文字形状に基づく認識駆動型ビデオモザイキング手法

宮崎　洋光†　　内田　誠一††　　迫江　博昭††

† 九州大学大学院システム情報科学府
†† 九州大学大学院システム情報科学府研究院
〒 812–8581 福岡市東区箱崎 6–10–1
E-mail: {hiromitu,uchida,sakoe}@human.is.kyushu-u.ac.jp

あらまし　本論文では，動画像を用いたテキスト認識を目的として，手ぶれ変形の補償，複数フレームの統合 (モザイキング)，ならびにテキスト認識を一括して実行する手法を提案する．静止画像ではなく動画像を用いることにより，テキストの長さの制約を排除できると考えられる．一方，解決すべき点としては，撮影中の手ぶれによる文字の変形および複数フレームへの分断が考えられる．本手法は様々な変形をパラメータ化し，それを segmentation-by-recognition 的な手法で同時最適化することで，最適変形補償ならびに最適フレーム統合を可能にしている．予備的な実験の結果，様々な手ぶれを起こした場合でも 90% 前後の文字認識率が得られ，本手法の有効性が確認できた．
キーワード　テキスト認識，文字認識，ビデオモザイキング，歪み

Video Mosaicing for Camera-Based Text Recognition

Hiromitsu MIYAZAKI†, Seiishi UCHIDA††, and Hiroaki SAKOE††

† Graduate School of Information Science and Electrical Engineering, Kyushu University
†† Faculty of Information Science and Electrical Engineering, Kyushu University
Hakozaki 6–10–1, Higasi-ku, Fukuoka-shi, 812–8581 Japan
E-mail: {hiromitu,uchida,sakoe}@human.is.kyushu-u.ac.jp

Abstract　In this paper, a mosaicing-by-recognition technique is proposed, where video mosaicing and text recognition are simultaneously and collaboratively optimized in a one-step manner. Specifically, multiple frames where a long text line is captured while moving a camera are optimally matched and concatenated with a guide of the text recognition framework. The optimization is performed by a DP-based algorithm and can compensates rotation, scaling, and speed fluctuation which appear in texts captured by hand-held cameras. The results of an experiment to evaluate not only the accuracy of mosaicing but also that of text recogntion indicates that the proposed technique is very practical and can provide reasonable results in most cases.
Key words　text recognition, character recognition, video mosaicing, distortion

1. はじめに

カメラによる画像中のテキスト認識，理解において，動画像の利用が検討されている [1], [2]．動画像を用いることの利点として，たとえば，静止画像では画角の制約により撮影できない長いテキストであっても撮影できることが挙げられる．また，隣接フレーム間のオーバーラップを活用することにより，解像度および 2 値化精度の向上が期待できる．

動画像を用いたテキスト認識においては，カメラの移動により変動が生じたフレームを何らかの方法で統合 (いわゆるモザイキング処理) する必要がある．従来，このフレーム間統合処理とテキスト認識処理は独立して扱われていた．具体的には，連続したフレームを統合しモザイク画像を生成 (たとえば [3]) した後に，文字・単語認識を行うことを想定している．このように 2 つの処理をの直列的に行うと，フレーム統合処理の段階で精度の高い結果を得ることができなかった場合 (通常この処理は難しい)，後の認識の段階に大きな影響を与え，誤認識を生じる原因となる．また，他の従来手法として，各フレーム内でテキスト認識を行った後，認識結果を統合する手法 [4] もあるが，これも 2 段階的手法となっている．

本論文では，フレーム統合処理とテキスト認識処理の 2 つの処理を一括して行う手法—mosaicing-by-recognition—を提案する．本手法では，移動するカメラにより，複数フレーム画像中に断片的に撮影されたテキストを対象とする．各フレームに

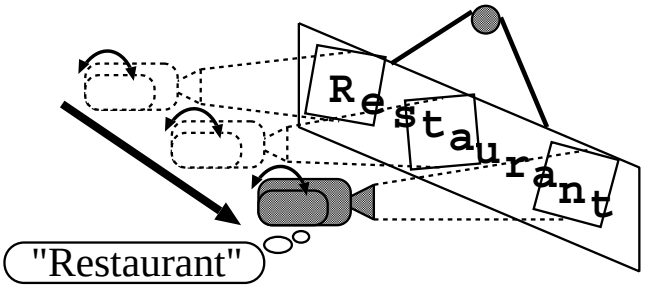

図 1　動画像を用いたテキスト認識

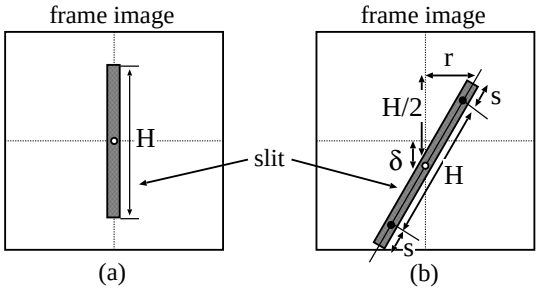

図 2　(a) 基準スリット ($r = s = \delta = 0$)，および(b) 形状制御されたスリット

生じる回転，拡大縮小，上下移動，カメラ移動速度の変動などの一般的な手ぶれ変動を補償しながら，各フレーム画像と標準文字パターンとを連続的にマッチングし，テキスト認識結果およびフレーム統合結果を同時に最適化する．本手法では，従来手法のように隣接フレーム間で直接フレーム統合するのではなく，標準文字パターンを介してフレーム統合を行うことになる．さらに，マッチングの最適化は DP アルゴリズムにより効率的に実行できる．

以下本論文では，まずカメラ移動速度の変動のみを考慮した単純な場合を説明し，次に，回転，拡大縮小，上下移動など一般的な手ぶれを考慮した場合について説明する．単純な場合のアルゴリズムは，従来の水平方向に非線形伸縮したテキストに対する segmentation-by-recognition (analytic approach [5] とも呼ばれる) と同様，一種のフレーム同期 DP アルゴリズム [6] に帰着できる．一般的な手ぶれを考慮した場合のアルゴリズムは，この単純な場合の拡張として与えられる．また，マッチング精度向上のための試みおよび高速なカメラ移動のための工夫についても考察する．

