

# Combinando Características Complementares em Modelos Escondidos de Markov: uma otimização para o reconhecimento de caracteres manuscritos

Murilo Santos<sup>1</sup>, Luis E. S. Oliveira<sup>1</sup>, Robert Sabourin<sup>2</sup>, Alessandro L. Koerich<sup>1</sup> e Alceu de S. Britto Jr<sup>1</sup>

<sup>1</sup>Programa de Pós-Graduação em Informática Aplicada, PPGIA  
Rua Imaculada Conceição, 1155 Prado Velho - Curitiba (PR) – Brasil  
{soares, alceu, koerich}@ppgia.pucpr.br

<sup>2</sup>École de Technologie Supérieure (ETS)  
1100 Rue Notre Dame Ouest - Montreal (QC) H3C 1K3 – Canadá

**Abstract.** Different strategies for combination of complementary features in a HMM-based method for handwritten character recognition are evaluated. In addition, a noise reduction method is proposed to reduce the negative impact of low probability symbols in the training database. New sequences of observations are generated based on the original ones. The experimental results based on 52 classes of alphabetic characters and more than 23,000 samples have shown that the proposed strategies used to optimize the HMM-based recognition method are promising.

**Keywords:** Reconhecimento de Caracteres Manuscritos, Modelos Escondidos de Markov, Combinação de Características Complementares

## 1 Introdução

Muitos esforços têm sido feitos na busca por soluções para o reconhecimento de textos manuscritos. A motivação está nas inúmeras aplicações possíveis. Neste contexto, constata-se na literatura certa maturidade nas pesquisas voltadas ao reconhecimento de dígitos isolados manuscritos. Taxas de reconhecimento acima dos 99% [1-3] têm sido atingidas usando-se diferentes métodos e estratégias. Contudo, quando se considera o reconhecimento de caracteres alfabéticos manuscritos não se verifica o mesmo, pois o problema demonstra maior complexidade. A diferença mais óbvia está no número de classes que pode chegar a 52 dependendo se letras maiúsculas (A-Z) e minúsculas (a-z) precisam ser distinguidas. Conseqüentemente há uma maior ocorrência de ambigüidades que àquela observada entre as classes de dígitos. Koerich et al. [4] acrescenta que a complexidade se deve ainda a outras diferenças, tais como: a variedade de padrões utilizadas para representar um caractere específico; a representação cursiva das letras; e a grande quantidade de caracteres com

traçado baseado em múltiplos segmentos desconexos (em inglês, *multi-stroke characters*). Na verdade, pode-se afirmar que o reconhecimento de caracteres manuscritos ainda é um problema em aberto e que o sucesso no reconhecimento de textos manuscritos ainda depende de mais pesquisas neste campo. Dentre as contribuições importantes destacam-se os trabalhos de Oh e Suen [5], Dong et al. [6], Koerich et al. [7] e Britto et al. [8]. Destes, os três primeiros consideram o uso de redes neurais e o último tem como base a modelagem Markoviana.

O uso da modelagem Markoviana merece destaque, pois esta tem mostrado ser uma alternativa interessante uma vez que modelos para caracteres isolados podem ser combinados dinamicamente durante o reconhecimento de cadeias numéricas ou palavras cursivas com sucesso [8, 9]. Além disso, é possível adicionar aos modelos conhecimento referente à interação entre os caracteres os quais variam na forma de acordo com o contexto, ou seja, a sua posição na palavra. Contudo, o processo de modelagem baseado em modelos de Markov é complexo e diferentes fatores devem ser levados em consideração, tais como: a topologia dos modelos – número de estados e observações por estado; a maneira como se representar e estruturar as características utilizadas, dentre outras. O método proposto em [8] é um exemplo típico deste tipo de abordagem, e tem sido utilizado com êxito no reconhecimento de caracteres isolados, cadeias numéricas e palavras cursivas. Contudo, ainda restam pontos a serem investigados na busca da otimização dos modelos Markovianos utilizados.

