

動画像中のテキスト認識を目的としたビデオモザイク手法

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 本報告では、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、複数フレームの接合 (モザイクング), ならびに認識を一括して実行する手法を提案する. 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接合を可能にしている. 静止画像ではなく動画像を用いることで, テキストの長さの制約を排除できると考えられる. しかし問題点として, 撮影中の手ぶれによる文字の変形, ならびに複数フレームへの分断への対処が考えられる. 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, ビデオモザイクング, 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 compensate 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 recognition 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. ま え が き

カメラによる画像中のテキスト認識, 理解において, 静止画像に加え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で, たとえば, 静止画像では画角の制約で取得できない長いテキストであっても取得できる. また, モザイクングによるテキスト画像の解像度の向上が期待できる.

動画像を用いたテキスト認識の従来手法において, モザイクング処理と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接合しモザイクング画像を生成 (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 つの段階の直列処理を行うと, モザイクングの段階で精

度の高い結果を得ることができなかった場合, 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.

本報告では, ビデオモザイクングとテキスト認識の 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. 文字レベルで生じる回転, 拡大縮小, 上下移動, 走査速度の変動などの変形を補償しながら, 各フレーム画像と標準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モザイクングを同時に

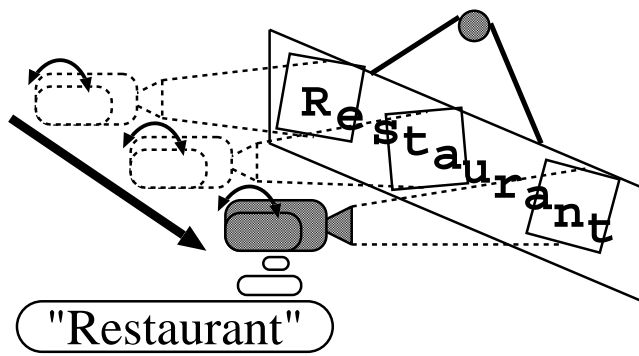


図1 動画像を用いたテキスト認識

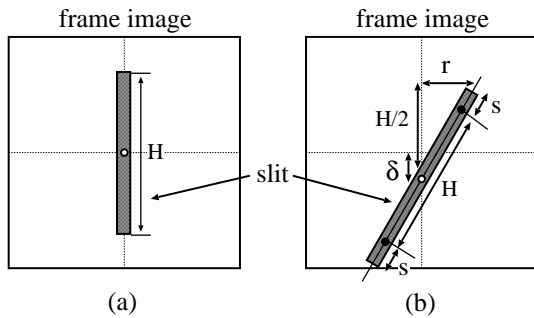


図2 (a) 基準スリット ($r = s = \delta = 0$), および(b) 形状制御されたスリット

最適化する. すなわち本手法では, このマッチングの過程において, 認識結果とモザイクング結果が同時に最適化されることになり, 従来手法のようにモザイクングのために事前にフレーム間での対応点探索を行う必要はない. なお, マッチングの最適化は DP アルゴリズムにより実行される.

以下本報告では, まず走査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 mosaicing-by-recognition 法に関しては, 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] に帰着する. 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.

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横幅 1 画素, 縦幅 H 画素の矩形領域 (以下, スリット) を考える (図 2(a)). スリットは横幅 1 画素であるために, 同時に 2 つ以上の文字を含まない. 後に明らかになるように, このスリットを考えることで問題を単純化できる. このスリットの幅については, 2.6 節で詳しく考察する.