2. Mosaicing-by-recognition

2.1 処理対象とする動画像および手ぶれ変動

本手法では，手持ちカメラを左から右へと移動させ，テキストを撮影したビデオフレーム列を対象とする．ただし撮影の際，1 文字が複数フレームに含まれるように仮定する．従って，フレームレートが高いビデオカメラ (もしくはカメラをゆっくり移動させること) を仮定することになる．

以上の仮定の下，撮影されたビデオフレーム列について，各フレームの中心より幅 W 画素，高さ H 画素の矩形領域 (以下，スリット) を考える．当面の間，スリット幅 W を 1(図 2 (a)) とする．幅 1 のスリット用いることで解くべき問題が単純化され，本手法の基本的な特性を把握することが容易になる．ただし後の実験から明らかになるように，幅 1 のスリットでは，認識ならびにモザイキング性能に限界がある．また，フレーム内の大部分の画像情報を無視している点も問題である．そこで 2.6.1 節では，より幅の広いスリットの利用について検討する．

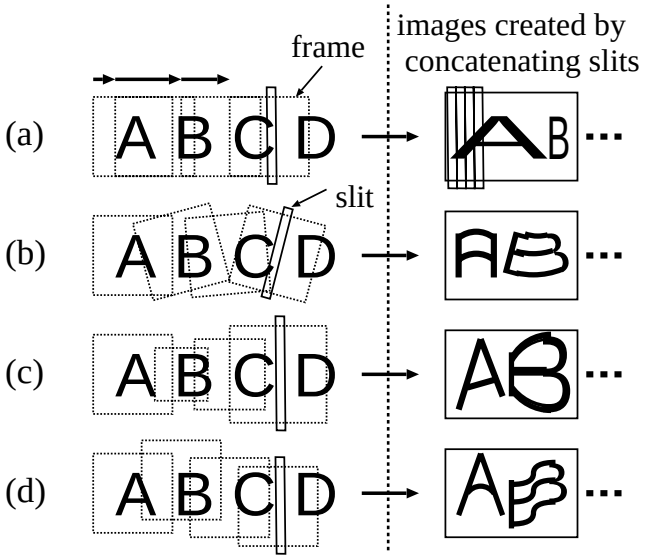

図 3　手持ちカメラにより撮影されたフレームに現れる主な変形：(a) カメラ移動速度の変動，(b) 回転，(c) 拡大縮小，(d) 上下移動

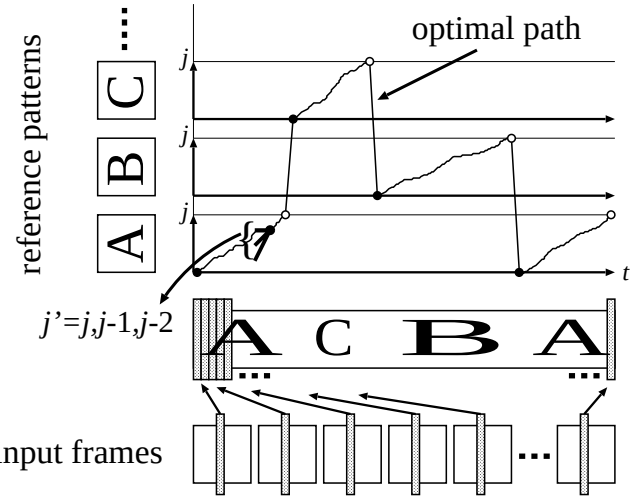

図 4　カメラ移動速度の変動のみを考慮した mosaicing-by-recognition

フレーム画像系列に現れる変動については，カメラ移動速度の変動と，回転，拡大縮小 (カメラの紙面との距離の変動)、上下移動 (画像内における文字の上下移動) などを一般的な手ぶれとして想定している．ビデオフレームに現れる変動の例を図 3 の左半分に，またそれらのフレームのスリットを並べたものを同図の右半分に示す．

2.2 カメラ移動速度の変動のみを考慮した場合の DP アルゴリズム

本節ではカメラ移動速度の変動のみを考慮した場合の手法について説明する．入力画像として全 T フレーム分のスリット ($W = 1$) を並べて生成した画像 (図 3 (a) の右側の画像)

/* Initialization */
1 **for** $c := 1$ **to** C **do begin**
2 $g_1(c,1) := d_1(c,1)$
3 **for** $j := 2$ **to** J_c **do**
4 $g_1(c,j) := \infty$
5 **end**
6 $D_1 := \infty$
/* DP Recursion */
7 **for** $t := 2$ **to** T **do begin**
8 **for** $c := 1$ **to** C **do begin**
9 $g_t(c,1) := d_t(c,1) + \min\{g_{t-1}(c,1), D_{t-1}\}$
10 **for** $j := 2$ **to** J_c **do**
11 $g_t(c,j) := d_t(c,j) + \min_{j' \in \{j, j-1, j-2\}} g_{t-1}(c,j')$
12 **end**
13 $D_t := \min_{c' \in C} g_t(c', J_{c'})$
14 **end**

図 5 カメラ移動速度の変動のみを考慮した DP アルゴリズム

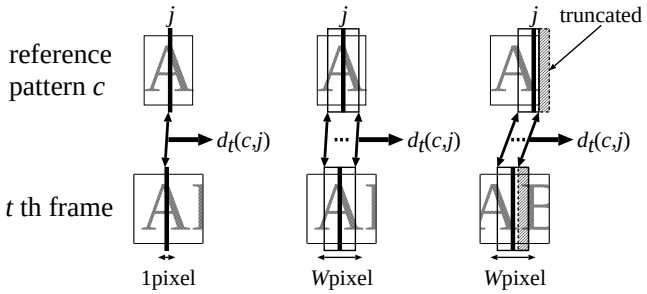

図 6 局所距離計算の対象となる領域：(a) スリット幅 1 画素の場合，(b) スリット幅 W 画素の場合，(c) スリット幅 W 画素で標準パターンの領域を超える場合

