

## WEBカメラを用いた肢体不自由者用入力インタフェースの試作

宮崎英一, 坂井聡\*, 谷口公彦\*\*, 佐野将大\*\*, 野田知智\*\*\*, 近藤創\*\*\*\*

(技術教育)(特別支援教育)\*

(香川県立高松養護学校)\*\* (香川県立聾学校)\*\*\* (香川県立善通寺養護学校)\*\*\*\*

760-8522 高松市幸町1-1 香川大学教育学部

\*\*761-8057 高松市田村町1098 香川県立高松養護学校

\*\*\*761-8074 高松市太田上町513-1 香川県立聾学校

\*\*\*\*765-0001 善通寺市仙遊町2丁目1-2 香川県立善通寺養護学校

## The Trial Production of Physically Handicapped Persons for Input Interface using the WEB Camera

Eiichi MIYAZAKI, Satoshi SAKAI, Shoudai SANO, Tomohiro NODA and Hajime KONDO

*Faculty of Education, Kagawa University, 1-1 Saiwai-cho, Takamatsu 760-8522*

*\*\*Kagawa Prefectural Kagawa Special education school, 1098 Tamura-cho, Takamatsu 761-8057*

*\*\*\*Kagawa Prefectural School for the deaf, 513 Ootakami-mati, Takamatsu 761-8084*

*\*\*\*\*Zentuzi Prefectural Kagawa Special education school, 2-1-2, Senyou-cho, Zentuzi 761-8057*

**要旨** 本研究ではWEBカメラを用いたトラッキング入力インタフェースを試作した。これはユーザの動作可能な任意の部分をリアルタイムでトラッキングターゲットとして選択し、この部分の動きだけを検出する事で、この動きに応じてコンピュータのマウスやキーボードといったインタフェースと同じ操作を行う。さらにWEBカメラで撮影された動画に画像処理をおこなうので、カメラと撮影対象部位の距離を調節することで、大きな動作から小さな動作に対応できるので、ユーザの可動域の大小や動作方向等の制限を受けない。このため本システムだけで、多くのユーザの利用が可能になると考えられる。

**キーワード** WEBカメラ, 障がい者支援, 入力インタフェース, マイクロコントローラ

## 1. はじめに

現在、インターネット環境やスマートフォン等の普及に伴い、これらの機器は日常生活において当たり前で使用されているIT機器として若年層から高齢者まで広く普及してきている。我々の日常生活におけるインターネット等を介した情報への依存度は日々拡大し続けており、ネットショッピングやコミュニケーションツールとして日常生活において必要不可欠なものとなっている。このように、我々の日常生活に様々な質的向上をもたらすことを可能にしたIT環境であるが、これらの環境が障害のある人にとっては十分に活用できる環境は整っていない。

その大きな原因の1つとして、入力インタフェースの問題が存在する。これは、一般的に使用されているマウスやキーボードといった入力インタフェースが、障害のある人の使用環境を前提に設計されていないことに起因している。特にマウスの位置決めや、クリック・ダブルクリックといったマウス操作が困難な場合、これらのインタフェースでは使用出来ない場合がある。

そこで本研究ではWEBカメラを用いたトラッキング入力インタフェースを試作した。これはユーザの動作可能な任意の部分を実時間でトラッキングターゲットとして選択し、この部分の動きだけを検出する事でマウスやキーボードといった入力インタフェースと同じ操作を行う。さらに、カメラと撮影対象部位の距離を調節することで、ユーザの大きな動作から小さな動作まで対応できるので、ユーザの可動域の大小や動作方向等の制限を受けない。このため本システムだけで、多くのユーザの利用が可能になると考えられる。

## 2. インタフェースについて

### 2.1 タッチパネル

最近、スマートフォンやタブレットPC等のタッチパネルを有する機器が多くの場合で使用されるようになってきているが、運動動作に制限がある場合、これがユーザインタフェースとして問題になる場合がある。元来、タッチパネルは直感的なインタフェースを提供し、子供やお年寄りといったコンピュータの専門的な知識が無い人でも、直感的に使用できるという特徴がある。しかし従来の入力インタフェースが備えていた、キーボードのキー操作やマウスのクリックといったユーザの操作に伴う機械的な打鍵感がないので、操作に対する感覚的なフィードバックが無い。このため、キーが押されたかどうかの判断が、困難な場合があり、これが操作感という点から入力インタフェースとして問題になる場合がある。

