

Webカメラを用いた身振り認識によるユーザインタフェースの実現

大野 健太朗 弘畑 和秀

茨城工業高等専門学校 情報・電気電子工学専攻
〒312-8508 茨城県ひたちなか市中根 866

Web カメラは普及されてきているが、ビデオチャットや監視用のTVカメラなどの限られた用途でしかあまり用いられることはない。低画素のカメラは画像処理に向かず、ビデオチャットのように画像を転送して相手とのコミュニケーションを図る程度の処理しか行えない。しかし、昨今のカメラの画素向上にあたり画像を認識できる程度までカメラの精度が向上し、また安価に購入することが可能になった。本稿では、上記のようなカメラ性能の向上に伴い Web カメラのユーザインタフェースの実現を目指し、これによって、Web カメラの用途の拡張を狙う。

The Realization of User Interface by Gesture Recognition with Web Camera

Kentarou Ohno Kazuhide Hirohata

Ibaraki National College of Technology Advanced Course of Computer, Electrical and
Electronic Engineering

866 Nakane, Hitachinaka, Ibaraki, 312-8508 Japan

The Web camera has been spreading. But, It is not versatile. Because a main use of the Web camera is to do the video chats and to watch out. Low pixel camera is unfit for the image processing. So, it cannot use various ways. It can use only the simple way as the video chats. Because it has not performance as to forward the image data to the other party. But, the current camera improved. It is done improvement to be able to do the image recognize and we can buy it at a low price. In this paper, from improvement of the camera performance aim to realize the user interface by Web camera. And it aims to extend the usage by Web camera.

1 はじめに

今日、大容量のデータ転送を可能にしたADSL、光ファイバ等のブロードバンド通信の普及、PCの性能向上によりビデオチャット等が多く使用されるようになってきた。また、近年の犯罪の上昇傾向により監視システムの普及からTVカメラとして用いる等、Webカメラの普及率が上がってきた。しかし、その普及率とは裏腹に、その主要用途は上記のようにビデオチャット、TVカメラなど、限られた使い方しかされていない。そこで、Webカメラの他の使用用途を考えそれをユーザインタフェースとして提供できれば、よりWebカメラの普及率をあげ、人々のPCの操作環境を良くできるのではないだろうか。今のパソコンユーザはマウスで操作するのが当たり前になっている。しかし、マウスもPCが普及して一般的になったとは言え、いまだに年配の方や、PCに触れたことの人間にとっては、使い慣れないとうまく動かすことが出来ない。手の身振り動作は日々どんな人でもやっている動作で、非常に直感的で、わかりやすい動きである。そこで本稿では、Webカメラから手の身振り動作を検出し、その動作に対するアクションをアプリケーション上で起こすユーザインタフェースを実現させる流れについて報告する。こういったユーザインタフェースを実現させることでPCとは無縁だった人でも気軽に操作でき、またPCを日々使っている人でもマウス以上に直感的な操作が出来るので、画面に注視しなくても操作が楽に行えるだろうと考えられる。

2 Webカメラ

一般にパソコンのUSBポートに直接接続して、動画や静止画を撮影できるカメラのことを指し、Webカメラ、インターネットカメ

ラ、USBカメラなど様々な名称があるが全て同様のカメラである。

3 開発環境

3.1 周辺機器

本研究に用いたPCスペック、ならびに用いたWebカメラの仕様を以下に示す。

- Intel(R) Celeron(R) 2.80GHz
RAM 512MB
HDD 40GB
- Qcam® Orbit MP
有効画素数: True130 万画素 1280x960pixel
フレームレート: 最大 30 fps
USB2.0