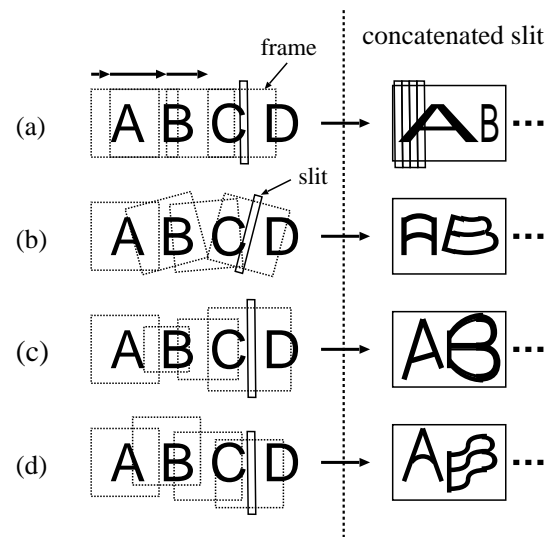


図3 手持ちカメラにより撮影されたフレームに現れる主な変形, (a) 走査速度の変動, (b) 回転, (c) 拡大縮小, (d) 上下移動

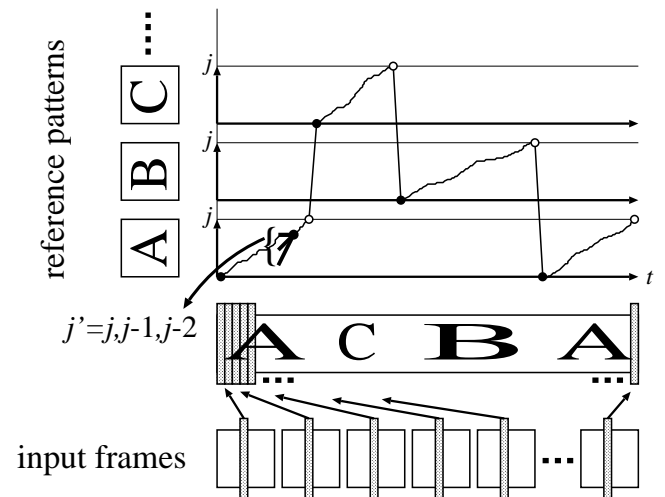


図4 走査速度の変動のみを考慮した mosaicing-by-recognition 法

フレーム画像系列に現れる変動については, カメラの移動による走査速度の変動と, 回転, 拡大縮小, 上下移動 (画像内における文字の上下移動) などの一般的な手ぶれを想定している. ビデオフレームに現れる変動の例を図 3 の左半分, 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を示す.

2.2 走査速度の変動のみを考慮した DP アルゴリズム

本節では走査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. この場合は, 全 T フレーム分のスリットを並べて生成した画像 (図 3(a) の右側の画像) と全 C カテゴリ分の標準パターン (横幅 J_C 画素) 間での最適なマッチングを求めること, テキスト認識およびモザイクングを行うことができる. 図 4 に手法の概要を示す. 走査速度の変動により, スリットを並べて生成した画像内において, 各文字は水平方向に非線形伸縮されていると考えられる. 従って解くべき問題は, 各文字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 のように入

```

/* Initialization */
1 for c := 1 to C do begin
2   g1(c, 1) := d1(c, 1)
3   for j := 2 to Jc do
4     g1(c, j) := ∞
5   end
6   D1 := ∞
/* DP Recursion */
7 for t := 2 to T do begin
8   for c := 1 to C do begin
9     gt(c, 1) := dt(c, 1) + min{gt-1(c, 1), Dt-1}
10    for j := 2 to Jc do
11      gt(c, j) := dt(c, j) + minj' ∈ {j, j-1, j-2} gt-1(c, j')
12    end
13    Dt := minc' ∈ C gt(c', Jc')
14  end

```

図5 走査速度の変動のみを考慮した DP アルゴリズム

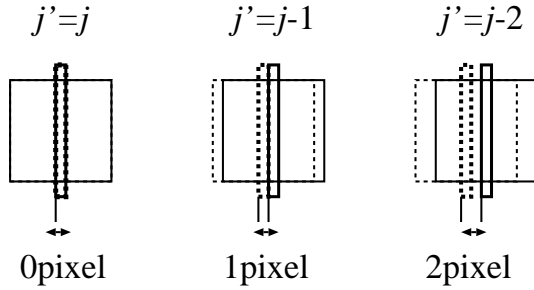


図6 選択される j' と補償される走査速度変動の関係

力フレームを横軸に、各単語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5にその DP アルゴリズムを示す。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。ステップ9, 11の $d_t(c, j)$ は、 t フレーム目のスリットと文字 c の第 j 列とのマッチング距離(局所距離)であり、本手法では単純な輝度値を画素特徴とした単純重ね合わせ距離を用いている。また、 $g_t(c, j)$ は、 t フレーム目のスリットが標準パターンの文字 c の第 j 列目に対応するまでの、初期フレーム ($t = 1$) からの最小累積距離である。

DP 漸化式内での j' に関する最小値選択は、走査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6に示すように、走査速度が 2pixel/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、0pixel/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。