また、現状の多くのOSの持つインタフェース画面は、タッチパネルの操作においても、基本的にはマウスやキーボードで操作されていたOSをベースとしており、細かな位置の制御やダブルタップ、スワイプ等タッチパネル特有の操作方法が困難な場合がある。

### 2.2 スイッチインタフェース

上記のことから、コンピュータの操作等において運動機能に制限がある場合、マウスやキーボードの代わりにスイッチを代用する場合<sup>1</sup>が多い。これらは機械的な入力装置であり、押しボタンスイッチや引きスイッチ、呼吸スイッチ等様々なスイッチが入力インタフェースとして利用されている。これら多種類スイッチが利用されるのは、運動機能の制限が個人の特性によって大きく異なるためで

ある。例えば、脳性マヒの方と筋ジストロフィーの方を比較した場合、動作の正確性と稼働範囲が相反の関係にあり、これらの条件を満足するスイッチが要求される。

動作に制限がある場合に使用されている機械的スイッチ例<sup>2</sup>を図1に示す。同図は押しボタンスイッチを示しており、このスイッチと同図左の専用インタフェース機器を組み合わせる事で、ワンボタンのスイッチでコンピュータの様々な操作が可能となる。この押しボタンスイッチも大きさが一定でなく、ユーザの使用特性に合わせて足での操作を行う大きなものから手の中に納まる小型のものまで、様々な大きさのスイッチがある。このように動作に制限がある場合、その人の特性に適応したインタフェースを準備する必要があるが、実際にはユーザの特性に合わせた多くのインタフェースを準備する事は困難である。このため、動作に制限がある場合には運動制限に適応できる入力インタフェースそのものが少ないという問題点があった。



図1 押しボタンスイッチ

### 2.3 WEBカメラインタフェース

上記の問題点を解決するため、本研究では、WEBカメラを用いたユーザインタフェースを提案する。図2にWEBカメラインタフェースのトラッキング対象画面を示している。これは同じシステムを用いてWEBカメラの位置を変えて、トラッキング部分を



図2 a) 可動域が大きい場合

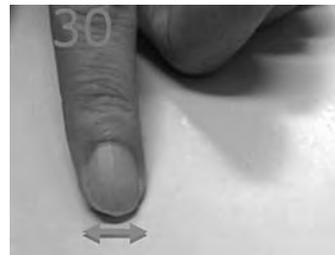


図2 b) 可動域が小さい場合

ズームアウトした場合を同図a)、またズームインした場合を同図b)に示している。同図の矢印(⇔)は同じ画素数を示しているが、実際の指の動作距離は同図a)の方が大きい。このように可動域の大小に関してはWEBカメラのズーミングによって対処できる事がわかる。

しかしここで問題になったのが、スイッチのオンオフを決定するトリガーの位置である。この問題を図3に示す。このスイッチは、ユーザの特性に合わせて(本来は親指の



図3 トラッキングミス

移動), スイッチのオンオフを実行するのであるが, 実は, 親指以外の指が意図せず動いてもオンオフを実行してしまうという問題点があった。これはこのトラッキングアルゴリズムでは, 各フレーム画面の差分から動作検出を行っているので, WEBカメラで撮影されたユーザのどの部位が動いても, 「移動があった」→「スイッチが押された」と判断してしまうためであった。このままではユーザの意思を反映しない入力インタフェースとなり, 使用に問題がある。

## 2.4 コンピュータビジョンライブラリ

そこで本研究では, 必応な領域だけを選択してトラッキングを行うアルゴリズムに変更し, 本来のスイッチ動作と異なる意図しない動きにも対応できるようにした。ここではトラッキングアルゴリズムにBoofCV<sup>3</sup>を用いた。BoofCVは, リアルタイムコンピュータビジョンとロボット工学アプリケーションのためのオープンソースのJavaライブラリである。ライブラリの使用に関しては自分でコンパイルを行う必要があるが, 本研究では橋本の手順<sup>4</sup>を参考にしてコンパイルを行った。

