

Web カメラを用いた 片手文字・座標入力システム

瀧口智史[†] 田中敏光[†] 佐川雄二[†]

Web カメラとクレジットカード大の矩形物体を使って、モバイル環境で、片手で文字入力とポインタ操作を行うシステムを作成する。文字はカード上の親指の位置と動きで入力するが、1文字を50音表の行と段の2ステップで入力することで指が動く範囲を限定する。ポインタは腕の動きで操作する。ポインタを視野内のカードの位置と同期させることで、ジェスチャで直感的に移動できる。この2つのモードは、腕の伸縮により、瞬時に切り替えることができる。

System to input characters and positions by one-hand action captured with web camera

Tomofumi Takiguchi[†] Toshimitsu Tanaka[†]
and Yuji Sagawa[†]

In this paper, we propose the one-hand character-input and pointing system using a web camera for mobile computing. A simple-color rectangular object, size of which is same as the credit card, is used as a marker. Characters are input by motion of the thumb on the rectangular object. One Japanese syllabary characters is input by two actions, selecting a row of the Japanese syllabary and then selecting a column. The pointer can be operated by the gesture, because position of the pointer is synchronizing to position of the rectangular object. The pointer-operation and character-input is quickly switched by stretching/bending user's arm.

1. はじめに

近年、PCの小型化が急速に進んでおり、Yシャツの胸ポケットに入るものも登場している。また、小型軽量のウェアラブルモニターや、網膜に直接光を当てて映像を映すディスプレイも開発されている。これらデバイスの進歩に伴い、再びウェアラブルコンピューティングの可能性が高まっている。

ウェアラブルコンピューティングでは、文字入力やポインティングに利用するデバイスも小型で携帯性に優れている必要がある。しかし、QWERTYキーボードの小型化には限界がある。仮想キーボードを使うとポインティングデバイスで文字を入力できるが、画面を大きく占有してしまう。さらに、多くのソフトウェアはポインティング操作を要求するが、従来の方法では、文字入力とポインティングでデバイスの持ち替えやモードの切り替えが度々発生し、作業が中断してしまう。

そこで本研究では、小型のWebカメラを装着し、クレジットカード大の矩形物体をガイドとして利用することで、片手による文字入力とポインティング操作を瞬時に切り替えられるシステムを作成する。

2. 従来研究

これまでに、ウェアラブル環境での文字・座標入力手法は多数提案されている。専用の入力デバイスを使用する方法の一例として、HandKey社のTwiddler[1]を挙げる。これは片手で操作することができ、背面の12個のキーと前面のALT/CTRL/NUM/SHIFTキーを組み合わせて押すことで、文字を入力できる。ただし、2~3個のキーを同時に押す必要があるため、指の動きが複雑になる。また、キーの組み合わせの学習にも時間がかかる。前面にスティックが付いているので、ポインティングも行えるが、細かな操作は難しい。

Keyboard+IE[7]などの携帯電話式キーボードを使う方法もある。携帯電話に慣れていればキー操作を覚えなくてもよい利点はあるが、1文字入力するために多数の打鍵が必要とする。また、携帯電話配置の仮想キーボードソフトもある。この場合、キーが大きくなることで誤った打鍵が減る利点はあるものの、クリック感に乏しいため、ハードキーボードのような操作性は期待できない。

参考文献[2]では、腕に装着した小型カメラに向かって空中で文字を書く方式を提案している。同様に、参考文献[3]では、頭部に装着したカメラで手に持った矩形物体を検出し、その内部で指先の位置を追跡することで、文字や絵を入力する方法を提案している。いずれの方法も、手書き文字認識技術を用いて文字を入力する。これらの手法には、操作の習得が簡単で、文字以外に図形も入力できる利点がある。しかし、一

[†] 名城大学大学院
Meijo University

文字ずつ手書きする必要があるため入力速度は遅いし、手書き文字認識の精度も高くない。また、両手を使用するので、ウェアラブル環境に適しているとは言えない。

参考文献[4]では、2個の加速度センサを手先や肘に装着し、ジェスチャ操作によってポインティングを行っている。ジェスチャを使うと、ポインティングやスクロールなどを直感的に行うことができる。しかし、文字入力に利用するには、文字一つ一つに対してジェスチャを定義する必要があり、文字とジェスチャの間に手書き文字入力のような関連性がないため、ジェスチャの学習に時間がかかる。