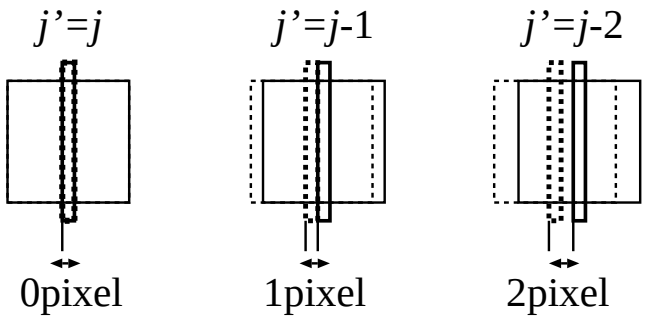

図 7 補償されるカメラ移動速度変動と j' の関係

を考えると，カメラ移動速度の変動によりこの画像内の各文字は水平方向に非線形伸縮されると考えられる．従ってこの場合のテキスト認識問題は，この入力画像 (横幅 T 画素) と全 C カテゴリ分の標準文字パターン (横幅 J_C 画素) 間での最適なマッチング問題に帰着する．すなわち，各文字の水平方向の非線形伸縮を補正しながら，同時に文字間境界を決定するという，segmentation-by-recognition [5], [6] と同様の問題となる．よく知られているように，この形式の問題は，図 4 に示すように入力画像を横軸に，各文字標準パターンを縦軸にとった平面中の最適経路問題として表現され，その最適化は DP アルゴリズムを用いて効率的に求めることが可能である．

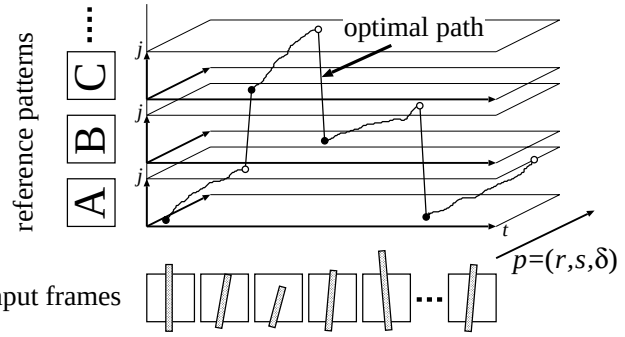

図 8 一般的な手ぶれを考慮した場合の mosaicing-by-recognition

図 5 にその DP アルゴリズムを示す．ステップ 9，11 の $d_t(c,j)$ は第 t フレーム目のスリット (入力画像の第 t 列目) と標準文字パターン c の第 j 列目とのマッチング距離である．本手法では単純な輝度値を画素特徴とした H 画素分の単純重ね合わせ距離を用いた (図 6 (a))．また，DP 漸化式処理により $g_t(c,j)$ には第 t フレーム目のスリットが標準文字パターン c の第 j 列目に対応されるまでの，初期フレーム ($t = 1$) からの最小累積距離が格納される．ステップ 13 の D_t は初期フレームから最終フレームまでの最小累積距離である．ステップ 11 が主要となる DP 漸化式であり，入力フレーム t の進行に従い，順次計算される．また，ステップ 9 は文字境界候補を定める処理に相当し，図 4 の白丸で表された点における処理である．

DP 漸化式内での j' に関する最小値選択は，カメラ移動速度の変動を補償することを目的として行われている．すなわち，図 7 に示すように，カメラ移動速度が 2.0 pixel/frame の場合にはスリットを飛ばすために $j' = j - 2$ が選択され，0 pixel/frame の場合にはスリットを重複させるために $j' = j$ が選択されると考えられる．本手法では，j' を $\{j,\ j-1,\ j-2\}$(図 4 の 3 本の DP パス) から選択するために，補償できるカメラ移動速度変動は 0 ～ 2.0 pixel/frame である．カメラの移動をより高速に行う工夫については，2.6.2 節で詳しく考察する．

図 5 では省略したが，実際には，上述した DP アルゴリズムの終了後，最短経路 (D_t を得るために選択した経路) をバックトラックすることにより文字列認識結果を求める必要がある．具体的には，バックトラックの際に通過する文字境界候補を用いて，文字境界の決定と認識の結果を得ることができる．このバックトラックの結果からモザイク画像も得られるが，この処理については一般的な手ぶれを含む場合と共に 2.5 節で述べる．

2.3 一般的な手ぶれを考慮した場合の DP アルゴリズム

本節ではカメラ移動速度の変動に加え、一般的な手ぶれ (回転，拡大縮小，上下移動) を考慮した場合の手法について説明する．一般的な手ぶれによる変動に対処するために導入した前節のアルゴリズムからの変更点は，スリットの形状をフレームに生じた変動に合わせて制御することである．スリット形状の制御が最適に行われた場合 (つまりフレームの変動に合わせたスリットの制御が行われた場合)，2.2 節で述べた最適マッチング問題と同様に考えることができる．しかし，実際には，スリッ

/* Initialization */
1 **for all** $p \in \{(r,s,\delta)\}$ **do begin**
2 **for** $c := 1$ **to** C **do begin**
3 $g_1(p,c,1) := d_1(p,c,1)$
4 **for** $j := 2$ **to** J_c **do**
5 $g_1(p,c,j) := \infty$
6 **end**
7 $D_1(p) := \infty$
8 **end**
/* DP Recursion */
9 **for** $t := 2$ **to** T **do begin**
10 **for all** $p \in \{(r,s,\delta)\}$ **do begin**
11 **for** $c := 1$ **to** C **do begin**
12 $g_t(p,c,1) := d_t(p,c,1) + \min_{p' \in \mathrm{pre}(p)} \{g_{t-1}(p',c,1), D_{t-1}(p')\}$
13 **for** $j := 2$ **to** J_c **do**
14 $g_t(p,c,j) := d_t(p,c,j) + \min_{\substack{p' \in \mathrm{pre}(p) \\ j' \in \{j,j-1,j-2\}}} g_{t-1}(p',c,j')$
15 **end**
16 $D_t := \min_{c' \in C} g_t(p,c',J_{c'})$
17 **end**
18 **end**

