2008－CVIM－165 (19)
2008／11／27

視覚障害者用のウェアラブルな文字認識デバイス

田中 誠[†1] 後藤英昭[†2]

景観中の文字が読めないことは，視覚障害者の日常生活において多大な負担となっている．彼らへの視覚代替装置として，身につけられるカメラを用いた文字認識システムが有用であると考えられる．動画像を利用して連続的な処理を行うことで，利便性を向上させることができるが，複数枚の入力画像に同じ文字が出現するため，文字の抽出だけでなく追跡処理も行う必要がある．本研究では文字の追跡処理に粒子フィルタを用いることによって，文字認識の効率化を図った．その結果，認識対象となる文字候補領域を，トラッキングを用いない場合と比べて1.15%まで大幅に削減することができた．提案手法は一般的なノートPC上で準実時間の10fpsで動く．

Wearable Reading Assistant Device for the Blind

Makoto Tanaka[†1] and Hideaki Goto[†2]

Disability of reading text in natural scenes imposes a huge impact on the Quality of Life of the blind. One of the most anticipated devices is a wearable camera capable of character recognition. The usability of such a device can be improved using an on-the-fly processing of video sequences. Since the same text regions appear in consecutive video frames, the device needs to track text regions automatically. We have developed a text tracking method based on the particle filtering to make the system more efficient. The number of text regions to be recognized has been reduced down to 1.15The proposed method runs at a near-real-time speed of 10fps on a typical laptop computer.

1. はじめに

文字は，人間が生活する上で重要な情報を提供してくれている．看板や標識，電化製品のボタンなど，実世界中の様々な場所に書いてある文字を読むことで，人間は円滑に生活を営んでいる．しかしながら，文字を読むのに支障がある視覚障害者は，これらの文字情報を読むことができないため，日常生活において不便を強いられている．

視覚障害者が文字情報を利用する手段として，点字などがあげられる．しかし，点字を覚えるには多大なる労を要し，点字文書を作る側も多大な手間を要する．その上，屋外にある標識や店の看板などには触れることができない場合が多く，それらの文字に対して点字が利用できることはほとんどない．そこで，非接触型で自動的に文字を読み取り，音声合成などにより内容を伝えるシステムの開発が求められている．

このような視覚代替装置の利用形態を考えると，カメラの動きが遅かったりほとんど動かない場合も多く，複数枚の画像に同じ文字が出現することが多い．これらの文字イメージのすべてを文字認識にかけることは，処理速度の面で現実的ではない．また，同一の文字から得られた情報を利用者に繰り返し提示することは，利便性を損なうことになる．従って，ビデオ映像からの情景文字の抽出では，同一の文字領域を追跡（トラッキング）して，グルーピングする処理が必要である．

現在，トラッキングに関しては主にコンピュータビジョンの分野で研究がなされている．例えば歩いている人間などをトラッキングする方法があるが，これらの手法は動体検出が主に用いられており，電光掲示板や動物体にプリントされている場合を除いて文字が実空間上で動くことはあまりないのでこのような手法をそのまま用いることはできない．そのため，文字領域に適したトラッキング方法を開発する必要があると考えられる．

我々は過去にアクティブカメラによる文字抽出ロボットに関する研究を行った[1)]．この研究では，景観画像中からの文字の抽出を行い，それら文字に関する簡単なトラッキングまで行っている．しかし，この研究で

†1 東北大学大学院情報科学研究科, Tohoku University Graduate School of Information Science
†2 東北大学サイバーサイエンスセンター Tohoku University CyberScience Center

は，ロボットの前進運動しか考慮されておらず，カメラの動きもあらかじめ決められた向きに拘束されている．人が身につけるウェアラブルカメラを考えると，ある程度激しい動きも想定されるため，トラッキングの更なるロバスト性が求められる．