3.2 開発環境

本研究におけるコンパイラならびに用いたライブラリを以下に示す。

- Visual C++ 6.0
- Java SDK2 1.4.2
- Intel Open Source Library : OpenCV

4 Webカメラからの身振り認識

本手法の流れを以下に示す。

STEP1:Webカメラからの画像入力。

STEP2:肌色認識。

STEP3:輪郭を囲む最小の矩形描画。

STEP4:矩形の中心点を導く。

次点での矩形の中心点同士を比較し身振りの方向を計算する。

STEP5:身振り認識

STEP6:アプリケーション上でのアクション

5 詳細

5.1 Webカメラからの画像入力

今回Webカメラからの画像入力、肌色認識

等において Intel からフリーで公開されている OpenCV ライブラリを用いる。

5.2 肌色認識

手の身振り動作を調べるには動かす手を認識させてそこから輪郭をとる必要がある。そこで、手を認識させる為に全体画像から肌色の部分のみを検出する。この際、OpenCV の関数を用いて、肌色のマスクを作成し、現画像との論理積をとることで現画像から肌色だけの部分を取り出す。用いる表相系は YCrCb 色を用いて行う。YCrCb は RGB の線形変換の式に当てはめることによって得られるが、OpenCV において YCrCb から RGB へ変換する関数が用意されているのでそれを用いる。Y は輝度, Cr, Cb 値は人種に依らず、 $133 \leq Cr \leq 173$ かつ、 $77 \leq Cb \leq 127$ を満たすので、この条件のとき、現画像との論理積をとる。今回、カメラから画像を得る環境としては一般的な部屋で、蛍光灯や日光等の光が入り混じる環境を主として画像の検出を行った。肌色の認識は光の反射などによって非常に変わりやすいもので、今回のシステムの実現においても、光の影響等で肌色がうまくとれない場面が多く生じた。閾値の値をいくつか変えてみたが、なかなか完全に肌色を抽出するのは難しく、今後の研究で肌色の検出部分についても、アルゴリズムの改良や、他の表相系で肌色検出を試せればと思う。

5.3 輪郭を囲む最小の矩形

肌色を検出した画像から輪郭を取り出す。その取り出した輪郭を囲む最小の矩形を描き出す。その際、OpenCV の関数を用いて、リアルタイムに描画を行う。

また、その際、光の入りかたでところどころ

で肌色以外の小さな雑画像が入ってしまい、その輪郭をとってしまうことで雑画像の矩形を描くという問題が発生する。それを防ぐために、ある一定以上の輪郭の大きさの場合のみしか矩形を描かないという条件を追加し、雑画像による誤差を少なくした。

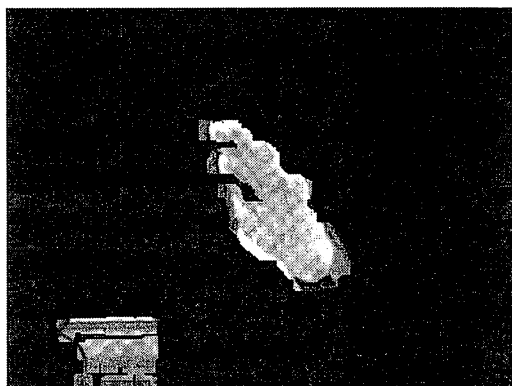


図1 雑画像による手の輪郭矩形描画失敗

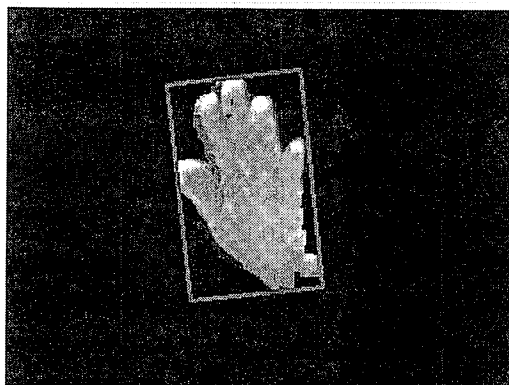


図2 条件追加後の輪郭矩形画像

肌色の認識が甘い部分が多く、輪郭をうまく取れないことが問題としてあげられた。また、顔と手が同じ肌色であるためカメラの位置によっては顔の画像の輪郭をとってしまうこともある。そういった問題を解決するためにも、肌色認識部の仕様を見直すこと、顔の画像を認識できないようにする等の処理を行い、輪

郭をとらなければならない。顔の画像を認識させず手だけの画像を得るためには、顔の画像を認識しその顔の画像と、手と顔の映っている画像との論理積をとり、手だけを認識させるような流れで行えばよいと考えられる。今後の研究において、そういったところ問題の検討も必要である。

5.4 矩形の中心点抽出

描いた矩形から中心点を導き出す。中心点の座標は矩形の4点の座標をそれぞれ足し合わせ、要素の4で割った値とする。

導き出した中心点を、次点における中心点と比較し、まず、縦方向であるか、横方向であるかを調べる。縦方向であった場合、それが上方向であるか、下方向であるかを、二点間の座標のY座標から判断する。また、横方向であった場合、それが右方向であるか、左方向であるかを、同様に、X座標から判断する。縦方向であるか、横方向であるかの判断は、縦方向の動きである場合、初めに検出した中心点からX座標 ± 10 ピクセルの間で動きが検出されたときとし、横方向は初めに検出した中心点からY座標 ± 10 ピクセルである場合とする。その検出を一定時間行い、各方向に動いた座標の要素数を全て足し合わせる。誤差も生じるので全体の要素数の半分以上方向の要素があったとき、縦方向であるか、横方向であるかを最終的に判断する。