ジェスチャによる文字入力の中には、Palm OS で使われた Graffiti のようにある程度普及したものもある。しかし、ジェスチャの学習には手間と時間がかかるため、他の手法はほとんど実用になっていない。同様の理由やコストの点から、特殊配置のキーボードもほとんど使われていない。

3. 提案システム

3.1 概要

本システムの利用者は、小型の Web カメラを衣服などに着け、単色のクレジットカード大の矩形物体をカメラにかざす(図1)。カードがカメラに近いときは文字入力モード、遠いときはタブレットモードとして動作するので、利用者は腕の伸縮によってカードをカメラに近づける/遠ざけることで、二つのモードを自在に切り替えることができる。この際にカードを持ち替える必要はないので、切り替えは瞬時に行える。



(a) 文字入力モード時 (b) タブレットモード時

図1 システム利用時のカメラとカードの位置関係

3.2 文字入力モード

カードがカメラに大きく映され、上を向いていれば文字入力モードとなる。システ

ムがカードを検出し、位置と回転を補正するので、利用者はカードをフレーム内に納めるだけでよい。

文字はカード上の親指の動きで入力する。カード中央付近に、縦一列に並んだ6つの領域が設定されているので(図2左)、ここで親指を上下左右に動かして文字を入力する。システムはガイド画面(図2右)をモニタに提示する。ここに、カメラ画像から検出した指先の位置、選択中の領域、カメラ視野内のカードの位置が示されるので、利用者は直接カードを見ることなく、文字の選択とカード位置の調整ができる。

図3にひらがな入力の例を示す。学習の負担を減らすために、50音表に準拠した入力方法を採用している。初期状態では、6つの領域に50音表の各行が2行ずつペアにして配置されている。ここで指を上下に動かして、(a)のように、1つの領域を選ぶ。この状態で、指を左に動かせば左側に表示された文字の行に(b)、右に動かせば右側の行に絞られる。その後、段の選択に入るので、同様に、指を上下して領域を選んでから(c)、指を右に動かすことで、表示された文字が入力される(d)。段選択時の一番下の領域は、濁音や清音などへの切り替えスイッチとなっており、選択する度に順番に切り替わる。入力後に指を中央に戻すと初期状態になるので、連続して文字を入力できる。また、段選択の状態でも指を左に動かしても初期状態に戻ることができる。文字入力が完了した直後に、指を上下いずれかに大きく動かすと、(e)に示す特殊操作に入り、同様の操作で文字の変換が行える。特殊操作の状態でも一番下の領域を選択し、指を左に動かすと初期状態になる。

このように、指の上下の動きで領域を選択し、左右の動きで確定するので、狭い範囲の指の動きで文字を入力できる。また、6つの領域の内外を判定するだけなので、指先の検出誤差にも強くなる。英数字についても同様の入力手順が用意されており、カードを左に傾けるジェスチャで交互に切り替えることができる。

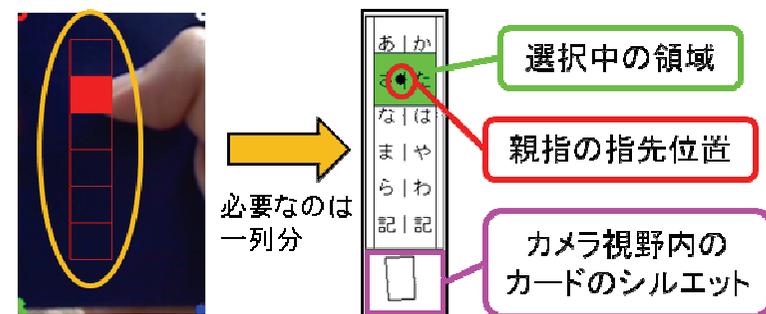
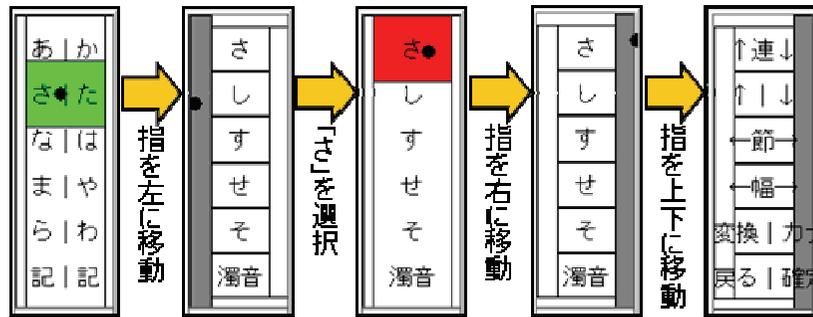