図 9 一般的な手ぶれを考慮した場合の DP アルゴリズム

ト形状の最適な制御は未知であるので，各フレームのスリットについて全ての形状制御を試みながら最適マッチングを行う必要がある．

図 8 に手法の概要を示す．各フレームのスリットについては，その形状を図 2 (b) に示すように回転を r，拡大縮小を s，上下移動を δ で表した制御ベクトル $p = (r,s,\delta)$ により制御する．具体的なスリット形状の制御は制御ベクトル p を用いて以下のように処理を行う．なお，以下で述べる基準スリットとは前節で用いたスリット (図 2 (a)) である．すなわち，形状制御を行わない場合 ($r = s = \delta = 0$) のスリットを指す．

- 回転：基準スリットの上下端の画素位置から水平方向に上端画素を r 画素，下端画素を $-r$ 画素ずらす．
- 拡大縮小：基準スリットの上下端の画素位置から垂直方向に上端画素を $-s$ 画素，下端画素を s 画素ずらす．
- 上下移動：基準スリットを垂直方向に δ 画素ずらす．

こうして形状制御されたスリット内の部分画像と標準文字パターンの各列との最適なマッチングを行う．r，s，δ の各々についての形状制御を試みるので，同図の探索空間は，前節図 4 の探索空間に新たに $p = (r,s,\delta)$ 分の 3 次元が加わったものとなる．

図 9 に一般的な手ぶれを考慮した DP アルゴリズムを示す．2.2 節のアルゴリズムと若干異なり，マッチング距離は $d_t(p,c,j)$，マッチング距離の総和は $g_t(p,c,j)$ と表現される．ステップ 14 では，隣接フレームの変動は互いに類似し連続的であるという仮定の下，以下の制約式を満たす $\mathrm{pre}(p)$ より最小値を選択している．

$$\mathrm{pre}(p) = \{(r',s',\delta') \mid r-1 \leqq r' \leqq r+1\ ,\ s-1 \leqq s' \leqq s+1\ ,\ \delta-1 \leqq \delta' \leqq \delta+1\}$$

2.2 節と同様に，得られた最小累積距離を与える最短経路をバックトラックすることにより文字列認識結果を行う．

2.4 計 算 量

カメラ移動速度の変動のみの場合の計算量は，入力フレーム数 T，標準文字パターン数（カテゴリ数）C，標準文字パターンの横幅 J とすると，$O(TCJ)$ となる．一般的な手ぶれを含む場合の計算量は，変動 r,s,δ の範囲をそれぞれ R,S,Δ で表すと $O(TCJRS\Delta)$ となる．

2.5 モザイク画像の生成

まず，カメラ移動速度の変動のみを考慮した場合のフレーム統合処理について述べる．2.2 節で述べたようにスリットを並べた画像内の文字は非線形伸縮されている．よって，フレーム統合処理は各スリット間の水平位置を文字の非線形伸縮を無くす方向にずらすことにより行うことができる．この水平位置のずらし量はカメラ移動速度から定まり，従って図 7 に示したように，DP 漸化式計算の際に選択した j' の値から決定される．よって，バックトラックにより順次 j' を求めれば，図 7 の規則に従いフレーム画像を重ね合わせることでフレーム統合が可能となる．なお，$j = j'$ となりスリットが重なった場合には，重複したスリットの部分画像の平均値を用いてモザイク画像を生成する．

次に，一般的な手ぶれを考慮した場合について述べる．この場合のフレーム統合処理は，各フレームについて，最適化されたスリット形状の制御パラメータ $p = (r,s,\delta)$ と水平方向のずらし量 j' を用いることにより行う．すなわち，バックトラックにより各フレームの最適な (r,s,δ) の値を求め，それを用いて各フレームの変動を補償すれば，後はカメラ移動速度の変動のみの場合と同様に考えることで，フレーム統合を行うことが可能となる．

以上のフレーム統合処理は，2.2 節および 2.3 節で述べた最適マッチングの結果を利用して行う．すなわち，従来手法のように別途各フレーム間での対応点を探索しておく必要はなく，標準文字パターンを介してフレーム統合を行う．これが本手法を mosaicing-by-recognition と呼ぶ理由となっている．

2.6 スリット幅およびカメラ移動速度の考察

2.6.1 幅を広げたスリットの利用

以上の節では，スリット幅 W を 1 画素として説明を行った．このため，前述したように解くべき問題が単純化でき，本手法の特性を容易に把握することができた．しかし問題点として，マッチング最適化やモザイク画像生成の際に，フレーム内に含まれる大部分の画像情報を利用していないためにマッチング精度が低下することが考えられる．加えて，次節で述べるようにより高速なカメラ移動を考える場合に関しても，幅 1 のスリットの利用では問題が生じる．

そこで本節では，マッチングの精度向上のために，スリット幅を W 画素に広げてフレーム内の画像情報を有効に活用することを考える．具体的には，マッチング距離 $d_t(c,j)$ を，標準

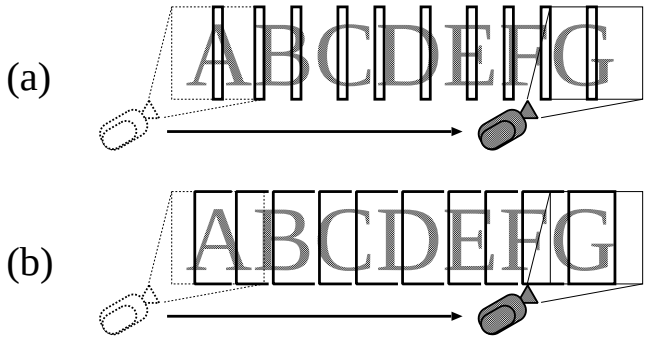

図 10　スリット幅およびカメラ移動速度と利用可能な画像情報の関係

表 1　実験に用いたテキストデータ

character category	alphabet(capital,small),digit
font	Times-Roman
character height H	40pixel
# character/text	~ 50
# text	20

文字パターン c の j 列を中心とした幅 W の領域と幅 W のスリットの部分画像とのマッチングにより求める (図 6 (b))．ただしこの際，スリットの幅が W 画素であることにより標準文字パターンの範囲を超える領域についてはマッチングを行わない (図 6(c))．このため，スリットに 2 つ以上の文字が含まれる場合でも 1 つの文字に対してのみのマッチング距離の計算を行うことができる．

