

小型カメラを用いた文字入力ユーザーインターフェースの提案

3C-02

西山京助 仙田修司 中島昇 山田敬嗣
NEC インターネットシステム研究所

1 はじめに

近年、カメラモジュールの省電力化および小型化が進み、携帯端末等に接続できる小型のカメラデバイスが実現されている。筆者らは、小型カメラが搭載された携帯端末と文字認識技術を用いて、紙面に印刷された文字を容易に入力できるユーザーインターフェース (以下 UI) の実現に向けて研究開発を進めている。

従来から、カメラと文字認識を組み合わせた研究としては^{[1][2]}などが報告されている。これらはカメラの利便性を活かして、スキャナで取り込めないような情景画像からの文字認識を実行できる。一方で、印刷文書を接写して認識するという使い方もある。例えば名刺を撮影して個人情報を入力したり、雑誌や広告に印刷された単語を撮影して Web 検索するような使い方がある。

しかし、小型カメラで撮影された画像は低解像度であり、手持ち撮影による傾き、レンズの歪み、ピントのずれ、などの様々な理由から画質の劣化が生じるため、スキャナ画像に比べて正確な文字認識は困難である。

低品質、低解像度画像向けの文字認識技術として、前処理や特徴抽出などを工夫して認識率を上げる手法が幾つか提案されている^{[3][4][5]}。また、文書全体を一度に撮影するのではなく、領域分割して拡大撮影した画像を個別に処理して、後で結合する方法も知られている^[6]。しかし、これらの手法は撮影と認識の処理を個別に行うため、認識後に誤読文字が見つかった場合は、キーボードなどを用いて修正する必要があった。

筆者らは、従来手法のように撮影した画像を後で時間をかけて文字認識するのではなく、携帯端末上でリアルタイムに処理を行いながら、認識結果をユーザーにフィードバックするインタラクティブな UI を提案する。

本稿では、筆者らが提案する UI 手法の有効性を検証するために開発したデモシステムの詳細と、その性能評価を行った結果について報告する。

2 提案 UI

携帯端末などに文字を入力する際には、URL、メールアドレスなどを入力する場面が多いと考えられる。よって本稿では 10~40 文字程度の英数字列を入力することを想定する。

この程度の長さの文字列であっても、小型カメラでは全体を一度に撮影して認識処理を行うには解像度が不足する。本手法では撮影と認識を繰り返しながら分割撮影することで長い文字列を入力する。1回の撮影ごとに認識結果が表示されるので、ユーザーはその結果を見ながら次の撮影の位置を決めることができる。また、分割撮影の流れのままシームレスに誤認識の修正が行える。

本章では、まず提案 UI 手法をユーザーの操作の観点で説明し、次にシステムの処理を説明する。

2.1 操作方法

図 1 に、提案手法の表示画面を示す。システムは「プレビュー画面」と「確認画面」の二つの画面モードを持っていて、撮影ボタンを押す度にこのモードが切り替わる。

ユーザーは、まず図 1(a) プレビュー画面において、カメラ映像を確認しながら、入力したい文字列の先頭が画面中央の基準線にかかるようにして、撮影ボタンを押す。すると認識結果が図 1(b) 確認モードのように表示されるので、印刷文字と、その下に表示された認識結果を画面上で確認する。確認したら再び撮影ボタンを押してプレビュー画面に戻り、2 回目の撮影に移る。

認識結果に問題がなければ、ユーザーはカメラを右に移動しながら残りの文字列を撮影していく。撮影の際には、以前の撮影結果と撮影文字列の一部が重複するように撮る必要がある。毎回の認識結果は、自動で結合されて画面の下に表示される。

確認の際に誤認識を見つけた場合、ユーザーは

A proposal of character input user interface through a small camera.

Kyosuke NISUYAMA, Shuji SENDA.

Internet Systems Research Laboratories, NEC Corp.

その文字を撮り直すことで修正できる(図2)。連結文字列の表示部分において、撮り直しによって修正された部分は色が変わるので、ユーザーは簡単に変化した部分を確認できる。重複して撮影された箇所は常に修正処理が働くため、画面右の方で起きた誤認識は修正を意識しなくても、次の撮影で自動修正される(図3)。

非常に似通った類似文字や特殊フォントなど、撮り直しても修正できないような誤認識があった場合には、矢印キーを持ちいて修正を行う。連結文字列表示部分には下線で表示されたカーソルがあり、左右キーで文字を選び、上下キーを押すことで認識の次候補を選ぶことができる。また、カーソルは常に前回画面中心で撮影した文字に自動的に合う様になっているため、①入力ミスがあったので、その部分を中心にして撮影しなおす、②まだ直らないので矢印キーの下を押して次候補を出す、という少ないキー操作で修正できる。