図2 ガイド画面の概要



(a) 初期状態 (b) 行選択後 (c) 段選択 (d) 入力完了 (e) 特殊操作
図 3 文字入力の流れ

3.3 タブレットモード

カードがカメラに小さく映されていると、タブレットモードとなる。この状態では、カメラの視野がディスプレイに対応付けられ、カメラ画像から検出したカードの中央の位置にカーソルが移動する。図 4(a)では、検出したカードの中央とカーソルの位置をオレンジの円で表示している。このように、対応した位置にカーソルが移動する。

クリックはカードを傾けるジェスチャで入力する。カードを左に傾けると左クリック、右に傾けると右クリックとなる。図 4(b)は右クリックの例で、図 4(a)のオレンジの円の位置を左下隅として、メニューが表示されている。

腕の動きがカーソルの動きに、手首を左右に振る動きがクリック動作にそれぞれ同期するので、利用者は直感的に操作することができる。



(a) カーソルの移動 (b) 右クリック
図 4 タブレットモード時のカーソル同期と右クリック処理

4. 実装

4.1 処理の流れ

初めにカメラ画像からカードの 4 頂点、傾き、および大きさを検出する。カードが上を向いている場合は、カードの大きさを参照し、閾値以上ならば文字入力モード、閾値未満であればタブレットモードとして動作させる。文字入力モードでは、指先を検出し、3.2 節で述べた入力領域との比較を行う。タブレットモードでは、カードの中央座標を検出し、モニタ座標系に変換する。カードが傾いている場合には、直前のモードと傾きの方向に応じた処理を行う。

4.2 カードの検出

カードは片手で保持するので、肌色領域に囲まれている。そこで、カメラ画像から肌色領域を抽出し、その近傍でカードを探索する。肌色抽出には、YCrCb 空間の CrCb 値と修正 HSV 空間[5]の HS 値を併用する。

次に、手で保持されている場所を探索する。抽出した肌色領域の境界において、Cr 値の勾配方向 θ を式(1)から算出する。式(1)において、 (x, y) は画素の座標値、 F_x は水平方向の Cr 勾配、 F_y は垂直方向の Cr 勾配をそれぞれ示している。また、勾配方向は 45 度刻みの 8 方向に圧縮する。

$$\begin{aligned} F_x &= C_r(x+1, y) - C_r(x-1, y) \\ F_y &= C_r(x, y+1) - C_r(x, y-1) \\ \theta &= \arctan(F_y / F_x) \end{aligned} \quad (1)$$

すべての肌色境界画素における勾配方向を計算したら、勾配方向で画素をペアにする。各画素において勾配方向に走査を行い、最初に到達した別の肌色境界画素の勾配方向を参照する。そして、それが自身の勾配方向の逆向き、すなわち 180 度違いであれば、それらの画素をペア画素として登録する。

その後、ペア画素間の色相ヒストグラムを計算し、ヒストグラムが最大となった色相値があらかじめ登録してあるカードの色相値の ± 15 の範囲内であれば、カードを保持している肌色領域の一部として保存する。これらの処理をすべての肌色境界画素で行うことで、カードのおおまかな位置を検出する。図 5 にペア画素の作成例を示す。赤が (0 度, 180 度)、緑が (45 度, 225 度)、青が (90 度, 270 度)、白が (135 度, 315 度) のペアをそれぞれ示している。