5.5 身振り認識

動きの方向を判断できたら、それを各方向の身振りとして認識する。

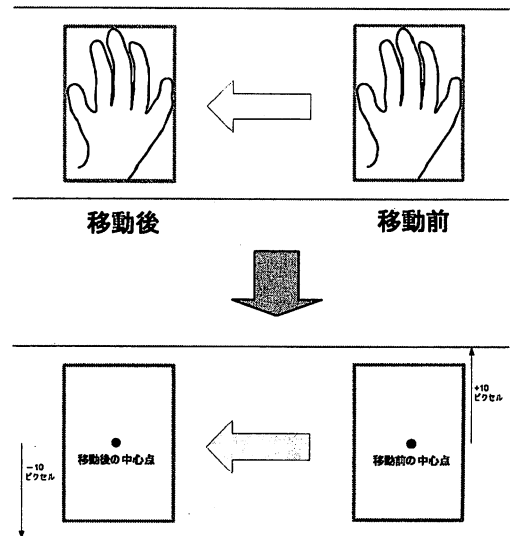


図3 中心点導出後の横(左)方向認識

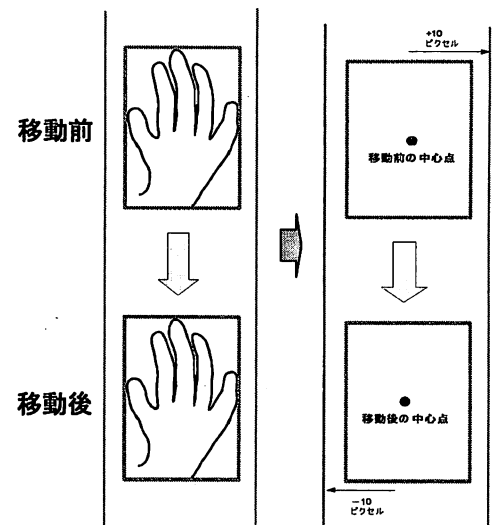


図4 中心点導出後の縦(下)方向認識

5.6 ユーザインタフェース

身振り認識後に、実際にアプリケーション上でアクションを起こす。今回は、Webブラウザで多く使われる、マウスジェスチャを身振りで行えるようなインターフェイスを実現

させる。マウスジェスチャはブラウザ上で直線等をマウスで描くことで、ブラウザの進むボタンを押されるのと同じアクションを起こしたり、戻るボタンや、ウインドウを閉じるなどといった機能をより効率的に行える拡張機能である。そこで、今回カメラから身振りを認識し、ブラウザ上で進むボタンや戻るボタンを押されるのと同じ動きをする機能を実現させることで、マウスの操作より直感的にブラウザの操作を行うことができ、マウスの使用を最終的には無くせるようなユーザインタフェースの実現を目指す。今回、ブラウザの作成は Java で行い、画像処理は C 言語で行う。実際、今回は手の動きが左から右へ移動した際、ブラウザのページが進む機能、また、右から左への動きをした際、ページが戻る機能の作成を行った。今回、「戻る」、「進む」、機能のみしか機能の作成を行えなかったが、身振り認識の段階でより複雑な動き等を認識、解析できれば、ブラウザを操作する上で必要な機能は全て追加できる。最終的にはそのようなブラウザの作成を目指したい。

6 おわりに

本研究では、Web カメラ等の手軽に購入でき、簡単に接続できるカメラを用いてより直接的・感覚的な操作を行うことが出来る手の身振りを検出し、それに対するユーザインタフェースを実現させることを目的として行った。今回は、Web ブラウザを作成し、そのブラウザ上で身振り動作による機能の拡張を行ったが、本システムを有効に使うことでさらに他のアプリケーションやユーザインタフェースとしての機能を実現できる。例えば、手話翻訳や、OS 上でマウスの代わりとして使える機能など、身振りを認識させてそれに対してア

クションを行うシステムの開発は多種多様に可能である。そういった、システムについても今後の研究対象として検討を行い、実現していきたい。

文 献

- [1] 佐々木整, “本格学習 Java 入門” 技術評論社
- [2] Intel, “OpenCV リファレンス”
- [3] 興石欣吾 “自発的にコミュニケーションを図るロボットのための判断基準の構築”, 奈良先端科学技術大学院大学 情報科学研究科情報処理学専攻 修士論文, pp34.
- [4] Webカメラをつけよう!
”<http://www.h2.dion.ne.jp/~oga/webcam.html>”
- [5] Japan.internet.com デイリーリサーチ -[監視用カメラ]に隠れたチャンス潜む? Webカメラの利用動向,
”<http://japan.internet.com/research/20040218/1.html>”
- [6] JavaでHello World JNI編,
”<http://www.hellohiro.com/native.htm>”