本研究ではこれを拡張し, スイッチのトリガーとなるトラッキングの結果から, 撮影対象となる部位がどの方向に移動したかを上下左右の4方向を検出している。そのプログラム部分を図4に示す。同図Aが「測定対象がX軸に関する右方向移動判定」を行っている。更にここでは, 測定時に発生する余分なノイズの影響を除去するため, 測定値に対して時間軸で平滑化を行っている。

次に動作検出があった場合, 式番号149, 160, 169に示すようにシリアルポートに「R」, 「L」, 「D」の文字を送信している。これは各「右方向」, 「左方向」, 「下方向」等の移動を

示している。

更に本研究では図5に示すようにユーザの動作可能領域に合わせて動きを感知する領域をリアルタイムで決定できる。このプログラムを実行すると, 同図a)に示すように移動感知領域がマウスで選択可能となる。この感知領域は四角形で固定されており, 左上頂点でマウスをクリック, そのまま任意の位置にドラッグした後で, マウスのボタンを離すと, 右下頂点が決定される。これはWEBカメラで撮影された動画を見ながら行うので, サポートを行う支援者の誰でもが簡単に設定を行う事ができる。

このようにして親指部分を感知領域として設定したものが同図b)である。ここでは, 親指部分に重なっている四角形の領域のみが移動感知の領域となっているので, この状態において何らかの原因で他の指が突発的に動いたとしても, これらの動きは検知されない。よって本システムは, ユーザの意図しな

```

143 if(target.a.x - a.x >= def){
144     text("0", width/2, height/4);
145     cnt=cnt+1;
146     //println(cnt);
147     if (cnt>=3){
148         //text("R", width/2, height/4);
149         myPort.write('R');
150         //myPort.write('1');
151         cnt=0;
152     }
153 }
154
155 if(target.a.x - a.x <= -def){
156     text("0", width/2, height/4);
157     cnt=cnt+1;
158     if (cnt>=3){
159         //text("L", width/2, height/4);
160         myPort.write('L');
161         cnt=0;
162     }
163 }
164 if(target.a.y - a.y >= def){
165     text("0", width/2, height/4);
166     cnt=cnt+1;
167     if (cnt>=3){
168         //text("D", width/2, height/4);
169         myPort.write('D');
170         cnt=0;
171     }

```

図4 トラッキング検出部分 (一部)

## WEBカメラを用いた肢体不自由者用入力インターフェースの試作

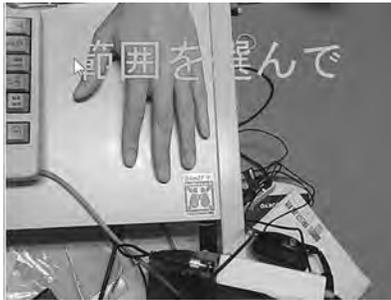


図 5a) 測定範囲決定



図 5b) 測定範囲指定 (親指)



図 5c) 測定範囲指定 (人差し指)

い動作が起こったとしても誤検知しにくいシステムとなっている。

また同図c) は同図b) を設定後、改めてマウスで感知領域を人差し指に設定し直したものである。この感知領域再設定は測定中の何時でも行えるので、ユーザの体勢が最初の位置とずれてしまい、感知不能になったような場合でも簡単に修正する事が可能になっている。

### 3. WEBカメラ入力インターフェースシステム

上記のユニットを組み合わせて作成したWEBカメラ入力インターフェースの概略を図6に示す。カメラとユーザの感知領域のセッティングが完了した後、実際の入力操作が行える。最初に①でWEBカメラが感知領域の動画撮影を行う、この動画はUSBインターフェースを介して1秒間に30フレーム、コンピュータに取り込まれる。このフレーム毎にトラッキング処理を行い移動の検出を行っている。移動があった時には、シリアルポートを介したシリアル通信でマイクロコントローラを用いた③HIDインターフェースに信号を送信している。これはシリアルポートを介して送信された信号に応じて、マウスが移動したり、クリックしたりといったマウスの操作だけでなく、キーボードからキーが押されたたと