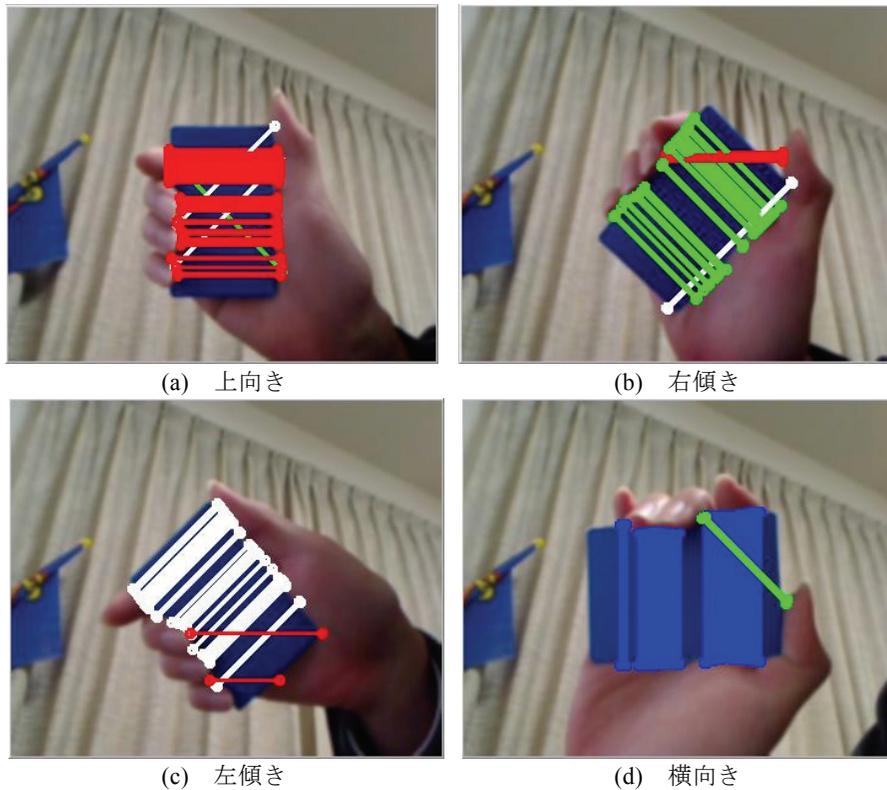


図 5 カードの傾き別のペア画素作成例

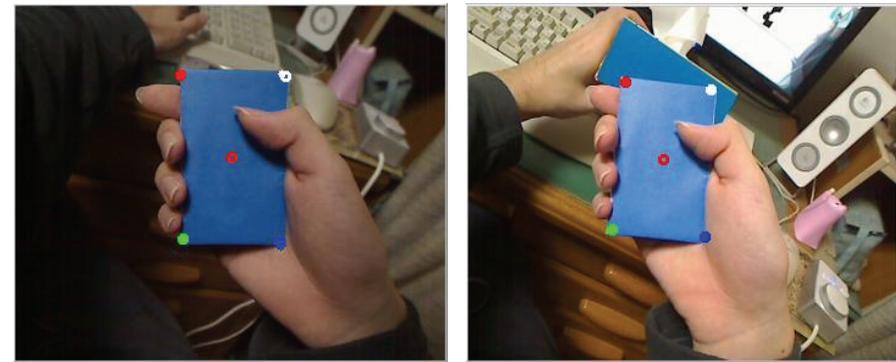
図 5 より、最も多くのペアが作成された方向が、カードのおおまかな傾きとなる。そこで、(a)のようにカードが上向きの場合には、カードの4頂点の探索を行う。ただし、実際には傾いていなくても、ペア画素の判定誤りにより、(b)または(c)のように傾いたペアが多数を占めることがある。そこで、(b)や(c)の場合でも、直前のフレームのカードの傾きが閾値未満の場合には、上向きと仮定してカードの頂点を探索する。

カードの頂点の探索では、まず、ペア画素ごとに水平方向の位置の差を求め、その最大値をカードの仮の横幅とする。カードの縦幅は、クレジットカードの縦横比がおおよそ1:1.6であることから、横幅の1.6倍とする。次に、縦方向(90度と270度のペア)のペア画素が複数作成されている場合は、それらの画素の中で最も下のものをカード頂点探索範囲の下端とし、それ以外の場合はペア画素群を包含する矩形領域の中

心座標を探索範囲の中心とする。

次に、探索範囲内で、色相値が事前に登録したカードの色相値の±15の範囲にある画素を選択する。その領域の輪郭点を抽出し、カード探索範囲の4頂点に最も近い輪郭点をそれぞれカードの頂点とする。ただし、背景にカードと同色相値の物体が存在する場合は、この方法では頂点を正しく検出できない。そこで、カード領域を抽出する際に輝度エッジで領域を分割する。そして、領域ごとに画素数を算出し、最も画素数が多いものから順に頂点を検出する。また、各頂点間の角度を検出し、指で頂点が隠れたと判断された場合には、頂点位置を補正する。

