチパネル操作支援システムの構築," 電気学会論文誌 D, vol.131, no.4, pp.458-465, 2011.
- [9] A. Yamawaki and S. Serikawa, "A wearable supporting system for visually impaired people in operating capacitive touchscreen," Applied Mechanics and Materials, vol. 103, pp. 687-694, 2011.
- [10] 三谷芳弘, 浜本義彦, "画像処理による抵抗器の読み取りに関する研究," 日本高専学会誌, vol. 15, no. 2, pp. 65-67, 2010.
- [11] Sung-Mo Kand and Yusuf Leblebici, "CMOS Digital Integrated Circuits Second edition", chapter 11, pp. 452-456, McGRAM-Hill, 1998.

## 〈発表資料〉

題名	掲載誌・学会名等	発表年月
指先カメラを用いた視覚障害者のためのタッチパネル操作支援システムにおけるカラーバーコード認識処理の検討	HCG シンポジウム 2012 論文集	2012 年 12 月
A Color Barcode Detecting Method in a Supporting System for Visually Impaired Persons Operating Capacitive Touchscreen	Proceedings of the International MultiConference of Engineers and Computer Scientists 2013	2013 年 3 月
指先カメラを用いた視覚障がい者向けタッチパネル操作支援システムにおける画像処理のハードウェア化	産業応用工学会論文誌 1 巻 1 号	2013 年 3 月