次に、以上のU Iを実現するシステム側の処理の流れについて説明する

2.2 システムの処理

プレビュー画面モードでは画面の上部表示領域にカメラが撮影する映像を表示する。また画面の中心に基準線を重ねて描画する。ユーザーによって撮影ボタンが押されると、画像を撮影して二値化処理と行抽出処理を行う。抽出のうち最も画面中心に近い行に関して、さらに文字抽出、認識を行う。

認識が終了すると、撮影画像と文字切り出しの結果を矩形で描画し、さらに画像に認識結果の第1候補を描画する(確認画面モード)。もう一度撮影キーが押されると、再びプレビューモードに戻る。

認識結果は、撮影の度に今までの認識結果と連結される。連結の際には、まず二つの認識結果に対して認識候補情報を元にマッチングを行い、重複部分を検出する。重複部分に関して、文字ごとに信頼性を比べ、より信頼性の高いデータの方を記憶する。ここで認識結果の信頼性の指針としては、

1. 認識スコア

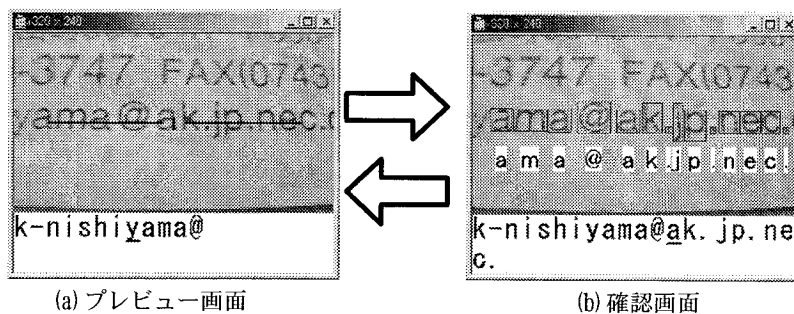


図1 提案U Iの画面表示

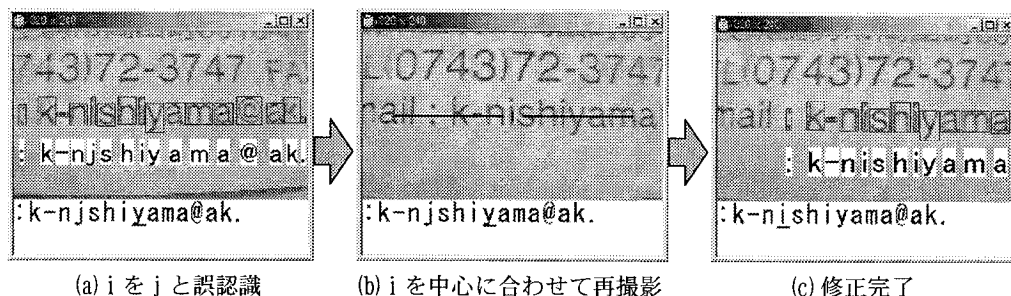


図2 再撮影による修正



図3 結合時の自動修正

2. 画面中心からの距離

を用いた。1. は文字認識の結果得られる値であり、値が小さいほど認識辞書パターンと類似していることを意味する。2. は、小型カメラで撮影された画像は画面端に近くなるほど歪みが激しくなり、ピントもボケる傾向があることから、画面中心からの距離が近いほど認識結果が信頼できる。

以上のようにして連結された文字列は、モードに関わらず画面の下に常に表示する。このとき、①新しい撮影によって追加された文字、②新しい撮影により結果が変化した文字、はそれぞれ色を変えて表示する。

3 評価実験

提案UIの有効性を確かめるため、デモシステムを開発し、評価実験を行った。

3.1 実験環境

実験用のシステムはWindows PC上で開発した。入力装置としては解像度 320x240 のUSB接続カメラを用い、ピントは約 2 cm の幅が撮影できる距離で固定した。文字を正確に入力するUIとしての評価を行うため、入力時間を計測する際には、入力開始から全て文字列が正しく入力できたと確認し終わるまでの時間を計測した。

3.2 実験 1

まず、名刺のメールアドレス部分を入力する実験を行った。撮影サンプルとしては、個人の所有する 150 枚の名刺から無作為に 50 枚を選んで使用した。

入力は本システムに熟練した被験者 1 名が行った。提案手法は、入力途中にいつでも前に戻って修正できるが、ここでは

① まず左から右に分割撮影してメールアドレス全体を入力し、

② その後、誤認識があればそこを再撮影などで修正する。

という流れの入力に統一した。

実験の結果、50 枚の名刺のうち 48 枚を正確に入力することができた。残りの 2 枚はメールアドレス最後の「jp」表記において、[j]と[.]が分離できない特殊なケースであり、今後の文字抽出手法の改良によって対応できる。

また、正しく入力できた 48 枚の中には、印刷が掠れているか薄いため、二値化後の文字ストロークが激しく分断されてしまうものが 3 枚、フォント同士が接近しているため文字切り出しが非常に困難なものが 5 枚存在した。これらの名刺は最終的に入力できたとはいえ、ピントを故意にずらして撮影するなどの一般的ではない技巧を用いる必要があったため、今後の考察ではリジェクト扱いとする。

