

Felipe Calliari Ribas

Caracterização de Algoritmos de Segmentação de Dígitos Manuscritos

Curitiba
2008.

Felipe Calliari Ribas

Caracterização de Algoritmos de Segmentação de Dígitos Manuscritos

Projeto de dissertação de mestrado apresentado como pré-requisito para conclusão do curso de Mestrado em Informática, da Pontifícia Universidade Católica do Paraná, área de concentração: Visão, Imagem e Robótica.

Orientador:

Prof^o. Dr^o. **Luiz Eduardo Soares de Oliveira**

Co-orientador:

Prof^o. Dr^o. **Alceu de Souza Britto Jr.**

Pontifícia Universidade Católica do Paraná

Curitiba

2008.

Sumário

Sumário	i
Lista de Figuras	ii
Lista de Tabelas	iv
Lista de Abreviaturas	v
1 Introdução	1
2 Trabalhos Relacionados	4
2.1 Fujisawa et al	4
2.2 Shi e Govindaraju	6
2.3 Oliveira et al	7
2.4 Chen e Wang	8
2.5 Yu e Yan	10
2.6 Pal et al	11
2.7 Elnagar e Alhajj	12
2.8 Lei et al	14
2.9 Suwa e Naoi	15
2.10 Sadri et al	16
2.11 Comparação dos Métodos	19
3 Método Proposto	20
4 Resultados	26
4.1 Resultados Esperados	26
4.2 Resultados Obtidos	27
5 Cronograma	29
Referências Bibliográficas	30

Lista de Figuras

1	Etapas de um sistema de reconhecimento de dígitos manuscritos.	1
2	Exemplos de imagens de entrada, as quais produziriam como saída os pares de dígitos “12” e “75” respectivamente	1
3	Exemplos de pares de dígitos e seus caminhos ótimos de segmentação.	1
4	Exemplo de caractere sólido.	2
5	Exemplos de caracteres não-sólidos. Imagem do contorno e do esqueleto, respectivamente.	2
6	Nesta imagem podemos observar a sequência de dígitos 04618-003 correspondente a um código de endereçamento postal, a qual é considerada uma sequência de dígitos manuscritos delimitados. Já o restante do texto é dito não-delimitado, pois não possui nenhuma marcação limitando o posicionamento dos caracteres.	3
7	Linhas verticais (em cinza) mostrando a distância horizontal dos ciclos (Imagem original à esquerda).	5
8	(a) Imagem Original, (b) Pontos de curvatura à direita, (c) Dígitos segmentados	6
9	Pontos de (a) contorno, (b) perfil e (c) esqueleto. BP (pontos básicos), IP (ponto de intersecção) e TP (ponto terminal).	7
10	5 possíveis tipos de conexão, segundo Chen et al [CW00].	9
11	(a) Imagem contendo “traço-ligador”, (b) identificação do traço, (c) imagem após a remoção do mesmo.	10
12	Diferentes padrões estruturais que representam as possíveis mudanças morfológicas no contorno da imagem.	11
13	Espaços (“Reservatórios”) criados pela conexão entre os dígitos, e suas respectivas bases.	11
14	Exemplos de segmentação incorreta gerada pelo método proposto.	12
15	Imagens das fases do método de segmentação: imagem original, imagem esquelizada, imagem segmentada e imagem restaurada.	12
16	Modelos utilizados na extração de características. Pontos Finais (e1;e2), pontos de ramificação (b1;b2;b3;b4) e pontos de cruzamento (c1;c2)	13
17	Conexões com ponto em comum.	13
18	Conexões com segmento em comum.	13
19	Conexões suaves.	14
20	Conexões com “traço-ligador”	14
21	Na esquerda temos uma imagem exemplificando um ponto de colina e na direita um ponto de vale. (ambos os pontos estão marcados com um “X” na imagem	14
22	Contorno inferior e superior de uma cadeia de dígitos conectados.	15
23	Par de dígitos e suas diferentes representações. (a) Imagem original, (b) Imagem do esqueleto, (c) Representação de grafo no contorno da imagem.	15