Este artigo tem como objetivo justamente buscar a otimização dos modelos utilizados no método proposto em [8]. Para tal, considera-se a investigação de diferentes abordagens para a combinação das características complementares utilizadas no método, bem como a avaliação de uma estratégia para a redução do impacto de ruídos normalmente presentes nas seqüências de observações discretas utilizadas no treinamento dos mesmos. Investiga-se a geração de novas seqüências de observações com base em filtragem para detecção de observações cuja ocorrência está abaixo de um limiar experimentalmente estabelecido.

O trabalho apresenta 5 seções. A Seção 2 apresenta uma visão geral do método otimizado, uma descrição dos conjuntos de características e dos Modelos Escondidos de Markov (MEM) utilizados. A Seção 3 traz as estratégias de otimização avaliadas: diferentes formas de combinação dos conjuntos de características extraídos dos caracteres a serem reconhecidos e a redução do impacto de ruídos com base na manipulação das seqüências de observações discretas utilizadas no treinamento dos modelos. A Seção 4 traz os resultados experimentais, enquanto a Seção 5 apresenta as conclusões e trabalhos futuros.

## 2 Visão Geral do Método

A principal premissa do método de reconhecimento utilizado está na combinação de características complementares extraídas do fundo e do traçado de caracteres manuscritos. Um esquema de zoneamento baseado em modelos de colunas e linhas fornece um modo de dividir o caractere em regiões absorvendo a variação de tamanho inerente a este tipo de padrão (ver Figura 1). A imagem de um caractere é varrida da esquerda para direita e de cima para baixo para a extração das características de colunas e linhas, respectivamente. No método original estas características são

combinadas em um mesmo vetor de valores contínuos e dois conjuntos de modelos de Markov são utilizados, a saber: modelos de colunas e modelos de linhas. Cada conjunto contém um modelo para cada classe de caractere.

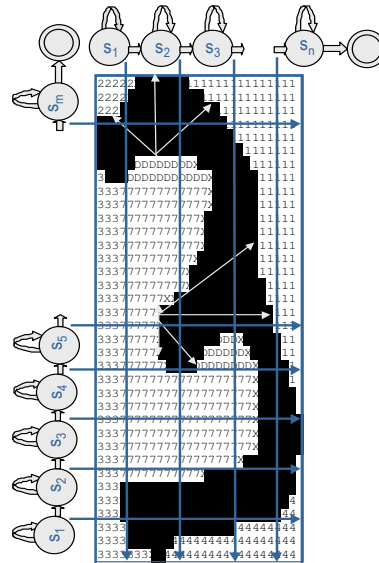


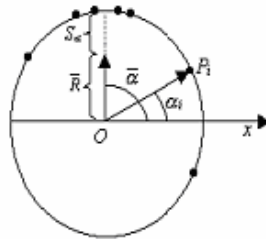
Figura 1. Esquema de zoneamento obtido pela combinação de MEMs de coluna e linha

### 2.1 Método de Extração de Características

O método de extração de características consiste em varreduras da imagem do caractere da esquerda para a direita (características de colunas) e de cima para baixo (características de linhas). Para cada linha e coluna da imagem são calculadas características do traçado e do fundo do caractere, sendo 34 características do traçado (baseadas em transições e superfície) e 13 de fundo (baseadas em concavidades).

**Características do Traçado (CT):** as características do traçado consistem em informações locais e globais calculadas levando em conta os *pixels* do traçado (ou em inglês, *foreground pixels*) da imagem do caractere obtidas para cada coluna e linha. As características locais são baseadas em transições dos *pixels* de fundo para os *pixels* do traçado e vice-versa. Para cada transição, a direção média e a variância correspondente são obtidas através de estimativa de média circular. Estas estimativas são mais sutis para observações direcionais, uma vez que são baseadas em escala circular. Como exemplo, dada a observação direcional  $\alpha_1 = 1^0$  e  $\alpha_2 = 359^0$ , estas fornecem a direção media ( $\bar{\alpha}$ ) de  $0^0$  ao invés de  $180^0$  calculado pela estimativa

convencional. Considerando  $\alpha_1, \dots, \alpha_i, \dots, \alpha_n$  ser o conjunto de observações direcionais com distribuição  $F(\alpha_i)$  e tamanho  $N$ .