文字領域のトラッキング処理についてもいくつか提案されている．Li[2] らはモーションベクトルを用いてトラッキングを行っている．しかし，ほぼ一定の速度の動きをする文字にしか対応できないという問題がある．Myers ら[3] は RANSAC を用いてトラッキングを行っている．この手法では，精度を得るためには計算時間がかかるという欠点があるが，ある程度汚い画像でもロバストにトラッキングできるという利点を持つ．Mirmehdi ら[4] は粒子フィルタという手法を用いてトラッキングを行っている．しかし，これらの手法は文字領域の抽出部分を手動で行っていたり，連結要素の手法が使えない文字には用いることができない．また，十分な評価も行われていない．

シーン文字の認識では，トラッキングにおいてグループ化された複数の文字画像のうちのどれを文字認識処理に渡すかを決める必要も出てくる．

そこで，本研究では，新たに文字領域の追跡に適したトラッキング手法を提案する．まず，従来手法である DCT 法を用いて画像中から文字ブロックを抽出し，それらブロックをつなぎ合わせて文字領域とする．これらの領域に粒子フィルタを適用してトラッキングを行った．

本研究では，視覚障害者用の視覚補助措置として，ウェアラブルカメラを用いて景観中文字の自動抽出ができるシステムを構築することを考える．また，本研究は，旅行者用の視覚情報による言語翻訳機やロボットの景観中文字認識にも応用できると考えている．

本報告の構成は以下の通りである．2 章では，本研究でのシステムの概要と改善点を述べる．3 章ではそれら改善手法の評価を行う．4 章がまとめである．

2. システムの構成

2.1 システムの概要

本研究で構築中のシステムの概要を図 1 に示す．

まず，ウェアラブルカメラによって景観の動画像を撮影し，その動画像をコンピュータに取り込む．画像から文字情報を抽出した後，文字認識処理により文字情報を文字コード化する．文字がコード化されたことにより，音声合成ソフトウェアなどで文字情報の変換が可能になる．

処理装置の部分ではまず，景観画像から文字の部分を抽出する処理が必要である．また，この抽出された文字の内容を認識する処理も必要である．さらに，時間的に連続した画像フレームでは，同じような位置から同じような方向を撮影することが多いため，同じ文字領域が複数回連続して現れることになる．そこで，これを考慮して，文字領域の対応付けを行い，それらをグループ化するトラッキング処理が必要になってくる．よって，処理装置の部分ではこれら 3 つの処理を行うことが必要であると考えられる．今回試作したシステムの外観を図 2 に示す．

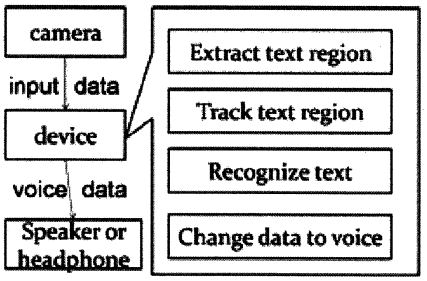

図 1　システムの概要
Fig. 1　system

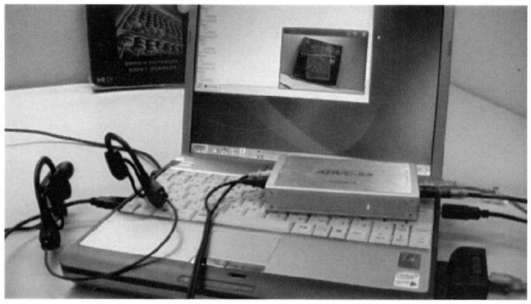

図 2　試作したシステムの外観

なお，本研究では文字領域の抽出とトラッキング処理に重点を置き，文字の認識処理は扱わない．

2.2 背景からの文字の抽出処理

文字抽出の手法として，周波数解析を用いる手法がいくつか提案されているが，文献[6] の調査結果から，低周波数領域の中でさらに周波数成分が低い部分を除いた部分を用いる方法が，他の従来手法よりも優れているという結果が出ている．そこで，本研究では[6] の手法を用いることにした．

DCT 法ではまず，入力された画像を 16 × 16 のピクセルのブロック単位に分割する．次に，それぞれのブロックに DCT を施し，ブロックごとに特徴量を求める．画像中の全てのブロックの特徴量の値を元に，判別分析閾値法[6] により閾値を求める．閾値より高い