また，モザイク画像の生成には幅 W 画素のスリットを j に応じてずらしながら貼り合わせていく処理を行う．このように基本的に 2.5 節の処理と同じであるが，幅が W に増えた分，平均化される領域も増え，結果的にモザイク画像の質の向上が期待できる．

2.6.2　より高速なカメラ移動を許容するための工夫

2.2 節で述べたように，本手法では DP 漸化式内での j' の選択により，カメラ移動速度の変動を補償していた．しかし，$\{j,\ j-1,\ j-2\}$ (3 本の DP パス) から選択を行うため，補償可能なカメラ移動速度は 0 ～ 2.0 pixel/frame と制限される．そこで本節では，カメラ移動速度の制約を緩和するために，j' の選択可能範囲の拡張 (DP パスの増加) を考える．具体的には，j' を $\{j-k \mid k=0,\ldots,K\}$ ($K+1$ 本の DP パス) から選択できるように変更する．この拡張により，補償可能なカメラ移動速度の制約を 0 ～ K pixel/frame に緩和できることが考えられる．ここで問題点として，カメラ移動速度が大きい場合，幅 1 画素のスリットでは各フレームより得られる画像情報が少ないことが考えられる．このため，テキストの画像情報が各文字につき数列分しか取得できず，マッチング精度が低下することが考えられる (図 10 (a))．従って，マッチング精度低下を避けるために，テキストの画像情報を全て内包できるように幅を広げたスリットの併用も同時に検討する (図 10 (b))．

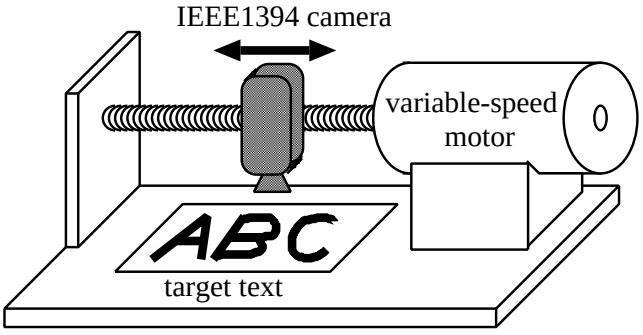

図 11　動画像データの取得装置

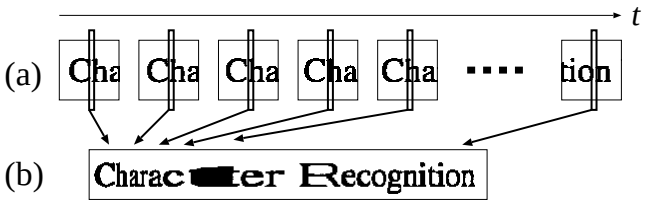

図 12　カメラ移動速度の変動のみを含む，(a) 動画像データの例，(b) スリットをならべて得られた合成画像

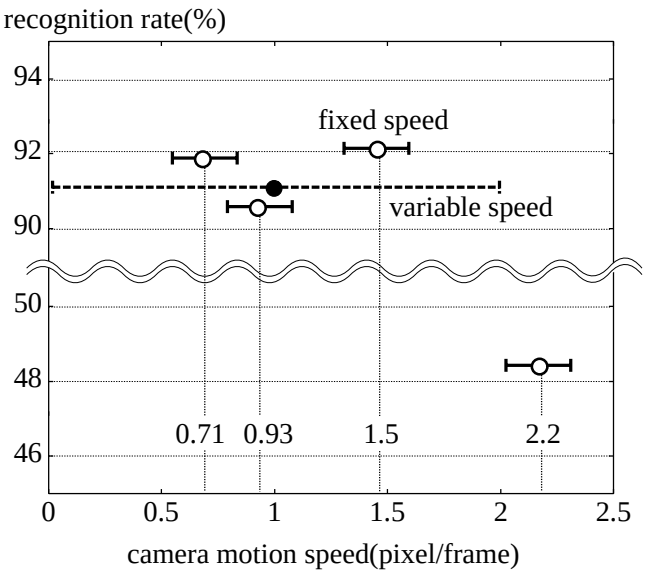

図 13　カメラ移動速度の変動のみの場合の実験結果

3. 実 験 結 果

3.1　カメラ移動速度の変動のみを考慮した場合

3.1.1　実 験 試 料

実験に使用したテキストデータを表 1 に示す．テキストは A4 用紙に印刷し，図 11 の装置を用いて撮影した．この装置は水平方向のカメラ移動速度をつまみで調節できるようになっている．カメラ移動速度を 0.71，0.93，1.5，2.2 pixel/frame で固定した場合と，0 ～ 2.0 pixel/frame の範囲で変動させた場合とを撮影した．

図 12 (a) にカメラ移動速度可変の場合に撮影したテキストを，同図 (b) に各フレームより取得したスリット ($W=1$) を並べた画像の例を示す．カメラ移動速度可変の場合，速度の変

図 14　図 12 のデータについての (a) モザイク画像，(b) 認識結果

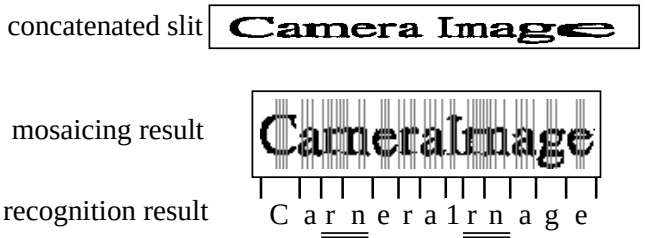

図 15　文字の分離による誤認識例

化は同図 (b) にあるように，文字の水平方向への非線形伸縮として現れる．同図 (b) では，特に「c」と「t」の境界付近では速度が遅くなっているために，文字が左右に大きく伸ばされている．

3.1.2　実 験 結 果

3.1.1 節で述べたカメラ移動速度を固定および可変にした場合それぞれについての認識実験を，2.2 節のアルゴリズムを用いて行った．本節の実験では，本手法の基本的な性能評価のためにスリット幅 $W = 1$ とし，またカメラ速度の可能補償範囲も 2.2 節で述べたように $K = 2$(すなわち最大 2.0pixel/frame) とした．