**Figura 2.** Media de direção circular  $\bar{\alpha}$  e variância  $S_\alpha$  para distribuição  $F(\alpha_i)$

A Figura 2 mostra que  $\alpha_i$  representa o ângulo entre o vetor unitário  $\overline{OP}_i$  e o eixo horizontal, enquanto  $P_i$  é o ponto de interseção entre  $\overline{OP}_i$  e o círculo unitário. As coordenadas cartesianas de  $P_i$  são definidas como:

$$(\cos(\alpha_i), \sin(\alpha_i)) \tag{1}$$

A direção média circular  $\bar{\alpha}$  das  $N$  observações direcionais no círculo unitário correspondem à direção do vetor resultante  $\bar{R}$  obtido pela soma dos vetores unitários  $(\overline{OP}_1, \dots, \overline{OP}_i, \dots, \overline{OP}_N)$ . O centro de gravidade  $(\bar{C}, \bar{S})$  das  $N$  coordenadas  $(\cos(\alpha_i), \sin(\alpha_i))$  são definidas como:

$$\bar{C} = \frac{1}{N} \sum_{i=1}^N \cos(\alpha_i) \quad \text{ou} \quad \bar{S} = \frac{1}{N} \sum_{i=1}^N \sin(\alpha_i) \tag{2}$$

Estas coordenadas são usadas para estimar o tamanho médio de  $\bar{R}$ , como:

$$\bar{R} = \sqrt{(\bar{C}^2 + \bar{S}^2)} \tag{3}$$

Então, a direção circular média pode ser obtida resolvendo uma das equações a seguir:

$$\cos(\bar{\alpha}) = \frac{\bar{C}}{\bar{R}}, \quad \sin(\bar{\alpha}) = \frac{\bar{S}}{\bar{R}} \tag{4}$$

Finalmente, a variância circular de  $\bar{\alpha}$  é calculada como:

$$S_\alpha = 1 - \bar{R} \quad 0 \leq S_\alpha \leq 1 \tag{5}$$

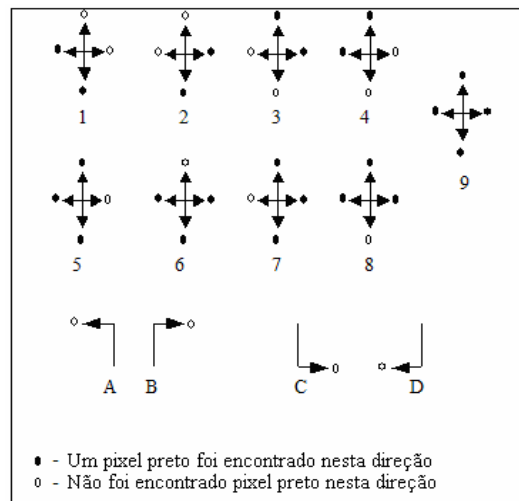
Para estimar  $\bar{\alpha}$  e  $S_\alpha$  para cada transição de um caractere temos que considerar  $\{0^\circ, 45^\circ, 90^\circ, 135^\circ, 180^\circ, 225^\circ, 270^\circ, 315^\circ\}$  como um conjunto de observações direcionais, enquanto  $F(\alpha_i)$  é computado através da contagem do número de *pixels* pretos

sucessivos na direção  $\alpha_i$  de uma transição até aparecer um *pixel* branco. Na Figura 3 é possível observar as transições em uma das colunas do caractere “e”, as quais são enumeradas de 1 a 6. Além disso, destacam-se as possíveis observações direcionais das transições 3 e 5.