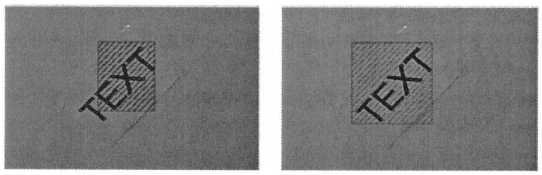

図 3 領域拡張による認識結果の差異 (左:従来手法[1], 右:改良手法)

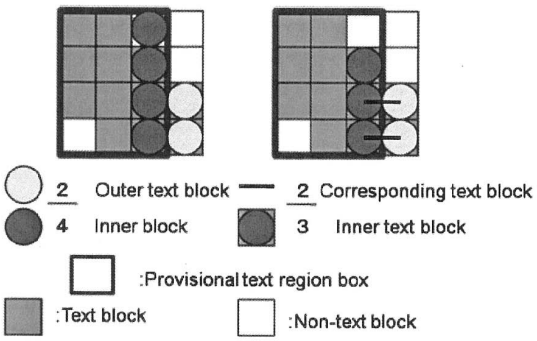

図 4 逐次領域拡張法の改良 (左:従来手法[1], 右:改良手法)

特徴量を持つブロックを文字ブロックと判定する.

文字ブロックの抽出を終えると，これらを併合して文字領域を作成する．これは，文献[5] で提案された逐次文字領域拡張法を改良したものを用いる．図 3 に示すような斜めの文章の抽出率を改善するために，領域拡張の方法を若干改良した．文字領域の作成は，領域を上下左右のいずれかの方向に逐次拡張して行う (図 4).

始めに，領域の最上部の領域に存在する文字ブロックを数える. 次に, そのブロックにつながっている領域の外 (この場合は上) の部分にあるブロックを数え，先ほどの文字ブロック数との割合を調べる．これを同様に左，下，右方向にも行い，それぞれの文字ブロックの割合を調べる．その文字ブロックの割合が一番高かった方向を領域の拡張する方向とする．この拡張する方向の文字ブロックの割合が閾値より高かった場合は，領域を拡張し，同様の操作を繰り返す．これを閾値より低くなるまで続け，拡張する方向がなくなったら，次の文字領域の拡張へと移る．図 3 の例では右方向の文字ブロックの割合が 2/3 なので，領域を右方向に拡張する.

DCT 法では，領域内に壁と床などの境界部分も文字として誤抽出しやすいという問題がある．この誤抽出を文献[7] で提案されたエッジカウントにより削減する．エッジカウントではまず，大津の二値化[8] により文字として抽出された領域の画像を二値化する．その領域の矩形の長さ方向にエッジを数え，その平均が閾値以下なら物と物の境界部分であるとして，削除する．文献[7] の考察からさらに改良を加え，本研究では閾値を 3 とアスペクト比+1 のいずれか大きい方とする．ただし, カメラの激しい動きなどによって撮影画像がぶれてしまい, 文字領域が閾値を下回る可能性があるため, 次節で述べるトラッキングの処理については，削除対象のブロックについても行っている.

2.3 トラッキング処理

2.3.1 粒子フィルタ

本研究では粒子フィルタを用いてトラッキングを行う．粒子フィルタによる領域追跡はロボットビジョンの世界で広く用いられており，その有用性は実証されている．DCT 法による文字抽出を用いた本手法に粒子フィルタを組み合わせることで, ロバストなトラッキングを実現することを考える.

粒子フィルタとは観測対象の移動先を確率的に求める方法である．ここでいう観測対象とは，現在のシステムの状態の中で知りたいものを指す．例として，観測したい物体の位置や角度，速度や加速度などがあげられる.

この粒子フィルタは大きく分けて 3 つのステップで現在の観測対象を推定する．その概略を図 5 に示し，以下にその内容を説明する.