24	Tipos de conexão: (a) Conexão em um ponto, (b) Conexão em um segmento, (c) Conexão múltipla, (d) Conexão com “traço-ligador”.	16
25	(a) Imagem pré-processada, (b) Esqueleto do plano frontal, (c) Plano de fundo (pixels brancos fora do objeto composto por pixels pretos, (d) projeção do perfil superior, (e) projeção do perfil inferior, (f) esqueleto superior do plano de fundo, (g) esqueleto inferior do plano de fundo.	17
26	Casos onde provavelmente um reconhecedor de dígitos isolados cometeria erro sem a ajuda de informações de contexto. Por exemplo, (a) poderia ser facilmente reconhecido como 020, (b) como 101, (c) como 01 e (d) como 10.	18
27	Exemplos de imagens geradas pelo algoritmo de geração automática	21
28	Exemplo de imagem contendo traço-ligador, classificada alternativamente como tipo 1.	21
29	Tipos de conexão utilizados para a classificação da nossa base de dados.	22
30	Exemplos de imagens removidas da base de dados. Classe 16 e 10 respectivamente.	22
31	Dígito “1”, com e sem “perna”, respectivamente.	22
32	Gráfico da distribuição da Base de Dados de acordo com os tipos.	23
33	Imagens da classe 33, seus caminhos ótimos de segmentação e duas hipóteses. Sendo a primeira hipótese incorreta e a segunda correta.	24
34	Desempenho obtido pelo método para cada um dos tipos de conexão (fig. 29).	27
35	Quantidade média de pontos de segmentação gerados pelo algoritmo, para cada tipo de conexão (fig. 29).	27

Lista de Tabelas

1	Tabela Comparativa dos Métodos.	19
2	Cronograma descritivo das atividades do projeto.	29

Lista de Abreviaturas

ASCII	American Standard Code for Information Interchange
CC	Componentes Conectados
HMM	Hidden Markov Model
MLP	Multi-Layer Perceptron
NIST	National Institute of Standards and Technology
SVM	Support Vector Machine

1 Introdução

Este documento trata da segmentação de dígitos manuscritos, a qual é considerada a fase mais crítica em sistemas de reconhecimento de dígitos manuscritos. As principais etapas comumente presentes em sistemas deste tipo são representadas pela figura 1. Esse tipo de sistema visa processar uma imagem de entrada (fig. 2) e transformá-la em uma sequência de caracteres ASCII.



Figura 1: Etapas de um sistema de reconhecimento de dígitos manuscritos.



Figura 2: Exemplos de imagens de entrada, as quais produziram como saída os pares de dígitos “12” e “75” respectivamente

Esse tipo de sistema pode ser empregado em inúmeros problemas encontrados atualmente. Podemos citar o processamento automático de cheques, envelopes postais e formulários diversos.

A segmentação há muito tempo vem sendo pesquisada e diversos algoritmos já foram desenvolvidos. O objetivo da segmentação é dividir a imagem de entrada em regiões as quais possuam uma e apenas uma entidade completa. No caso do reconhecimento de dígitos manuscritos, essa entidade seria um dígito. Então para dividir uma imagem contendo uma cadeia de dígitos em imagens de dígitos isolados, precisamos encontrar os chamados “caminhos de segmentação” (fig. 3), que nada mais são que “cortes” que devem ser realizados na imagem para que estas sejam divididas em imagens contendo cada uma um dígito isolado.



Figura 3: Exemplos de pares de dígitos e seus caminhos ótimos de segmentação.