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、文字の切り出しと認識の結

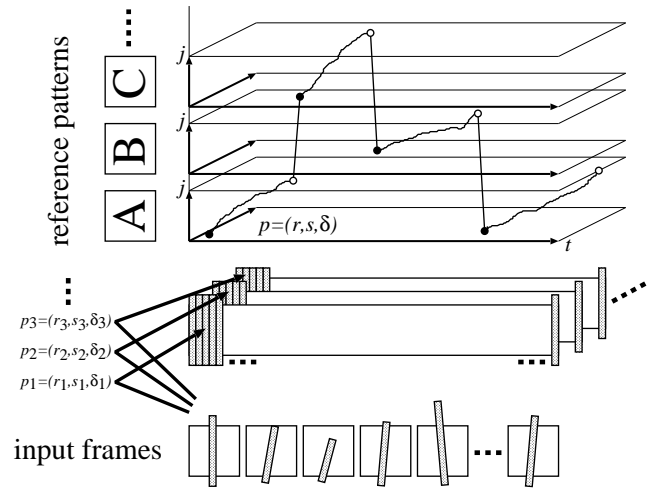


図7 一般的な手ぶれを考慮した mosaicing-by-recognition 法

```

/* Initialization */
1 for all p ∈ {(r, s, delta)} do begin
2   for c := 1 to C do begin
3     g1(p, c, 1) := d1(p, c, 1)
4     for j := 2 to Jc do
5       g1(p, c, j) := ∞
6     end
7     D1(p) := ∞
8   end
/* DP Recursion */
9 for t := 2 to T do begin
10  for all p ∈ {(r, s, delta)} do begin
11    for c := 1 to C do begin
12      gt(p, c, 1) := dt(p, c, 1)
13      + minp' ∈ pre(p) {gt-1(p', c, 1), Dt-1(p')}
14    for j := 2 to Jn do
15      gt(p, c, j) := dt(p, c, j)
16      + minp' ∈ pre(p) minj' ∈ {j, j-1, j-2} gt-1(p', c, j')
17    end
18    Dt(p) := minc' ∈ C gt(p, c', Jc')
19  end
20 end

```

図8 一般的な手ぶれを考慮した DP アルゴリズム

果を得る。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 DP アルゴリズム

本節では走査速度の変動に加え、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。手ぶれによる変動に対処するための中心的な発想は、スリット形状を動的に制御しながら標準パターンとのマッチングを行うことである。具体的には、各フレームのスリットの形状を図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 $p = (r, s, \delta)$ により変形させる。

まず、スリット形状の制御に関して、具体的な処理を以下に説明する。以下で述べる基準スリットとは前節で用いていたスリット (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。

各フレーム内において、形状制御されたスリット内の部分画像を全フレームについて並べた画像を考える (図 7)。ただし並べる際には、そこで仮定されている回転などの変動を補償し、基準スリットと同様の形状にしておく。

こうした形状制御の下、スリット内の部分画像を全てのフレームについて並べてから、標準パターンとの最適なマッチングを行うことを考える。図 7 に手法の概要を示す。ここで各フレームでは、 r, s, δ の各々について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 8 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異なり、マッチング距離の総和は $g_t(p, c, j)$ 、マッチング距離は $d_t(p, c, j)$ と表現される。ステップ 14 では、フレーム間の変動は類似しているという仮定の下、以下の制約式を満たす $\text{pre}(p)$ より最小値を選択している。

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

2.2 節と同様に、得られた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得る。

2.4 計算量

走査速度の変動のみの場合の計算量は、入力フレーム数 T 、標準パターン数 (カテゴリ数) C 、標準パターンの横幅 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、走査速度の変動のみを考慮した場合のモザイク画像の生成手法について述べる。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。このため、モザイクは各スリット間の水平位置をずらすことにより行うことができる。この水平位置のずれは、図 6 に示したように、DP 漸化式計算の際に選択した j' の値から定まる。よって、バックトラックにより順次 j' を求めれば、図 6 の規則に従いフレーム画像を重ね合わせることでモザイクが可能となる。

次に、一般的な手ぶれを考慮した場合について述べる。この場合のモザイクは、最適化された形状制御パラメータ $p = (r, s, \delta)$ と水平方向のずれ j' をバックトラックにより求めることを行うことができる。すなわち、バックトラックにより各フレームの (r, s, δ) を求め、それを用いてフレームの変動を補償すれば、あとは速度の変動のみの場合と同様に考えること