1 つ目のステップは予測を行う．観測対象を多数の粒子で表わし，それぞれが遷移する可能性がある移動先に遷移を行う．なお，今回の実験では観測対象は単純に文字領域候補の中心の x 座標，y 座標の値とする．また，粒子の移動推定のモデルには前回座標との差分を用いた 1 次のモーションベクトルにガウス分布に基づく乱数を足し合わせた.

2 つ目のステップでは重みづけを行う．これは，先ほどの予測ステップで移動させた各粒子の位置の確からしさ (以下尤度と呼ぶ) を調べ，各粒子に対する重みを変更する．この尤度を決める方法にはいろいろなものが考えられるが，今回は簡単に文字領域に属している粒子の重みとして 1 を，属していない粒子の重みとして 0 を掛けることにする．また，1 ループ毎の結果を出力したい場合は，ここで出力する．本実験では，重みづけをした粒子の重心を文字領域候補の現状態として出力させている．この重心との距離が一番近い文字領域をトラッキングされた文字領域として出力している.

3 つ目のステップではサンプリングをやり直す．重みの総和を計算し，粒子数の総和を変えずに，重みに応じた粒子数にする.

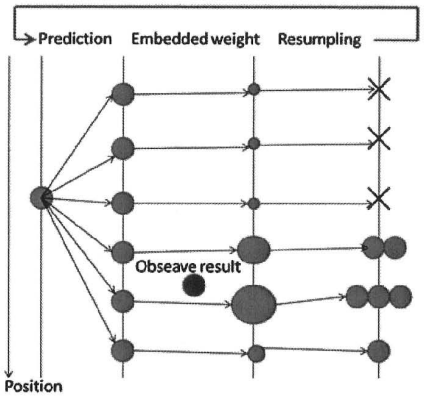

図 5　粒子フィルタの処理概要

2.3.2　トラッキング方法

トラッキング処理は，始めに 16 × 16 ピクセルのブロック間での対応付けを行い (図 6)，その結果を用いて領域間の対応付けを行う (図 7) という二段階の処理で行っている．

まず，文字領域の区別のために全ての文字領域に固有のラベルを持たせる．文字領域内のブロック全てに，文字領域と同じラベルを持たせておく．ブロックの類似度の指標として，比較するブロックのそれぞれの内部の濃淡値の累積ヒストグラムを用いる．類似度を市街地距離で定義し，これが一番小さいブロックを対応ブロックとする．比較するブロックは，先述の粒子フィルタを粒子の設定を文字ブロックの位置として用いて粒子が存在する場所とした．

全ての文字ブロックの対応付けが終わったら，文字領域の対応付けを行う．文字領域の対応付けは，その領域内のブロックの多数決で決定する．しかし，全ブロック数に占める割合が閾値以下であった場合は，新しい文字領域とする．同じ画像に同じラベルが複数存在する場合は，前のフレームでの元の領域と横幅が近いものを過去の画像からトラッキングされた対応領域とし，その他の文字領域には，新しいラベルを割り当てる．

今回，ノイズや文字の境界部分による文字領域特定のゆらぎにより，連続画像間で同じ文字領域が 2 つ以上に分離したり 1 つに統合したりを繰り返す領域が存在した．こういった場合，今までのラベル付けの方法ではラベルの競合が起こり，正しくトラッキングできないという問題が生じる．そのため，同一のラベルで文字領域の距離が比較的近いときは同じ文字領域と判断することにした．

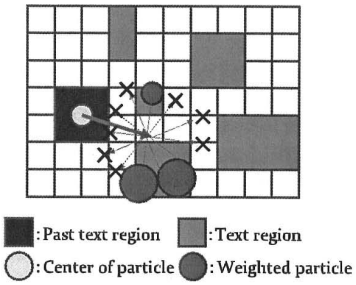

図 6　ブロック間のトラッキング

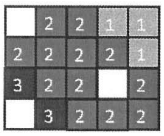

図 7　文字領域のラベル付

2.4　画像の選択

トラッキングを行うと，同じ文字領域であると判断された画像が複数枚出力される．これらすべてに文字認識処理を行うのは処理時間および出力データの重複の観点から見て望ましくないので，文字認識に適した出力画像を自動選択する必要がある．本研究では，カメラが文字に一番近づいた場合や抽出する文字部分が欠けなかった場合を考え，横幅が一番長い出力画像を最終的に文字認識を行う画像とした．