Diversas abordagens já foram pesquisadas para a realização da segmentação, porém a escolha depende de um conjunto de variáveis, as quais necessitam de um profundo estudo para

análise de qual pode ser mais eficaz para cada caso. Casei et al [CL96] propõem a classificação dos métodos de segmentação em dois tipos, segmentação implícita e explícita. Basicamente a segmentação explícita é aquela realizada antes do reconhecimento, tipo este correspondente a todos os algoritmos que serão abordados neste projeto. Já a segmentação implícita é realizada simultaneamente com o reconhecimento, sendo então a segmentação nada mais que o resultado do mesmo. Podemos citar como exemplo de segmentação implícita o método baseado em modelos ocultos de Markov (HMM) proposto por Britto et al [ASBS03].

Como citado no parágrafo anterior, para escolha do método de segmentação devemos levar em consideração vários fatores. O primeiro que pode ser observado é o tipo de imagem de entrada. As representações mais comuns são: caracteres sólidos binários (fig. 4), esqueleto da imagem, imagem em nível de cinza e contorno (fig. 5). Chama-se de imagem binária a imagem contendo apenas duas cores, uma para o plano frontal (no caso o dígito) e outra para o fundo, de maneira geral utiliza-se: preto e branco, respectivamente. Caractere sólido entende-se por um caractere não vazado. Como representações não-sólidas podemos citar esqueleto e contorno da imagem.



Figura 4: Exemplo de caractere sólido.



Figura 5: Exemplos de caracteres não-sólidos. Imagem do contorno e do esqueleto, respectivamente.

Imagens coloridas dificilmente são utilizadas para este tipo de aplicação, devido à maior complexidade de processamento. De modo que até o momento nenhum método proposto demonstrou um ganho em desempenho que justificasse esta complexidade. Desta maneira, de modo geral transforma-se as imagens coloridas para uma das representações citadas anteriormente antes de realizar o processamento. No oposto, a transformação para uma maneira muito simplificada, como a imagem binária, dependendo do caso, pode resultar em uma imagem com qualidade insuficiente para uma boa segmentação. Existem também métodos que utilizam mais de uma representação [NA98].

Outro ponto crucial dos métodos de segmentação são as características utilizadas. Ou seja, para descobrir os possíveis caminhos de segmentação é necessário extrair da imagem algumas características as quais quando analisadas forneçam informações que possam ajudar na localização dos possíveis caminhos de segmentação. Diversos tipos de características já foram propostos na literatura, desde as mais simples, até características com alto grau de comple-

xidade. Como exemplo de características podemos citar: relação entre largura e altura da imagem, espessura do traço, pontos de máximo e mínimo no contorno da imagem, perfil da imagem, densidade de pixels em cada região da imagem (supondo uma divisão da mesma em várias regiões), pontos de conexão e pontos terminais ambos no esqueleto da imagem, dentre diversas outras características existentes.

Outro fator importante na escolha do método a ser utilizado é o tipo de documento a ser tratado, no que diz respeito ao nível de degradação que a imagem dos dígitos pode vir a apresentar, a variabilidade de fontes a ser considerada e o fato de podermos ter que tratar de manuscritos não delimitados. Chamamos de manuscritos delimitados, imagens onde existem áreas determinadas para a escrita, como por exemplos formulários, onde cada caractere tem um espaço pré-determinado. A diferença entre caracteres delimitados e não delimitados é representada pela figura 6.

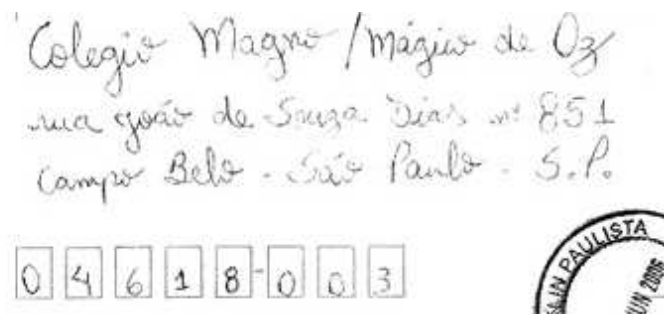


Figura 6: Nesta imagem podemos observar a sequência de dígitos 04618-003 correspondente a um código de endereçamento postal, a qual é considerada uma sequência de dígitos manuscritos delimitados. Já o restante do texto é dito não-delimitado, pois não possui nenhuma marcação limitando o posicionamento dos caracteres.