**Figura 3.** Transições em uma coluna do caractere “e”, e as observações direcionais para estimar a direção média das transições 3 e 5.

Adicionalmente a esta informação direcional, calculam-se duas outras características locais: a) posição relativa de cada transição, levando em conta o topo do caractere e b) se a transição pertence ao contorno interno ou externo, a qual mostra a presença de contornos fechados na imagem do caractere. Para cada coluna consideram-se 8 possíveis transições. Neste ponto o vetor de característica é composto por 32 valores. As características globais são baseadas em projeções verticais (PV) de pixes pretos para cada coluna, e a derivativa de PV entre colunas adjacentes. Finalmente, as características CT perfazem um total de 34 valores normalizados entre 0 e 1.



**Figura 4.** Configurações de Concavidade

**Características de Fundo (CF):** consistem em informações referentes às concavidades do caractere. Estas características são usadas para destacar as propriedades de topologia e geometria das classes de caracteres. Cada característica de concavidade representa o número de *pixels* brancos que pertencem a uma configuração específica de concavidade. O rótulo para cada *pixel* é escolhido baseado nos códigos de quatro direções. Cada direção é explorada até encontrar um *pixel* preto ou o limite imposto pela menor caixa que contém o caractere (*bounding box*). Um *pixel* branco é rotulado se ao menos duas direções consecutivas atingirem *pixels* pretos. Assim, obtém-se 9 possíveis configurações. No entanto, consideramos mais quatro configurações, para detectar com mais precisão a presença de contornos fechados. Desta forma, o número total de características CF é 13. O vetor de concavidade é normalizado entre 0 e 1, dividindo-se cada entrada pelo total de códigos de concavidade computado para cada coluna ou linha da imagem do caractere. A Figura 4 mostra as 9 configurações de concavidade e também 4 configurações (A,B,C,D) para falsos contornos fechados.

## 2.2 Modelos escondidos de Markov

Cada classe de caractere é representada por dois MEMs discretos: um baseado em colunas  $(\lambda_c^a, \lambda_c^b, \dots, \lambda_c^z)$  e outro baseado em linhas  $(\lambda_l^a, \lambda_l^b, \dots, \lambda_l^z)$  da imagem do caractere. Estes modelos de colunas e linhas provêm um modo de combinar características complementares de traçado e de fundo em um sistema zoneado como mostrado na Figura 1. A entrada de cada modelo são seqüências de observações discretas geradas a partir da discretização dos vetores de características extraídos dos caracteres. Para tal, cada vetor é mapeado para um de 256 possíveis símbolos discretos disponíveis em um *codebook* previamente construído usando o algoritmo de quantização vetorial K-means [8]. Assim, a saída do método de extração de características consiste em seqüências de observações discretas para cada caractere: ao menos uma seqüência de observações extraídas das colunas e outra das linhas.

A topologia do modelo de caractere é definida levando em conta o reconhecimento do texto manuscrito. Isto significa o uso de um modelo esquerda-direita com o número de estados definido como descrito em [10].

## 3. Estratégias de Otimização

As estratégias de otimização avaliadas consistem em diferentes abordagens para a combinação das características complementares descritas nas seções anteriores, bem como no estudo da redução do impacto de ruídos normalmente presentes nas seqüências de observações discretas utilizadas no treinamento dos modelos de Markov. Investiga-se a geração de novas seqüências de observações com base em filtragem para detecção de observações cuja ocorrência está abaixo de um limiar experimentalmente estabelecido.

### 3.1 Combinando CT e CF

Duas novas estratégias foram investigadas além da combinação em um mesmo vetor já implementada no método original. As novas estratégias dizem respeito à combinação das características CT e CF usando diferentes matrizes de observações por estado (ou diferentes matrizes B) nos modelos de Markov de linhas de colunas do caractere, e a combinação considerando-se múltiplos modelos de Markov, ou seja, quatro modelos (2 modelos baseados em CT e CF extraídos das colunas e 2 modelos baseados em CT e CF extraídos das linhas do caractere). Os resultados experimentais são apresentados na Seção 5.

