

修士論文概要書

CD

2008 年 7 月提出

学籍番号3606U121-9

専攻名 (専門分野)	情報・ネットワーク	氏 名	中村 友一	指 導 教 員	中島 達夫 印
研究指導名	分散システム研究				
研 究 題 目	Web カメラを用いたリアルタイム文字認識システム				

1 はじめに

近年検索エンジンの性能向上や、Wikipedia などの Web サイトなどにより、インターネット上で様々なことを学習することが可能になってきたが、まだまだ Web 上にない情報も多く、書籍などの紙媒体もまだまだ重要である。しかし、紙媒体のままの文章は検索性が悪いなどの問題点がある。最近ではドキュメントスキャナや専用のブックスキャナも個人で購入できるが、まだまだ高価であったり、書籍を裁断する必要性があったりする。そこで、Web カメラを頭部に設置し、本を読みながらリアルタイムで文字を認識、さらに読んでいる際の視線情報を取得すればそれらの問題を解決でき、文章に対するアノテーションも可能になると考え、その検証のためのシステムを開発した。本論文では試作したシステムの概要と、その検証結果に付いて述べる。

2 システム概要

本システムではデスクトップコンピュータ及び視線認識・文字認識用に 130 万画素の Web カメラを 2 台使用した。カメラは自作の支えを使用し眼鏡に固定してある。開発に使用した言語は使用した開発環境は Visual Studio 2008 (C# 3.0)である。カメラ部の外観を図 1 に示す。



図 1: カメラ部外観

2.1 処理概要

本システムでは視線認識用カメラからの入力画像より視線位置を、文字認識用カメラより指先の位置を認識し、その付近の画像を切り出した後、文字認識処理を行う。全体の処理フローを図 2 に示す。

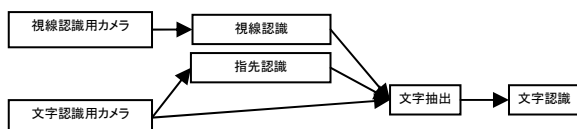


図 2: 本システムの処理フロー

2.2 視線認識

視線認識用カメラの入力画像から、文字認識用カメラ中の注目点の座標求める。視線認識の処理フローを図 3 に示す。

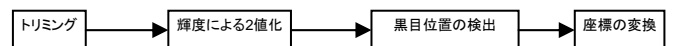


図 3: 視線認識の処理フロー

1. トリミング

視線認識用のカメラの入力画像から、目の部分だけをトリミングする

2. 輝度による 2 値化

黒目位置を検出しやすくするため、輝度による 2 値化を行う

3. 黒目位置の検出

2 値化の結果を基に、周辺分布を用いて黒目の位置を検出

4. 座標の変換

得られた黒目位置を基に、文字認識用カメラのどこを見ているかの座標の変換を行う

2.3 指先認識

文字認識用カメラの入力画像から、指先にあたる点の座標求める。指先認識の処理フローを図 4 に示す。

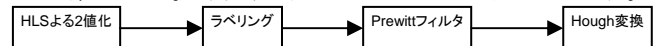


図 4: 指先認識の処理フロー

1. HLS による 2 値化

文字認識用カメラの入力画像から HLS 色空間を利用して画像の肌色部分を抽出

2. ラベリング

検出された肌色部分の最大の連結画素を抽出

3. Prewitt フィルタ

直線検出のために、Prewitt フィルタを利用して輪郭を抽出

4. Hough 変換

画像中の直線の検出し指の方向とし、その先端部分の座標を求める

2.4 文字抽出

得られた視線、指先の座標から該当部分の画像を抜き出し、傾きを補正した後、文字の抽出を行う。文字抽出のフローを図 5 に示す。

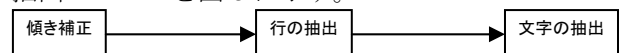


図 5: 文字抽出のフロー

1. 傾き補正

Hough 変換を行い画像の傾きを補正

2. 行の抽出

周辺分布を用いて行を抽出

3. 文字の抽出

文字の高さを考慮し周辺分布を用いて文字を抽出

2.5 文字認識

得られた文字画像より文字の認識を行う。文字認識の処理フローを図 6 に示す。

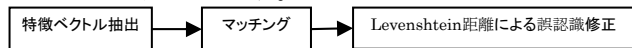


図 6: 文字認識の処理フロー

1. 特徴ベクトル抽出

周辺分布とクロスカウント法を併用し特徴ベクトルを求める

2. マッチング

あらかじめ求めておいたテンプレート画像の特徴ベクトルとのマッチングを行う

3. Levenshtein 距離による誤認識修正

辞書を用いて誤認識結果の補正を行う

3 検証・評価

今回、認識精度と処理速度の面から評価を行った。

認識精度に関しては視線、指先位置の検出は良好で、実用に耐えるレベルであったが、文字認識がカメラの解像度不足から文字つぶれを起こしてしまい、低解像度時の漢字の認識率が極端に悪かった。結果を表 1, 表 2 に示す。

種別	分解能
カメラ入力	1280×1024px
黒目位置	400×300px
座標変換後の視線情報	160×120px
指先認識	320×240px

表 1: 各入力 of 分解能

入力画像	認識率
20×20px(ひらがな)	約 70%
20×20px(漢字)	1%未満
60×60px(漢字)	約 90%

表 2: 入力画像に対する文字認識率

処理速度に関しては各処理において最適化を行った結果、およそ 1~2FPS であった。更なる処理の最適化とアルゴリズムの見直しを行えば改善すると思われる。各処理の処理時間を表 3 に示す。

処理内容	処理時間
カメラからの静止画のキャプチャ	81ms
視線位置の取得	46ms
指先位置の取得	49ms
文章の傾き補正	144ms
文字の抽出	75ms
文字の認識(1 文字)	60ms
認識誤差の修正	170ms

表 3: 各処理の処理時間
(Athlon X2 5600+ 2.8GHz Dual-Core)

4. 将来課題

将来課題としては、認識精度、処理速度の向上、半透過 HMD などと組み合わせた分かりやすいインターフェイスの構築が挙げられる。処理速度に関してはページ全体を一度に取り込むことを目標としたい。認識率の向上に関しては、抽出した文字の分解能が一定以上であれば認識率は高くなることから、光学的な補正、コンポジット処理（超解像）、高画質なカメラなどを用いれば解決できると考える。

5. 結論

今回は解像度の面から、認識精度がいまひとつであったが、書籍を読みながらのリアルタイムでの文字認識は可能であると考ええる。

デバイスが小型化し、精度が上がればコンピュータの画面や日常の視覚データのすべてを電子化し、有効活用できるようになるであろう。また、HMD などのディスプレイデバイスと組み合わせることにより AR 的な応用も期待できる。

6. 参考文献

- [1] 酒井幸市「デジタル画像処理の基礎と応用」CQ 出版、2003
- [2] 酒井幸市「画像処理とパターン認識入門」北森出版、2006
- [3] 須崎亮太郎、荻野友隆、内村創「フルスクラッチによるグラフィックスプログラミング入門」秀和システム、2004