3.　実験結果と考察

3.1　実験環境

提案したトラッキング処理の有効性を検証する．本実験に使用した機器を述べる．カメラには 38 万画素のカラー CCD カメラの SVR-41Versatile ヘッドセットカメラを用いた．これを NTSC-DV コンバータで変換し,352 × 224 の解像度で画像をノート PC に取り込んだ．この PC は Intel Core2Duo(1.06GHz) を搭載し，またその OS には SUSE Linux を用いている．文字領域トラッキングの処理結果を図 8 に示す．なお，取得できた文字領域候補の総数は 5,192 枚であった．廊下の 8 箇所に文字領域を設定したので，抽出すべき領域の数は全部で 8 である．

3.2　結果と考察

以上の手法でトラッキングを行った結果および今回用いた粒子フィルタによる文字領域の抽出枚数とトラッキングを用いた最終結果をともに表 1 に示す．なお，エッジカウント有とはトラッキング時にエッジカウン

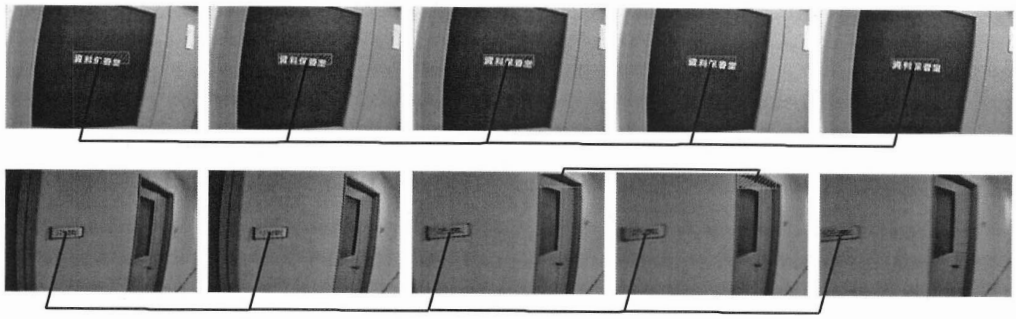

図 8　文字領域トラッキングの例

表 1　それぞれの処理の段階における文字領域候補の数

	edgecount		no edgecount	
	image	time	image	time
Conventional[1)]	113	56	185	56
Region-based	111	56	176	56
Block-based	86	81	158	81

(msec)

表 2　文字領域統合の距離しきい値と画像数

distance allowance	0	1	2	∞
obtained images	86	60	65	53

(time=82msec)

トによる削減を行った場合のことを指す. また,Region-based とは粒子フィルタを文字領域に対して適用したもの, Block-based とは粒子フィルタを文字ブロックに適用したものを表す.

この結果をみると, 従来の手法と Region-based の差はあまりないが,Block-based が一番効率よくトラッキングを行っていることがわかる. また, 今回用いた DCT 法ではドアのふちなどの誤った部分も抽出してしまうが, これらをエッジカウントを用いることによって減らすことができた. その結果, 最終的に 86 枚 (文字領域候補数の 1.66%) にまで文字認識処理を行う回数を減らすことができた.

しかし, 現時点では最終抽出目標の枚数は 8 枚なので, まだ差がみられる. そこで, 領域同士がある程度近い場合においてもラベルが等しければ統合するという処理 (2.3.2 参照) を加えた. この領域間の距離を変更して実験を行った結果を表 2 に示す. なお,distance allowance の値は, 何ブロックの距離を許容するかを表している.