実験結果を図 13 に示す．同図ではカメラ移動速度の変動範囲を水平線分で表している．このように実験に用いた動画像データには，カメラ移動速度可変の場合だけでなく，カメラ移動速度固定とした場合でも，装置のすべりや摩擦により多少の速度の変動が含まれていた．実験結果より，カメラ移動速度が 0.71，0.93，1.5pixel/frame と固定の場合については認識率 90% 前後であった．この認識率はスキャナベースの OCR のそれに比べ低いが，現在は文字の特徴量として単なる輝度値，また識別には単純なマッチング距離を用いていることを考慮すれば，改善の余地は大いに残されていると言ってよい．ところでカメラ移動速度が 2.0pixel/frame 以上になると，極端に認識率は下がっている．これは，本手法で補償可能なカメラ移動速度が 0 ～ 2.0pixel/frame の範囲であり，それを超えた部分があるとカメラ移動速度変動を補償できないためである．

一方，カメラ移動速度が可変の場合，認識率は 90% 前後であった．このように速度固定の場合と比べても同程度の性能が得られており，本手法の速度変動補償能力の高さが示されている．

図 14 に，図 12 のデータについての認識結果と生成されたモザイク画像を示す．図 14 より，非線形伸縮されていたテキストが正しく認識およびフレーム統合されている様子がわかる．ここでモザイク画像内の灰色の線は，幅 1 のスリットが速度変動補償で 2 画素ずらされたために生じた画像情報のない領域である．この空隙については，認識に影響を与えるものではない

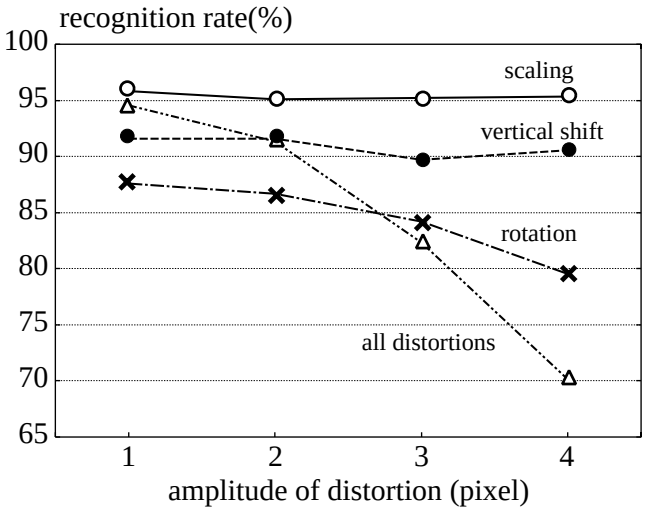

図 16　回転，拡大縮小，上下移動のいずれかを含む場合および全ての変動を含む場合の実験結果

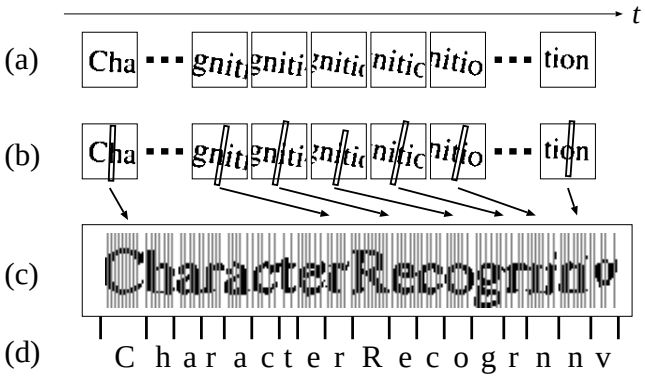

図 17　より一般的な手ぶれ変動を含む場合の (a) フレーム系列，(b) 最適形状制御されたスリット，(c) モザイク画像，(d) 認識結果

が，モザイク画像を重要視する場合は，後述するように幅の広いスリットを用いることで解消できる．

誤認識には，図 15 に示すような文字の分離が多くがみられた．このような誤認識は segmentation-by-recognition 型の認識手法一般に見られる本質的な問題であり，単語辞書などの利用により低減できると考えられる．また，2.6.1 節で述べたように，幅を広げたスリットを用い，フレーム内の画像情報を有効利用することによる低減も考えられる．

3.2　一般的な手ぶれを考慮した場合

3.2.1　実 験 試 料

図 11 の装置では回転などの制御まではできないので，3.1.1 節で取得した動画像データ (速度可変としたもの) を人工的に変形させ，回転，拡大縮小，上下移動を受けたデータを生成した．ただしデータ生成の際には，回転，拡大縮小，上下移動が，前のフレームとの連続性を満たすように生成した．なお，フレーム内に生じる変動 (r, s, δ) は最大で $\pm 1 \sim \pm 4$ 画素とした．

3.2.2　実 験 結 果

3.2.1 節で述べた画像の認識実験を，2.3 節のアルゴリズムを用いて行った．一般的な手ぶれ変動に対する本手法の基本性能の評価のため，3.1 節と同様に $W = 1$，$K = 2$ とした．図 16