**Vetor único:** informações de traçado e fundo são combinadas em um vetor de 47 características sendo: 34 de traçado e mais 13 de fundo. Estas características são extraídas separadamente para as colunas e linhas, assim o método funciona com dois MEMs para cada classe de caractere.

**Múltiplas matrizes B:** estratégia utiliza múltiplas matrizes B para cada MEM. Cada MEM usualmente utiliza uma matriz A que a matriz de probabilidade de transição de estado, e uma matriz B que é a matriz de probabilidade de observação de símbolos por estado. A estratégia aplicada foi utilizar dois MEMs e cada MEM utilizando uma matriz A e duas matrizes B, sendo assim temos cada matriz B contendo apenas informações sobre um tipo de características, a saber: a) traçado coluna; b) traçado linha; c) fundo coluna; e d) fundo linha. O intuito desta estratégia é separar as características de traçado e de fundo para que cada matriz B apresente uma distribuição de probabilidades mais adequada. Logo, neste caso, cada modelo considerando duas matrizes de observações independentes a cada passo no tempo discreto tem os parâmetros  $\lambda = (A, B^0, B^1, \pi)$  onde  $B^0$  é a matriz de probabilidade de observações por estado  $b^0_j(k)$ , que nos dá a probabilidade de observar o símbolo (ou observação)  $k$  no estado  $j$ , e  $B^1$ , de maneira similar, é a matriz de probabilidade de observações por estado  $b^1_j(k)$ . No treinamento e no reconhecimento  $b_j(k)$  é substituído por  $b_j(k^0, k^1)$ , o qual é computado como:

$$b_j(k^0, k^1) = b^0_j(k^0) \times b^1_j(k^1) \quad (6)$$

Desta forma cada um dos dois MEMs sendo um que contém características de linha e o outro de colunas passarão a ter suas subcategorias (traço e fundo) processadas em matriz B separadas gerando assim um total de quatro matrizes B.

Alterações no algoritmo de Baum-Welch [10] utilizado para treinamento dos modelos, e no algoritmo de Viterbi [10] utilizado para classificação dos modelos foram realizadas a fim de se adequarem aos parâmetros da equação (6).

**Múltiplos modelos:** consiste em combinar quatro MEMs, cada MEM treinado com base em um tipo de característica sendo: a) traçado coluna; b) traçado linha; c) fundo coluna; d) fundo linha. Como na estratégia do uso de múltiplas matrizes B a intenção é que cada modelo criado se adapte melhor ao tipo de características nele inserido.

### 3.2 Redução dos Ruídos na Base de Treinamento

A redução do impacto de ruídos na base de seqüências de observações discretas utilizadas no treinamento dos modelos de caracteres pode ser obtida através da geração de novas seqüências a partir das existentes. A estratégia aqui denominada de “Deixe uma de fora” (DUF) pode ser utilizada para aumentar o número de seqüências de observações na base de treinamento, e desta maneira torná-la mais consistente e menos susceptível a ruídos. Uma abordagem semelhante é apresentada em [11], porém neste trabalho a escolha da observação a ser deixada de fora se dá através de um redutor de ruídos que desenvolvemos, o qual permite reduzir o impacto de observações com baixa freqüência na base de treinamento. A idéia então é deixar de fora a observação cuja freqüência na base de treinamento é menor que determinado limiar, experimentalmente definido. Desta forma, é possível criar a partir de cada seqüência de observações  $M$  novas seqüências, onde  $M$  será menor que o número de observações disponíveis na seqüência processada (ver Figura 5).

O uso de redutor de ruídos permite que somente observações que são possíveis candidatas a serem ruídos sejam eliminadas na seqüência original para criar uma nova seqüência.