Outro ponto a ser observado é se o método a ser escolhido é invariante as distorções que o problema em questão pode vir a apresentar. Variações estas que podem ser inclinações, rotações, dentre outras. De modo geral, para definir o método de segmentação a ser utilizado, devemos levar em conta todos os pontos levantados nos parágrafos anteriores.

Já foram propostos também métodos para o reconhecimento de manuscritos onde a segmentação não é necessária e pares de caracteres conectados são enviados diretamente ao classificador, um exemplo deste tipo de método é o proposto por Choi e Oh [CO99]. Este método utiliza um classificador que possui 100 classes na saída (00 à 99). Como problemas deste tipo de método temos: a dificuldade de treinar um classificador com tão alto número de classes, que possua bom desempenho, além do alto custo computacional do mesmo.

Na seção 2 estão relacionados os principais algoritmos existentes, com uma breve análise de suas particularidades e desempenho obtido. Na seção 3 é descrito o método a ser adotado para o desenvolvimento do trabalho. Já na seção 4 são colocados os resultados que esperamos atingir ao longo deste trabalho, bem como alguns resultados que já foram obtidos. Por último na seção 5 apresentamos um cronograma descritivo das atividades do projeto.

2 Trabalhos Relacionados

Nesta seção serão descritos os principais métodos de segmentação de dígitos manuscritos presentes na literatura, destacando suas particularidades e comparando os desempenhos obtidos nos testes realizados pelos autores. Todos os métodos são de segmentação explícita, pois é neste tipo de algoritmo que todo o projeto está fundamentado. Todos os algoritmos descritos à seguir serão implementados nas próximas etapas do projeto.

2.1 Fujisawa et al

Fujisawa et al [HFK92] propõem um método de segmentação baseado em reconhecimento, ou seja, o algoritmo utiliza um classificador para selecionar uma dentre as hipóteses de segmentação geradas.

Basicamente o algoritmo num primeiro passo identifica através de perseguição de contorno, todos os componentes conectados (CCs) presentes na imagem. Então estes CCs são classificados em dígitos isolados ou dígitos conectados, através de dois limiares baseados na largura comum de dígitos isolados. O contorno é dividido horizontalmente em duas partes: contorno superior e contorno inferior. Esta separação é feita com base nos ponto mais a esquerda da imagem e no ponto mais a direita. A busca por pontos de segmentação é então feita individualmente para cada parte do contorno. Para cada um dos contornos, superior e inferior, é obtida uma função. Essas funções são então utilizadas para calcular a chamada largura vertical do traço. Esta medida é comparada com um limiar para encontrar as possíveis regiões de toque.

As regiões que possuem conexão entre dois ciclos fechados (*“loops”*), são tratadas de maneira diferenciada. O algoritmo extrai os contornos destes ciclos e divide a imagem verticalmente, separando os ciclos existentes em dois grupos: pertencentes ao dígito da esquerda ou ciclos pertencentes ao dígito da direita. Para que este algoritmo seja válido, a seguinte condição deve ser satisfeita: deve haver uma distância horizontal (fig. 7) mínima entre o ciclos que pertencem ao grupo da esquerda e os ciclos que pertencem ao grupo da direita. Esta distância mínima é dada por um limiar pré-determinado. Caso a distância seja menor do que o limiar, então o CC é tratado como sendo um único componente, o qual possui mais de um ciclo. Poderia ser, por exemplo, o dígito 8. Porém se a distância for maior que o limiar, então ali existe uma região de conexão e os pontos são comparados com pontos obtidos do contorno externo, para se chegar aos pontos de segmentação candidatos. Os caminhos de segmentação são gerados através da ligação de pares de pontos de segmentação com segmentos de reta.