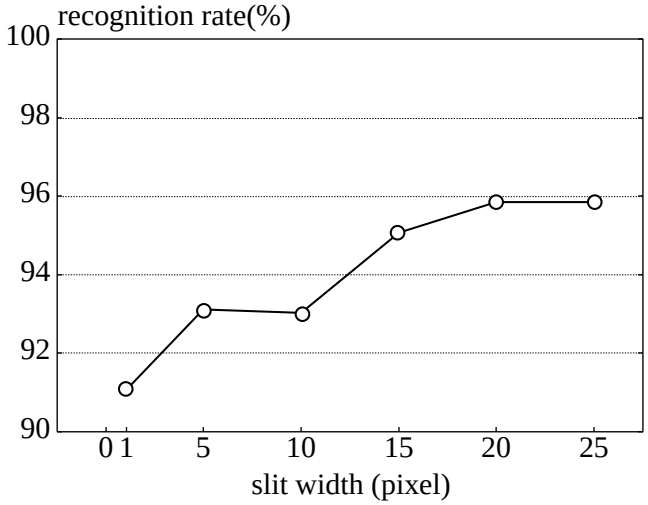

図 18　スリット幅の増加に伴う認識精度の影響

に，回転，拡大縮小，上下移動の変動を個別に施した場合および全ての変動を含む場合についての実験結果を示す．拡大縮小，上下移動の変動については，変動が増大しても認識率への影響は少なく，適切に補償されていると考えられる．ただし回転については，他に比べて変動が大きくなるにつれ認識率が低下している．原因としては，スリットの変形を補償する際 (すなわち図 2 (b) を図 2 (a) の形状に戻す際) に行う線形補間により，画素位置に微少なずれが生じたことが考えられる．撮影した文字の解像度が低いことに加え，マッチング距離 $d_t(p, c, j)$ が単純であるために，微少なずれ (特に水平ストロークの上下移動) であってもマッチング距離の精度に大きな影響を与えると考えられる．また全ての変動を含む場合においては，変動が増大するにつれ認識率が低下していることがわかる．原因としては，個別に変動を施した場合にも見られたように，回転の変動が悪影響を及ぼしていると考えられる．

図 17 に，図 12 に人工的に手ぶれを加えたデータついての認識結果と生成されたモザイク画像を示す．前半部分のテキストについては認識およびフレーム統合が精度良く行われている．しかし後半部分は，誤認識および文字がつぶれたようにフレーム統合されている．これは前述したように回転変動の補償が精度良く行えなかったことが原因であると考えられる．同図 (a) からも，誤認識が生じた文字のフレーム画像には回転変動がみられる．このように，回転の影響で精度良くマッチングが行えなかった結果，文字の分離および結合による誤認識 (同図 (d)) や，精度の低いフレーム統合 (同図 (c)) を行ってしまう．従ってマッチングの精度を向上させるため，一般的な手ぶれ変動を含む場合においてもより洗練された文字の特徴量やマッチング距離の利用およびスリット幅についての検討が必要であると考えられる．

3.3　スリット幅を増加させた場合

まず，3.1.2 節の実験で用いたのカメラ移動速度変動の変動を含む動画像データを用いて，スリット幅の増加に伴う認識精度の考察を行った．ただし，カメラ移動速度の補償許容範囲は $K = 2$ である．図 18 にスリット幅を増加した場合の認識率を

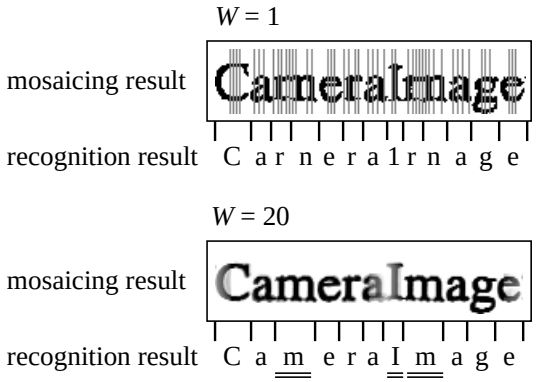

図 19　スリット幅の増加による改善例

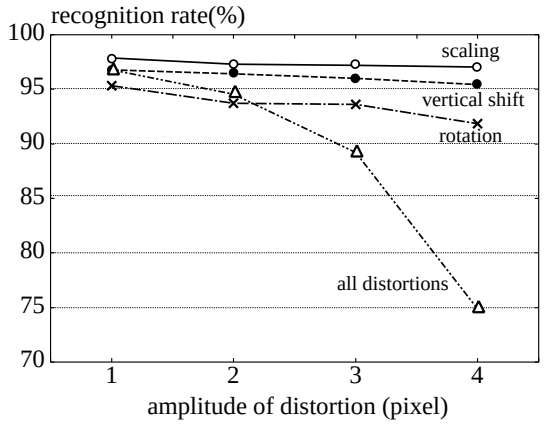

図 20　一般的な手ぶれ変動を含む場合の実験結果 (スリット幅 20 画素)

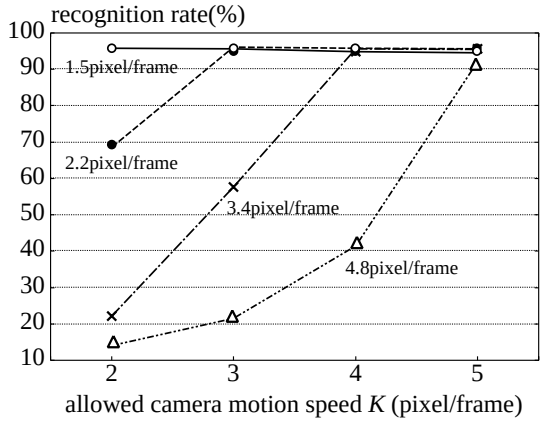

図 21　カメラ移動速度変動の補償可能範囲を変えた場合の実験結果 (スリット幅 20 画素)

示す．同図 から，スリット幅の増加により認識精度の向上が確認できた．これは，スリット幅の増加により，$W = 1$ の場合に比べフレーム内の画像情報を有効に利用できた結果であると考えられる．また，スリット幅 $W = 20$ 程度 (1 つの文字幅の半分程度) で認識率の飽和が見られた．

スリット幅を広げたによる改善例を図 19 に示す．「I」が「1」に認識されるなどの誤認識しやすい文字の改善や，「m」が「r」と「n」に分離されてしまう場合の改善が見られた．同図の結果のように，3.1.2 節の実験ではマッチング精度の低さによる

誤認識されていた文字が，スリット幅を広げることにより正しく認識できるようになった．

次に，3.2.2 節の実験で用いた一般的な手ぶれ変動を含む動画像データを用いて，一般的な手ぶれを考慮した場合のスリット幅の増加よる認識率の影響を観察した．

一般的な手ぶれ変動に対して，スリット幅を 20 画素とした場合の実験結果を図 20 に示す．図 16 のスリット幅を 1 画素とした場合の実験結果と比較すると，個別に変動を加えた場合および全ての変動を加えた場合の動画像データについて認識精度の向上が確認できた．従って，一般的な手ぶれ変動に対してもスリット幅を広げることの有効性が確認できた．