認識不能 2 枚、リジェクト 8 枚を除いた 40 枚に関して、重複撮影による自動修正が働いていない最初の撮影における文字認識率は 498/510 文字 (97.7%) であった。さらにメールアドレスを右まで撮影し終えた後の認識率は 961/977 文字 (98.4%) であり、結合部分の自動修正によって、誤読率が 2.3% から 1.6% に減少している。

修正も含めた入力時間を入力文字数で割った 1 文字あたりの入力時間は、平均約 435msec であった。名刺ごとの入力時間の分布を図 4 に示す。グラフの網掛け部分は誤認識の修正に要した時間、黒塗り部分は、重複文字数が足りないなどの理由で認識結果が記録された文字列に連結できなかったことによるタイムロスを示している。

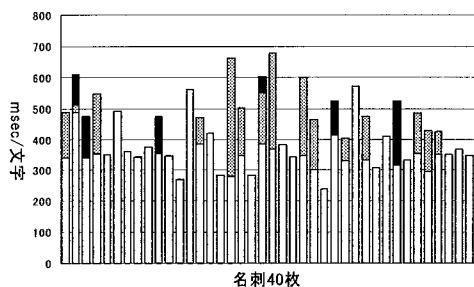


図4 入力時間と修正時間の割合

連結の失敗は、主に重複部分の文字数が連結に必要な3文字を満たしていないことにより起こる。これは言わば被験者の操作ミスであるが、重複部分の位置や長さを意識せずに入力できることを目指しており、今後のUIの改良が必要だと思われる。

誤認識の修正に要した総時間を連結文字列の誤認識数で割ると、誤認識1字あたりの修正時間は約3850msecであることがわかる。

分割撮影入力の誤認識率は1.64%であるから、 $3850\text{msec} \times 1.64 \div 100 = 63.1\text{msec}$

がこのシステムの、1文字あたりの誤認識修正時間である。これは入力全体に要した時間(435msec)の約15%であり、時間のかからない簡単な修正UIで、正確な入力を実現できたと考える。

3.3 実験2

次に他の入力手法との比較実験を行った。実験1とは別の名刺5枚に関して次の5手法で入力を行い、要した時間を比較する。

- ・提案手法
- ・ノートPCのキーボード
- ・携帯電話に接続して用いる小型のキーボード
- ・携帯端末上に表示され、スタイラスでタップして入力するソフトキーボード

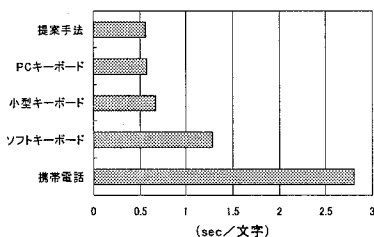


図5 他手法との文字入力速度比較

- ・携帯電話テンキーによるマルチタップ入力。

5枚の名刺の平均値は図5のようになった。図から、カメラ入力はPCのキーボードとほぼ同じ程度の速度で入力が可能であり、他の何れの携帯端末向け入力手法よりも速く入力できることが確かめられた。

4 まとめ

分割撮りと撮り直しによる認識修正、カーソルによる次候補選択などを取り入れた小型カメラによる文字入力UIを提案した。

50枚の名刺から認識困難な10枚を除いた40枚に関して認識実験を行い、文字認識処理に分割撮影による自動修正処理を組み合わせることによって認識率が向上することを確かめた。また、撮影による修正UIによって、短時間で全てのメールアドレスを正しく入力することができた。

他の端末向け文字入力手法と比較実験を行い、入力速度の点で優れていることを確認した。

今後はリジェクトした認識困難な対象の対処方や、修正UIの操作性などを改善していきたい。

また、今回は提案手法の有効性評価と言うことで一人の被験者で実験を行ったが、UIとしてのユーザビリティを調べる為の複数被験者による評価実験も行いたいと考えている。

[参考文献]

- [1] 沢木、村瀬、萩田：“変型推定に基づいた参照パターンの自動選択による文字認識” 信学技報, PRMU97-187, pp. 15-22.
- [2] 高橋、葛西、中嶋：“複合特徴を用いた情景画像からの文字領域抽出” 信学技報, HCS2000-30, pp. 47-54.
- [3] 村形、恩田、小沢：“多値文書画像における適応的二値化手法” 信学技報, PRMU99-232, pp. 9-16.
- [4] 小佐井、星野、岡本、加藤、山本：“低解像度画像からの文字認識手法について”, 信学技報, PRMU97-221, pp. 33-38.
- [5] H. Nishida: “A note on practical uses of gray-scale image analysis in document recognition”, Pattern Recognition Letters, 19, pp. 889-897.
- [6] N. Nakajima, K. Yamada: “High Resolution Document Image Acquisition with Active Camera”, Document Analysis Systems, 2000, pp. 65-76.