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

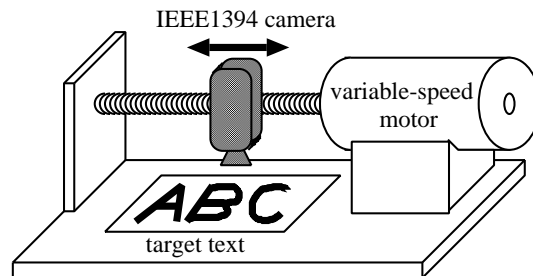


図 9 動画データ取得装置

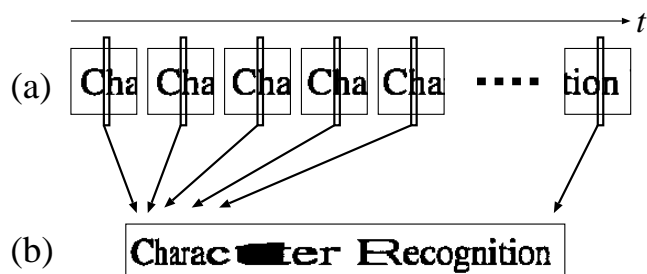


図 10 走査速度の変動のみを含む、(a) 動画データの例、(b) スリットをならべて得られた合成画像

で、モザイクを行うことが可能となる。

このようにモザイクは、2.2 節および 2.3 で述べた最適マッチングの結果を利用して行うため、別途各フレーム間での対応点を探索しておく必要はない。

2.6 スリット幅の考察

今回の手法では、スリットの横幅を 1 画素としている。この長所として、前述したように、問題を単純化できることが挙げられる。特に各スリットに同時に 2 つ以上の文字が含まれないことは重要である。しかし短所として、マッチングを行う際に、フレーム内に含まれる画像情報のほとんどを捨てていることが挙げられる。

従って、今後の課題として、アルゴリズムをスリット内に 2 つ以上の文字が含まれる場合にも拡張し、フレーム内の情報を有効に活用することが挙げられる。その結果、マッチングの精度の向上により、正確なテキスト認識およびモザイクができると考えられる。特に現在の手法では、 j' が最小 $j - 2$ までしかとれない、すなわち 2pixel/frame 内のカメラ進行速度しか許容されていないが、この制限についてもスリット幅を拡大することで緩和できると考えられる。さらに、モザイク画像を生成する際も、重複している部分の画像情報を利用することで高品質なモザイク画像が得られると考えられる [7], [8].

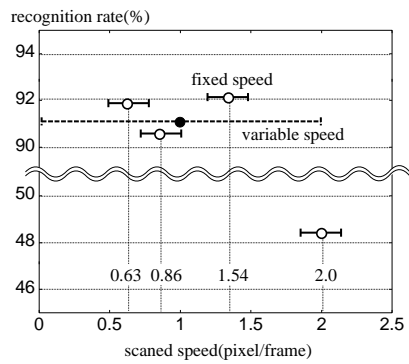


図 11 走査速度の変動のみの場合の実験結果



図 12 図 10 のデータについての (a) モザイク画像, (b) 認識結果

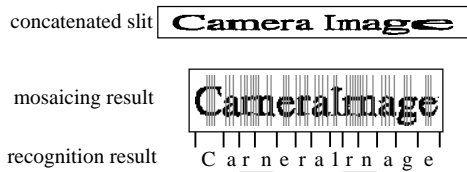


図 13 文字の分離による誤認識例

3. 実験結果

3.1 走査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 9 の装置を用いて撮影した。この装置は水平方向の走査速度をつまみで調節できるようになっている。走査速度を 0.63, 0.86, 1.34, 2.00 pixel/frame で固定した場合と、0 ~ 2 pixel/frame の範囲で変動させた場合を撮影した。

図 10(a) に走査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を並べた画像の例を示す。走査速度可変の場合、速度の変化は同図 (b) にあるように、文字の非線形伸縮として現れる。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 節で述べた走査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。実験結果を図 11 に示す。同図では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データには、走査速度可変の場合だけでなく、走査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、走査速度が固定の場合、0.63, 0.86, 1.34 pixel/frame については認識率 90% 前後であった。走査速度が 2 pixel/frame 以上になると、極端に認識率は下がってしまった。これは、本手法で補償可能な走査速度が 0 ~ 2 pixel/frame