3.4 カメラ移動速度の制約を緩和した場合

3.1.2 節で用いたカメラ移動速度を固定した動画像データを用いて，カメラ移動速度変動の補償可能範囲 K を拡張することによる認識率への影響を観察した．なお，対象は固定カメラ移動速度 1.5，2.2，3.4，4.8pixel/frame (いずれも平均値) で撮影した動画像データとする．また，幅を広げたスリット ($W = 20$) を用いた．

これらの動画像データそれぞれについて，カメラ移動速度変動の補償範囲 $0 \sim K$pixel/frame を変えた場合の認識率の結果を図 21 に示す．結果より，全てのカメラ移動速度について，速度変動が補償可能な場合は高い認識率を得ている．このように補償可能範囲 K の拡張により，3.1.2 節の考察で述べた問題であるカメラ移動速度が大きい場合について本手法が有効性が確認できた．なお，今回の実験では K を大きくしてマッチングの自由度を上げた場合でも，他の文字への合わせすぎを原因とした誤認識はほとんど見られなかった．以上のように，カメラ移動速度を高速に行うための K の拡張および幅を広げたスリットの利用について，その有効性が確認できた．

4. ま と め

本論文では，動画像中のテキスト認識を目的としたビデオモザイキング手法として，手ぶれ変形の補償，複数フレームの統合 (モザイキング)，ならびに認識を一括して実行する手法を提案した．実験結果より，カメラ移動速度の変動のみを対象とした場合は，単純なマッチング距離を用いても認識率は 90% 前後であり，カメラ移動速度が固定の場合と同程度であった．すなわち，速度変動がほぼ補償されており，本手法の有効性が確認できた．ただし,「m」を「r」と「n」に分離する場合があるなど，segmentation-by-recognition 型の認識手法に共通した問題も見られた．一般的な手ぶれを対象とした場合については，回転の影響による認識率の低下がみられたものの，それ以外 (拡大縮小，上下移動) については，現在の単純な実装でも手ぶれなしの場合と同程度の認識率であった．また，スリット幅の増加によるマッチング精度の向上および DP パスの本数の増加によるカメラ移動速度の制約の緩和についても，それぞれ所期の効果が得られることを確認した．

今後の課題としては以下が挙げられる．

（1） マッチング距離の高精度化：今回用いたマッチング距離は，単純にスリットと標準パターン間の輝度値の差を用いた．このため，フレームの変動度合いによっては，精度の良いマッチング距離を求めることができない．よって，マッチング精度向上のためにマッチング距離の高精度化 (特にマルチフレーム特性を活用する手法 [9], [10]) を検討する．

（2） 単語辞書の利用：segmentation-by-recognition 型の認識手法で良く用いられているように，本手法においても辞書の利用が考えられる．これにより「m」が「r」と「n」に誤認識されると言った問題が解決されると考えられる．

（3） 計算量の低減：2.4 節で述べたように，手ぶれによる様々な変動を考慮した結果，計算量は膨大なものとなっている．一般的な手ぶれを考慮した場合において、$W = 1$，$K = 2$，$R = S = \Delta = 1$ の時でも，PC(Pentium Ⅳ，2.8GHz) で 2.79s/character となる．今後，さらに課題 2，3，のように 1 フレームのスリット幅を画像のフレーム幅程度まで増加させた場合，計算量はさらに大きくなると考えられる．具体的な計算量低減手法としては，ビームサーチ法や辞書による制約などが考えられる．

（4） 複数行にわたるテキストへの対処：今回の手法では 1 行のみのテキストを処理対象としているため，複数行にわたるテキストについては考慮していない．今後，さらなる実用化に向け，複数行のテキストへの対処を検討する．

謝辞 本研究の一部は (財) 大川情報通信基金研究助成金ならびに科研費 (若手 (B)，No. 17700198) によった．

文 献

[1] D. Doermann, J. Liang and H. Li, “Progress in Camera-Based Document Image Analysis,” Proc. ICDAR, pp. 606–616, 2003.

[2] 黄瀬 浩一, 大町 真一郎, 内田 誠一, 岩村 雅一, “カメラを用いた文字認識文書画像解析の現状と課題,” 信学技報, PRMU2004-246, March 2005.

[3] A. Zappala, A. Gee, M. Taylor, “Document mosaicing,” Image and Vision Computing, vol. 17, no. 8, pp. 585–595, 1999.

[4] 仙田 修司, 西山 京助, 旭 敏之, “携帯カメラによる日本語文字認識の手法と実現,” 信学技報, PRMU2004-124, Dec. 2004.

[5] R. Plamondon and S. N. Srihari, “On-Line and Off-Line Handwriting Recognition : A Comprehensive Survey,” IEEE Trans. Pat. Anal. Mach. Intell., vol. 22, no. 1, pp. 63–84, Jan. 2000.

[6] 迫江 博昭, 藤井 宏美, 吉田 和久, 亘理 誠夫, “フレーム同期化, ビームサーチ, ベクトル量子化の統合による DP マッチングの高速化,” 電子情報通信学会論文誌, vol. J81-D-Ⅱ, no. 6, pp.1251–1258, June 1988.

[7] 池谷 彰彦, 中島 昇, 佐藤 智和, 池田 聖, 神原 誠之, 横矢 直和, 山田 敬嗣, “カメラパラメータ推定による紙面を対象とした超解像ビデオモザイキング,” 画像の認識・理解シンポジウム MIRU2004, vol. 1 of 2, pp. I-505–I-510, Jul. 2004.

[8] H. Li and D. Doermann, “Text Enhancement in Digital Video Using Multiple Frame Intergration,” Proc. ACM Multimedia, pp. 19–22, 1999.

[9] 柳詰 進介, 目加田 慶人, 井手 一郎, 村瀬 洋, “携帯カメラによる動画像を用いた低解像度文字の認識手法,” 画像の認識・理解シンポジウム MIRU2004, vol. 2 of 2, pp. I-321–I-326, Jul. 2004.

[10] 小佐井 潤, 星野 雄一, 岡本 正義, 加藤 邦人, 山本 和彦, “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, Feb. 1998.