Depois de encontrados os caminhos de segmentação candidatos, a imagem é segmentada utilizando-se todos eles (técnica conhecida como *“segmentação em excesso”*). Os componentes conectados (CCs) obtidos após a segmentação, são então colocados em uma lista ordenada de acordo com o posicionamento dos mesmos em relação à coordenada x da imagem original. Tendo esta lista formada, é então criado um grafo com as possibilidades de segmentação da imagem,

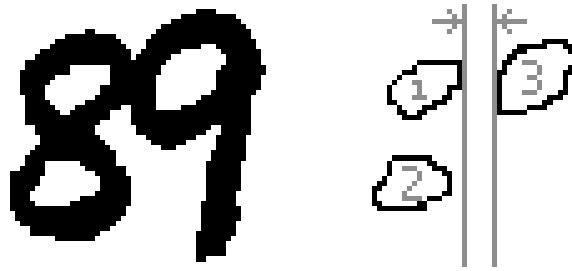


Figura 7: Linhas verticais (em cinza) mostrando a distância horizontal dos ciclos (Imagem original à esquerda).

podendo ter desde todos os CCs correspondendo à elementos isolados, até vários componentes consecutivos (segundo a ordem anterior) constituindo um só componente. Para determinar qual o caminho do grafo de hipóteses representa a melhor segmentação que pode ser obtida com base nos pontos gerados, é utilizada uma função de classificação do CC. Esta função adota três comparações de medidas de tamanho dos CCs em relação à alguns limiares. Cada comparação pode retornar três tipos de resultado: Componente precisa ser agrupado; componente precisa ser segmentado ou componente indefinido. Então se realiza uma “votação” sobre os resultados fornecidos por cada uma das comparações. O resultado mais votado é utilizado como sendo o ideal. Os autores propuseram também uma segunda maneira de realizar o teste, tentando reconhecer cada um dos componentes do grafo de hipóteses. Depois, de acordo com os escores de reconhecimento, avalia-se qual dos caminhos possui maior probabilidade de reconhecimento.

A avaliação do algoritmo foi realizada sobre base de dados própria do autor, contendo 46 classes, sendo essas as que mais aparecem nos dígitos conectados obtidos pelos autores com auxílio de um formulário. Para cada uma das classes foram utilizados 20 exemplos. Contabilizando assim 920 imagens de teste. Vale lembrar que esta base de dados não possui casos de conexão múltipla. O resultado obtido foi uma taxa de acerto de 95% com 5% de rejeição.

Como limitações deste método podemos citar os casos onde o algoritmo falha ao determinar a largura do traço vertical. Outra limitação que deve ser destacada é que o algoritmo não consegue segmentar casos de conexão múltipla (dígitos que se tocam em mais de um ponto) nem cadeias numéricas contendo três ou mais dígitos conectados. Um ponto negativo do algoritmo é a necessidade de determinação manual de alguns limiares, os quais poderiam necessitar de uma redefinição caso a base de dados utilizada para testar o algoritmo fosse substituída por outra.

Como ponto positivo podemos citar a ausência de algoritmos de alta complexidade e custo computacional, como esqueletização ou operações morfológicas. Podemos citar também, apesar da alta taxa de rejeição (5%), que a taxa de erro é nula para este algoritmo, o que é de extrema importância para um caso de aplicação do algoritmo na solução de problema real.

2.2 Shi e Govindaraju

Shi e Govindaraju [SG97] propõem um método que se diferencia dos tradicionais por não determinar os pontos de segmentação unicamente pelo traço que conecta os pares de dígitos, mas por identificar regiões com potencial para serem utilizadas como pontos de segmentação. Essas regiões são determinadas basicamente pela análise da trajetória do traçado dos dígitos. A tarefa de busca do ponto de segmentação ou do “traço ligador” não pode ser considerada trivial, principalmente devido à largura do traço não ser uniforme. A esqueletização seria uma boa solução para este problema, entretanto não foi utilizada devido ao alto custo computacional deste tipo de algoritmo.