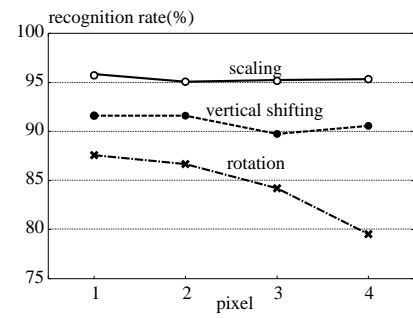


図 14 回転, 拡大縮小, 上下移動のいずれかを含む場合の実験結果

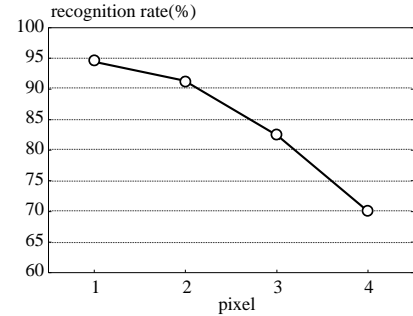


図 15 一般的な手ぶれをすべて含む場合の実験結果

の範囲であり、それを越えた部分があると速度変動を補償できないためである。一方、走査速度が可変の場合、認識率は 90% 前後であった。文字の特徴量として輝度値、またその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、この程度の認識率でも本手法の有効性が確認できたとと言える。

図 12 に、図 10 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 12 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モザイクされている様子がわかる。誤認識には、図 13 に示すような文字の分離が多がみられた。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。

3.2 一般的な手ぶれを考慮した場合

3.2.1 実験試料

図 9 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。データは、フレーム内に現れる変動 (r, s, δ) を最大で $\pm 1 \sim \pm 4$ 画素の場合としたの生成した。

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。図 14 に、回転, 拡大縮小, 上下移動の変動を個別に施した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

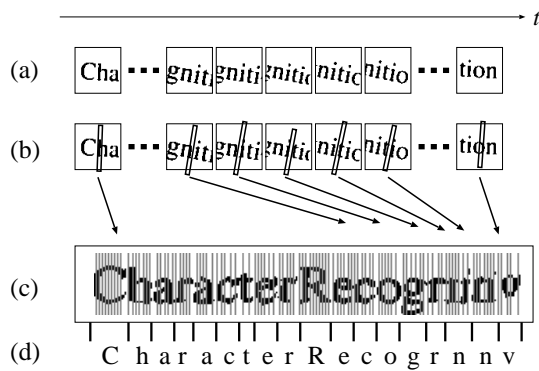


図 16 より一般的な手ぶれ変動を含む場合の (a) フレーム系列, (b) 最適形状制御されたスリット, (c) モザイク画像, (d) 認識結果

図 2(b) を図 2(a) の形状に戻す際) に行う線形補間により, 画素位置に微小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, 微小なずれ (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られ.

図 15 に, 全ての変動を同時に施した場合についての実験結果を示す. 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, 図 14 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 16 に, 図 10 に人工的に手ぶれを加えた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. 前半部分のテキストについては認識およびモザイク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モザイクされている. 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, 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, 文字の切り出しに失敗し, 文字の分離および結合による誤認識 (同図 (d)) や, 精度の低いモザイク (同図 (c)) を行ってしまう. よってマッチングの精度を向上させるために, 文字の特徴量やマッチング距離の検討が必要がある.

4. ま と め

本報告では, 動画像中のテキスト認識を目的としたビデオモザイク手法として, 手ぶれ変形の補償, 複数フレームの接合 (モザイク), 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, 走査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, 走査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「rn」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みられた. また一般的な手ぶれを対象とした場合は, 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.

今後の課題としては以下の 4 点が挙げられる.

- 1 スリット幅の拡張: 2.6 節で述べたように, 幅 1 画素のスリットではフレーム内で利用される情報が少ない. また, 幅が狭いために走査速度が制限される. よって, フレーム内の情報の有効活用および走査速度の高速化のためにスリット幅の増加を検討する.
- 2 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.
- 3 単語辞書の利用: segmentation-by-recognition 法で良く用いられているように, 辞書の利用は有効であると考えられる. これにより 「m」が 「rn」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.
- 4 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. 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.

謝辞 本研究の一部は (財) 大川情報通信基金の研究助成金によった.

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-, 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, 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, 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ク," 画像の認識・理解シンポジウム MIRU2004, vol. 1 of 2, pp. I-505-I-510, 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, 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, 1998.