輪郭点は、ノイズの影響を大きく受けるので、現在フレームを含めて直前の8フレームの頂点座標の平均値を使用する。ただし、カードが動く場合には、頂点座標の変化に遅延が発生するため、頂点ごとに移動量の分散を計算し、2点以上で分散が閾値を超えた場合には、現在のフレームの頂点座標値を用いる。図6にカード4頂点の検出結果を示す。



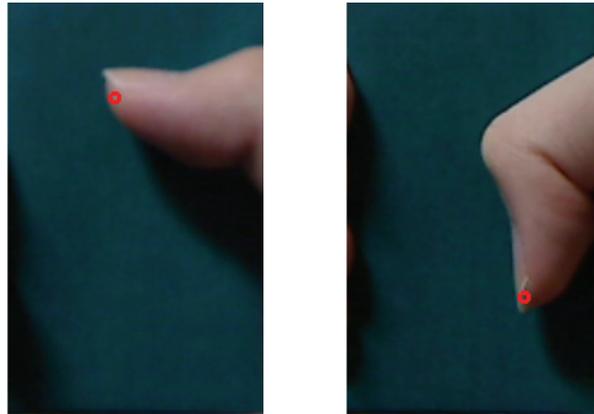
(a) 通常時 (b) 背景に同色相値の物体が存在

図 6 カード4頂点の検出例

4.3 文字入力モード時の処理(親指の指先検出)

3.2節で述べたように、文字はカード上の親指の動きで入力する。そこで、4.2節で検出したカードの4頂点を射影変換し、カード座標系を取得する。次に、カード座標系で一定以上の肌色領域の輪郭点を抽出し、指先の候補とする。親指はカードの領域外から領域内へ進入しているため、カードの辺を分断している。また、指先は端点である。そこで、分断している辺から最も水平方向の距離が大きい輪郭点を基点Hとし、基点Hから一定距離内の輪郭点を求め、それらの重心を指先の座標とする。基点Hを指先としないのは、指先の位置を安定して求めるためである。ただし、図7(b)のように指が曲がると、指先が正しく検出できなくなる。そこで、抽出した輪郭点のうち最

も下の輪郭点を基点 V とし，基点 H と基点 V の垂直距離が水平距離の 2 倍以上であれば，基点 V を採用する．そして，基点 V から一定距離内にある輪郭点の重心を指先の座標とする．図 7 の赤丸は検出された指先の位置を示している．



(a) 屈曲なし (b) 屈曲あり
 図 7 指先検出の例

検出した指先座標と入力領域との進入判定を行いながら，段階的に文字入力を行う．また，ノイズの影響を減らすために，指の動きが小さい場合には Lucas-Kanade 法を用いて指先を追跡する．

4.4 タブレットモード時の処理

カードの 4 頂点から中央座標を検出し，カーソルの位置とする．ただし，4.2 節の手法では，カードを保持する手がカメラ視野内に収まっていないと，4 頂点を正しく検出できない．そこで，図 8 中央に示すように，カメラ視野の上下左右を一定範囲排除した部分をタブレット座標系として定義する．そして，カメラ座標系→タブレット座標系→モニタ座標系と座標変換する．



図 8 座標系変換の流れ

クリックは，カードを左右に傾けるジェスチャで行う．カードの傾きは図 5 に示されているペア画素の数とカードの頂点間の角度から判定する．クリックしようとカードを傾けることでカーソル位置がずれてしまうことを防ぐため，カードが静止したらその位置を記憶する．そして，クリックを検出したら，カーソルを記憶した位置に戻してからクリック処理を行う．

5. 入力速度の比較

3 章で提案したシステムを実装し，文字入力のテストを行った．比較する文字入力方式は一般的な携帯電話で採用されているマルチタップ方式で，比較対象は入力に要する時間とした．表 1 に実験に用いた文字列，およびそれぞれの手法の動作環境と入力時間を示す．