Seqüência Original: (a, b, c, d, e)
Seqüências Geradas: (b, c, d, e)
(a, c, d, e)
(a, b, d, e)
(a, b, c, e)
(a, b, c, d)

Figura 5. Seqüência de observações geradas pelo DUF a partir da seqüência original

## 4. Resultados Experimentais

A base de dados utilizada para os experimentos foi a NIST SD19 [12]. Foram utilizadas 74,880 amostras das series *hsf\_0*, *hsf\_1*, *hsf\_2*, e *hsf\_3* para treinamento, 23,670 amostras da serie *hsf\_7* para validação, e 23.941 amostras da serie *hsf\_4* para testes. Duas linhas de experimentos foram seguidas: um experimento foi baseado na combinação de CT e CF em múltiplas matrizes B, e o outro na combinação de CT e CF em múltiplos modelos, a combinação de CT e CF em um mesmo vetor é apresentada em [8]. A estratégia DUF não é usada nos experimentos baseados em diferentes matrizes B pelo fato que é necessário ter a mesma quantidade de observações nas seqüências a fim de combiná-las. Contudo, é possível observar a contribuição da redução de ruídos quando consideram-se múltiplos MEMs por caractere.

### 4.1. Experimentos Considerando Múltiplas Matrizes B

Os experimentos combinando múltiplas matrizes B mostraram uma melhoria na taxa de reconhecimento comparado ao método original, conforme observa-se na Tabela

1. Este aumento na taxa de reconhecimento se deve ao fato que cada matriz B trata individualmente um tipo de característica: CT ou CF.

**Tabela 1.** Taxas de reconhecimento em (%)

Letras	Vetor Único	Múltiplas Matrizes B	Múltiplos Modelos	Múltiplos Modelos + DUF em (%)
Minúsculas	<b>84,0</b>	<b>85,03</b>	<b>84,94</b>	<b>85,67</b> DUF 0.12
Maiúsculas	<b>90,0</b>	<b>91,44</b>	<b>91,14</b>	<b>91,62</b> DUF 0.16
52 Classes	<b>87,0</b>	<b>88,27</b>	<b>87,76</b>	<b>88,22</b> DUF min. 0.11, mai. 018

#### 4.2 Experimentos Considerando Múltiplos MEMs

Combinando múltiplos modelos obtivemos melhorias nas taxas de reconhecimento semelhantes à obtida na estratégia de combinar múltiplas matrizes B como mostra a Tabela 1. Contudo, usando a estratégia de combinar distintos MEMs temos a vantagem de poder considerar quantidades de vetores de observações diferentes em cada MEM e também diferentes quantidades de observações por seqüência. Estas particularidades dão a esta estratégia a liberdade de se avaliar a redução de ruídos manipulando-se a seqüência de observações da base de treinamento – conforme aplicada no próximo experimento.

**Tabela 2:** Comparação com Trabalhos Relacionados

Método	Treinamento	Validação	Teste	Taxa Rec. (%)
(Oh, 1998)	26,000	-	11,941	90.0
Minúsculas (26 classes)				
(Dong, 2001b)	23,937	-	10,688	92.3
Minúsculas (26 classes)				
(Koerich, 2002)				
Maiúsculas (26 classes)	37,440	12,092	11,941	92.3
Minúsculas (26 classes)	37,440	11,578	12,000	84.6
Ambas (52 classes)	74,880	23,670	23,941	85.5
(Britto, 2004)				
Maiúsculas (26 classes)	37,440	12,092	11,941	90.0
Minúsculas (26 classes)	37,440	11,578	12,000	84.0
Ambas (52 classes)	74,880	23,670	23,941	87.0

#### 4.3 Experimento Considerando Redutor de Ruídos

A redução do impacto de ruídos na base de seqüências de observações discretas utilizadas no treinamento dos modelos de caracteres foi obtida através da geração de novas seqüências a partir das existentes. Para a escolha da observação a ser deixada de fora na estratégia DUF, foram avaliados diferentes valores de probabilidade como limiares. O valor que apresentou melhores resultados foi 1% (ver Tabela 1), ou seja, foram deixadas de fora para geração de novas seqüências as observações cujas probabilidades de aparecimento na base de treinamento eram menores que 1%.