図 6 WEBカメラ入力インターフェース

いう仮想的なマウス・キーボードとしての操作を行う事ができる。本研究では、1つのデバイスだけでWEBカメラからの画像処理結果から、あたかも人間が操作しているようなキーボードやマウスの操作を行うデバイスとなっている。

このインタフェースは組み込み機器の制御に用いられるマイクロコントローラを用いて製作した。本研究ではこのような用途で多く使用されているArduino互換機の「ダ・ヴィンチ32U with Arduino Bootloader」<sup>5</sup>を用いた。これは1チップで仮想的なシリアルポートやHIDポートを持っており、コンピュータのUSB端子とそのまま接続できるという利点がある。このため、非常に簡単はハードウェアとなり、学校や病院等の環境下においても、導入が簡単となっている。

この仮想インタフェースのプログラム部分を図7に示す。ここでは、上方向の動き「U」(式番号11)を受信すると、自動的にノートパットが起動し、続けてソフトウェアキーボードが起動する。これが同図Bのルーチンで実行されている。式番号17-20がWindowsの持つGUIの「ファイル名を指定して実行」を行っている。式番号21ではメモ帳を起動しているが、ここではメモ帳のファイル名「notepad.exe」を記述する事で実行している。ここに自分が実行したいファイルのファイル名を記述すれば、そのアプリケーションが起動する。同図Cでは同様にソフトウェアキーボードのファイル名「osk」を指定しソフトウェアキーボードを起動している。また、使用しているコンピュータ側ではソフトウェアキーボードに対して予めキースキャンの操作を「エンター」キーでトリガーするように設定しているので、エンターキーの操作だけで文字入力が可能になっている。図7には表示

```

11 | if (c == 'U') {
12 |     ↓
13 |     Mouse.move(120, 100, 0);
14 |     Mouse.move(120, 0, 0);
15 |     ↓
16 |     ↓
17 |     Keyboard.press(KEY_LEFT_GUI);
18 |     Keyboard.write('r');
19 |     Keyboard.release(KEY_LEFT_GUI);
20 |     delay(100);
21 |     Keyboard.print("notepad");
22 |     Keyboard.write(KEY_RETURN);
23 |     ↓
24 |     delay(500);
25 |     Keyboard.press(KEY_LEFT_GUI);
26 |     Keyboard.write('r');
27 |     Keyboard.release(KEY_LEFT_GUI);
28 |     delay(100);
29 |     Keyboard.print("osk");
30 |     Keyboard.write(KEY_RETURN);
31 |     ↓
32 |     delay(5000);
33 |     ↓
34 |     Mouse.move(0, 100, 0);
35 |     Mouse.click();
36 |     ↓
37 |     ↓
38 |     /*
39 |     if (c == 'D') {
40 |         Keyboard.press(KEY_LEFT_GUI);
41 |         Keyboard.press('R');
42 |         Keyboard.releaseAll();
43 |         Keyboard.println("osk");
44 |         Keyboard.press(KEY_RETURN);
45 |         Keyboard.releaseAll();
46 |         delay(1000);
47 |         ↓
48 |         ↓

```

図7 仮想インタフェースプログラム (一部)

されていないが、右方向の移動を検出すると「エンター」キーも出力されるようになっている。

このようにして文字を入力している画面を図8に示す。ここでは手の掌全体を位置検出領域として定義している。本システムも用いて実際に文字を入力してみた所、検出移動は問題なく行えているが、ソフトウェアキーボードを介して入力の制御を行った時のみ取りこぼしが多く発生し、実用には困難である事がわかった。