マルチタップ方式は，被験者が普段使用している携帯電話のソフトウェアキーボードを使用した．また，提案システムの入力方法は学習済みである．

表 1 文字入力テストの実行環境と結果

	提案システム	マルチタップ方式
入力文	名城大学大学院理工学研究科情報工学専攻 Computer Graphics Laboratory Web カメラを使った片手文字・座標入力システム	
入力した文字	めいじょうだいがくいだいがくいんりこうがくけんきゅうかじょうほうこうがくせんこう (39 文字) Computer Graphics Laboratory (空白含めて 28 文字) Web カメラをつかったかたてもじ・ざひょうにゅうりよくしすてむ (31 文字) の合計 98 文字．これに変換操作と 2 回の改行が加わる．	
CPU	Intel Core i7 U640M 1.20GHz	Qualcomm Snapdragon 1.0GHz
IME	Google 日本語入力	POBox Touch 3.0
OS	Microsoft Windows 7 Professional 32bit	Google Android 1.6
入力に要した時間	204 秒 (2.1 秒/文字)	156 秒 (1.6 秒/文字)

ウェアラブル環境下における文字入力の主な用途は，短文のメールや検索語の入力と考えられる．2008 年度の調査では，携帯電話のメール文字数は，50 文字未満である割合が全体のおよそ 55.3%，100 文字未満だとおよそ 75%となっている[6]．このため，今回のテストでは 70 文字の入力文（平仮名に換算すると 98 文字）を用いている．

表 1 に示すとおり、提案システムでは 1 文字あたり約 2 秒で入力できている。マルチタップ方式と比べるとおよそ 1.3 倍の時間が必要だが、時間差にすると約 50 秒にすぎない。メール操作にはポインティングも必要なので、モードの切り替え時間を考えると、差は小さくなる。また、検索語やサイト名の文字数はもっと少ないので、Web 検索では、提案システムとマルチタップ方式の入力時間の差は十分に小さくなる。

6. おわりに

本報告では、クレジットカード大の単色カードをガイドとして使用する片手文字・座標入力システムを提案し、実装方法を示した。文字はカード上の親指の動きで入力する。提案手法では、1 文字を 2 ステップで指定することで、指先が縦 6 区画の内部、またはその左右上下、のどこにあるのかを識別するだけの処理により、文字入力を実現している。このため、親指の動きは狭い範囲に限定される。平仮名は 50 音表の行と段で指定するので、操作方法の学習も容易である。

ポインティングでは、タブレットのように、カメラに映ったカードの位置をポインタの座標として入力する。クリックはカードを左右に傾けるジェスチャで行う。手首と腕の動きが直接マウスの動きとなるため、直感的に操作することができる。2 つの入力モードは、腕を伸縮してカードとカメラの距離を変えることで、瞬時に切り替えることができる。

実装したシステムの文字入力時間と、携帯電話で使用されているマルチタップによる文字入力時間を比較した。その結果、提案システムの文字入力時間はマルチタップ方式のおよそ 1.3 倍となった。提案手法の方が低速だが、実際の文字入力用途では大した時間差にならないことや、タッチデバイスを持ち歩く必要がないこと、文字入力とポインティングを瞬時に切り替えることができることなどを考慮すると、提案手法はモバイル環境において有効な文字・座標入力方法といえる。

今後は、より多くの被験者で、文字入力の速度と精度や、システムの操作方法の学習に要する時間などを評価する。また、タブレットモードにおいて、ジェスチャによるウィンドウ操作やスクロールなどの機能を追加する。

参考文献

- 1) Handkey Official Site, <http://www.handykey.com/>
- 2) 久米祐一郎, 島田正文: 腕装着型カメラと反射指標を用いた装用型入力デバイスの検討, 映像情報メディア学会誌, Vol.60, No.2, pp.249-253 (2006).
- 3) 浮田宗伯, 寺部亮紘, 木戸出正継: ウェアラブル仮想タブレット: 赤外線照射カメラを利用した指先入力インタフェース, 情報処理学会論文誌, Vol.45, No.3, pp.977-990 (2004).
- 4) 所洋平, 寺田努, 塚本昌彦: 2 つの加速度センサを用いたポインティング手法, 第 16 回イン

タラクティブシステムとソフトウェアに関するワークショップ予稿集 (2008)

<http://www.wiss.org/WISS2008Proceedings/papers/paper0031.pdf>

5) 松橋聡, 藤本研司, 中村納, 南敏: 顔領域抽出に有効な修正 HSV 表色系の提案, テレビジョン学会誌, Vol.49, No.6, pp.787-797 (1995).

6) japan.internet.com 「一度のメールで入力する文字数、若い年代ほど”短い”傾向に一アイシェア調べ」 <http://japan.internet.com/wmnews/20080806/3.html>

7) 株式会社メヴァエル Keiboard+IE, <http://www.mevael.co.jp/product01.html>