この結果から, 一見領域間距離を無視してラベルが同じものをすべて統合すると文字認識処理を行う枚数を減らすことができることがわかる. しかし, この統合方法では, 画像にあるすべての文字領域候補を誤って統合してしまっており, その後の認識処理に向かない. そこで, 次点の領域間距離 1 ブロックが最適であると考えられる. また, 統合処理による処理時間の増加はほとんどなく, 全体の処理時間をほぼ実時間 (10〜11fps) に抑えることができた. 最終的には文字領域候補数を 60 枚 (文字領域候補数の 1.15%) まで減らすことができた. このような処理を行ってもまだ, 目標の 8 枚の 7.5 倍の文字領域候補が残るため, さらなる改良が必要であると考えられる.

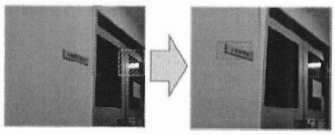

図 9　マッチングのミスの例

処理結果をさらに詳しく見ると, 領域内の画像があまり似ていないにも関わらず, 近傍の領域を対応付けてしまった例が認められる (図 9). これは, 従来手法[1)]においても表れていた事象であり, 累積ヒストグラムを用いた類似度による明確な閾値を定めることができなかったため, 誤った対応付けを修正することができなかった点である. ノイズが激しいウェアラブルカメラを用いた場合は, このような文字領域抽出が不完全な時の誤認識が起こる可能性が高くなるので, 認識精度を上げるような明確な閾値の定め方やより優れた手法が今後必要であると考えている.

4. ま と め

本研究では, 始めに景観中からの文字情報抽出において, トラッキング処理が必要であることを述べた. そこで, 既存の文字領域抽出手法にトラッキング手法として数多く用いられ, その有用性が実証されている粒子フィルタに着目し, これを組み合わせることでより精度の高い文字抽出システムの実現を目指した. ま

た, 既存の文字抽出手法を改良し, 最終的には 1.15%の文字領域候補に絞り込むことができた. しかし, 目標数にはまだ差があるため, さらなる手法の改良が求められる.

また, 今回はトラッキングによりグループ化した画像群からの OCR 処理を施す画像の抜き出しの基準は「横幅が長いこと」としたが, 実際には文字以外の画像が紛れ込んでしまったり, ぼやけた画像を選択してしまって文字が読めない場合も存在する. そのため, 文字抽出・認識処理に適した画像を選択する基準についても研究していく必要があると考えられる.

参 考 文 献

1) M.Tanaka,H.Goto, "Autonomous Text Capturing Robot Using Improved DCT Feature and Text Tracking," International Conference on Document Analysis and Recognition, Vol2, pp.1178-1182, 2007.
2) Huiping Li, David Doermann, and Omid Kia, "Automatic Text Detection and Tracking in Digital Video," IEEE TRANSACTIONS ON IMAGE PROCESSING, VOL. 9, NO. 1, 2000,pp.147-55,
3) Gregory K.Myers, Brian Burns, "A Robust Method for Tracking Scene Text in Video Imagery,"First International Workshop on Camera-Based Document Analysis and Recognition, pp.30-35, 2005.
4) Carlos Merino,Majid Mirmehdi, "A Framework Towards Realtime Detection and Tracking of Text,"Second International Workshop on Camera-Based Document Analysis and Recognition, pp.10-17, 2007.
5) 齋藤靖二,"シーン中からの文字情報抽出に関する研究," 東北大学大学院情報科学研究科修士学位論文, 2005.
6) H. Goto, "Redefining the DCT-based feature for scene text detection," International Journal on Document Analysis and Recognition (IJDAR). Vol.11, No.1, pp.1-8, 2008.
7) Hiroki Shiratori, Hideaki Goto, Hiroaki Kobayashi, "An Efficient Text Capture Method for Moving Robots Using DCT Feature and Text Tracking," Proceedings of 18th International Conference on Pattern Recognition (ICPR),2, pp.1050-1053, 2006.
8) 大津展之, "判別および最小 2 乗規準に基づく自動しきい値設定法," 電子情報通信学会論文誌 D, Vol.J63-D, No.4, pp.349-356, 1980.