## 5 Conclusão

Este artigo apresentou diferentes estratégias para otimizar um sistema baseado em MEM para reconhecimento de caracteres manuscritos, considerando a combinação de características em múltiplas matrizes B, a combinação em múltiplos MEMs, bem como a redução do impacto de ruídos na base de treinamento. Os resultados obtidos com a utilização de múltiplos modelos em conjunto com a estratégia de redução de ruídos na base de treinamento são promissores. Na literatura, poucos trabalhos consideram 52 classes (ver Tabela 2). É possível observar que para 52 classes os modelos otimizados são realmente promissores. Isso nos incentiva a investigar novos conjuntos de características, bem como uma avaliação mais detalhada do impacto de observações com baixa frequência em cada classe de caractere.

## 6. Referencias

1. Oliveira, L.S. e Sabourin, R., Support Vector Machines for Handwritten Numerical String Recognition, 9th International Workshop on Frontiers in Handwriting Recognition (IWFHR-9), Kokubunji, Tokyo, Japan, pp 39-44, 2004.
2. Dong J., Krzyzak A., Suen C. Y. A multi-net learning framework for pattern recognition. Proc. of the Sixth Int. Conf. on Document Analysis and Recognition, pp. 328-332, 2001.
3. Suen C. Y., Xu Q., Lam L.(1999b). Automatic recognition of handwritten data on cheques - fact or fiction? Pattern Recognition Letters, 20(13):1287-1295, 1999.
4. Koerich A. L. Large Vocabulary Off-Line Handwritten Word Recognition. PhD thesis, École de Technologie Supérieure, Montreal-Canada, August, 2002.
5. Oh I.S. e Suen C.Y. "Distance features for neural network-based recognition of handwritten characters". Int. Journal on Document Analysis and Recognition, 1(2):73-88, 1998.
6. Dong J., Krzyzak A., and Suen C.Y. "Local learning framework for recognition of lowercase handwritten characters". In Proc. Int. Workshop on Machine Learning and Data Mining in Pattern Recognition, Leipzig, Germany, pp. 226-238, 2001.
7. Koerich A. L., Sabourin R. e Suen C. Y. Lexicon-Driven HMM Decoding for Large Vocabulary Handwriting Recognition with Multiple Character Models. *Int. Journal on Document Analysis and Recognition (IJ DAR)*, Vol.6 No.2, pp.126-144, October 2003.
8. Britto JR., A. S., Sabourin R., Bortolozzi F. e Suen C.Y. Foreground and Background Information in an HMM-Based Method for Recognition of Isolated Characters and Numeral Strings. 9th International Workshop on Frontiers in Handwriting Recognition (IWFHR-9), Tóquio, Japão, pp. 371-376, 2004.
9. Cavalin, P.; Britto JR, A. S. ; Oliveira, L. E. S. ; Sabourin, R.; Bortolozzi, F. An Implicit Segmentation based Method for Recognition of Handwritten Strings of Characters. In: Proc. of the The 21st Annual ACM Symposium on Applied Computing, 2006. v. 1. pp. 836-840.
10. Rabiner, L. R. "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition". Proc. of the IEEE, Vol. 77, No. 2, pp.257-286, 1989.
11. Ko, Albert ; Sabourin, Robert ; Britto Jr, A. S. . Leave-One-Out-Training and Leave-One-Out-Testing Hidden Markov Models for a Handwritten Numeral Recognizer: the Implication of Single Classifier and Multiple Classifications. IEEE Trans. on Pattern Analysis and Machine Intelligence, 2008.
12. Grother, P.J. NIST Special Database 19 - Handprinted Forms and Characters Database. National Institute of Standards and Technology (NIST), March 1995.