WEBカメラを用いたトラッキングによる

## WEBカメラを用いた肢体不自由者用入力インターフェースの試作

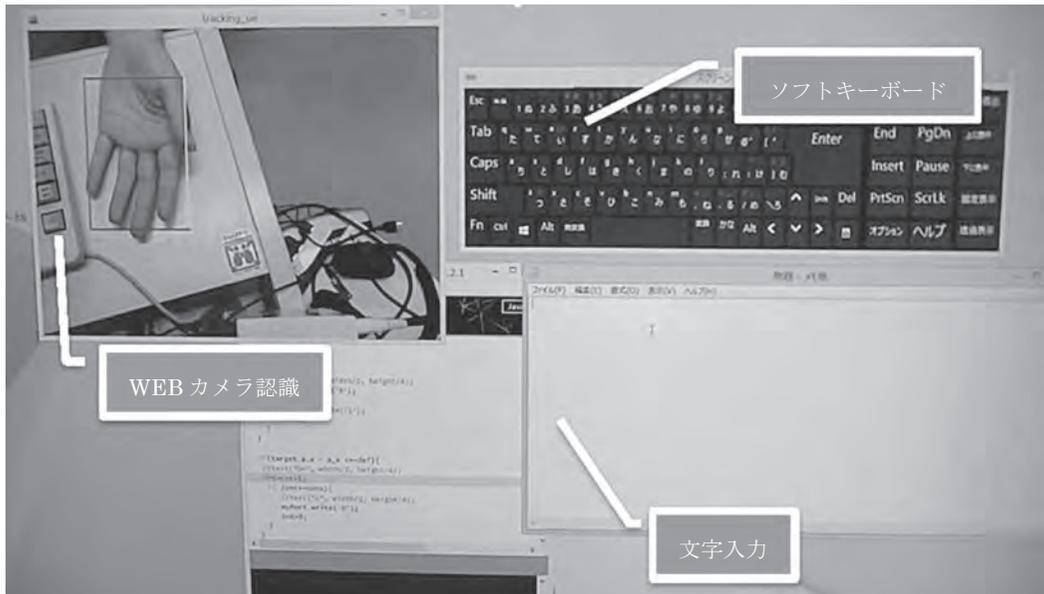


図8 WEBカメラを用いた文字入力画面

移動検出を単独でテストした時は問題なく移動を検出していた。測定例としてアナログ腕時計の秒針を測定したが、1秒毎に秒針の移動検出が行えた。また仮想インターフェースを用いてシリアル信号でエンターキーを入力させた時も取りこぼしなく、エンターキーを出力できていた。しかし、この両者を組み合わせるとソフトウェアキーボードの入力時に頻繁に取りこぼしが発生する事が確認できた。今後はソフトウェアキーボードを介したテストを行い、誤動作の原因を特定する必要がある。

#### 4. さいごに

本研究ではWEBカメラを用いて、撮影している動画から任意の部分をマウス操作だけで選択し、その領域のみを検出領域とするモーショントラッキング型の入力インターフェースを作成した。これはユーザの随意運

動が可能な部分のみをトリガーとする事で、ユーザの意図しない動作に影響されず、正確にユーザの意思に対応した入力インターフェースとなる。

さらにコンピュータの入力インターフェースとして使用されているマウスやキーボードといった入力機器と同じ動作を行う仮想インターフェースと組み合わせる事で、ユーザが普段使用しているソフトウェアがそのまま利用できるという特徴がある。しかし現在、この部分に動作上の問題があるので、これを改良する必要がある。その上で実際のユーザに使用感をフィードバックしてもらい実用化を目指すものである。

#### 5. 謝辞

本研究は、平成27年度科学研究費補助金(基盤研究(C))「マルチモーダル・インターフェースを応用した肢体不自由児における意

思表出構造の解明」(課題番号15K01460)の一部として行われたことを記して謝意を示す。

## 6. 参考文献

- <sup>1</sup> 「重度障害者用意思伝達装置」導入ガイドライン 2012-2013, 日本リハビリテーション工学協会, <http://www.resja.or.jp/com-gl/>
- <sup>2</sup> 機器紹介 ～支援ソフト・モニター～ なんでもスイッチUSB1, [\[ed.jp/shokus/joho/joho\\\_soft.htm\]\(http://ed.jp/shokus/joho/joho\_soft.htm\)](http://www.mie-c.</a></li></ol></div><div data-bbox=)

- <sup>3</sup> BoofCV, BoofCV 日本語情報トップページ —OSDN, [https://osdn.jp/projects/sfnet\\_boofcv/](https://osdn.jp/projects/sfnet_boofcv/)
- <sup>4</sup> ProcessingでコンピュータビジョンライブラリBoofCVを使う, <http://d.hatena.ne.jp/kougaku-navi/20140905/p1>
- <sup>5</sup> Da Vinci 32U w/Arduino Bootloader/ATMEGA32U4, <https://strawberry-linux.com/catalog/items?code=25005>