Pode ser observado na literatura que métodos heurísticos são comumente utilizados, porém apesar da eficiência, sua precisão é limitada. Histogramas verticais também são amplamente utilizados, porém estes são muito propensos a erro. O método proposto utiliza informações do traço, como direção, pontos de curvatura e pontos finais. Estas informações são obtidas da representação de contorno em forma de cadeia de códigos.

Essa cadeia de códigos que representa o contorno da imagem, além das coordenadas do ponto, a inclinação e a curvatura, também possui o tipo do ponto, o ponto oposto correspondente e outro ponto calculado a partir do ponto oposto, o qual é utilizado para calcular a largura do traço e para a reconstrução das imagens dos caracteres segmentados.

Os autores puderam observar através de um estudo dos pontos de toque e dos “traços ligadores” entre dois dígitos, que o contorno faz curvaturas significantes à direita (fig. 8), em cada ponto de toque. Para determinar se uma curvatura é ou não significante, calcula-se um limiar, utilizando para isso a espessura do traço e o vetor normal.

A perseguição do traço é uma alternativa atraente, porém complicada. Por este motivo, apesar de já existirem métodos que utilizam esta abordagem, seu potencial ainda é pouco utilizado.

Alguns métodos da literatura se baseiam em reconhecimento, porém chamadas freqüentes ao classificador tornam estes métodos ineficientes.



Figura 8: (a) Imagem Original, (b) Pontos de curvatura à direita, (c) Dígitos segmentados

Os pontos significantes de curvatura à direita, junto com seus pontos opostos correspondentes, dividem o contorno em pedaços. Estes pedaços são então classificados como pertencentes ao dígito da esquerda ou da direita. A maneira mais fácil de realizar a classificação é utilizando uma linha de decisão. Esta é uma vertical, centralizada na imagem, a qual divide a imagem em

duas partes de igual tamanho. Então se o centro de massa de determinado pedaço do contorno se encontra à esquerda da linha, este pertence ao dígito da esquerda, senão ao da direita. Porém este método falha se os dígitos possuem uma diferença muito grande de largura. Devido a este motivo outra maneira de determinar a linha de decisão é proposta, a qual utiliza o histograma vertical da imagem.

Na figura 8 podemos visualizar a imagem original, os pontos de curvatura encontrados e o resultado da segmentação.

O método proposto foi testado sobre uma base contendo 1966 imagens do CD-ROM do CEDAR, base esta que contém pares de dígitos conectados. Os autores exploram apenas a segmentação de pares de dígitos conectados, não tratando cadeias de três ou mais dígitos conectados. O método obteve 78% de acerto na segmentação. Se para os casos onde existe um “traço ligador” for adicionado um classificador de cadeias de contorno, para auxiliar na tomada de decisão da segmentação, então a taxa de acerto obtida é de 80%.

2.3 Oliveira et al

Oliveira et al [OLBS00] propõem um novo método de segmentação para dígitos manuscritos, baseado no método proposto por Fenrich [Fen91]. O método utiliza uma combinação de dois tipos de características estruturais e foi desenvolvido para ser utilizado em um sistema de segmentação baseado em reconhecimento. O algoritmo recebe como entrada uma imagem binária dos dígitos. Como resultado o algoritmo fornece uma lista com os melhores caminhos de segmentação e o número de dígitos a serem segmentados.

O contorno e o perfil encontram-se entre as características mais comumente encontradas na literatura. São então extraídos os mínimos locais do contorno (fig. 9a) e do perfil da imagem (fig. 9b). Porém essas características não são suficientes para localizar qualquer tipo de segmentação, por exemplo, possuem grande chance de falhar em casos onde os dígitos encontram-se inclinados. Por esse motivo os autores optaram por adicionar ao método proposto por Fenrich [Fen91], uma nova característica baseada em pontos de intersecção encontrados no esqueleto da imagem (fig. 9c).

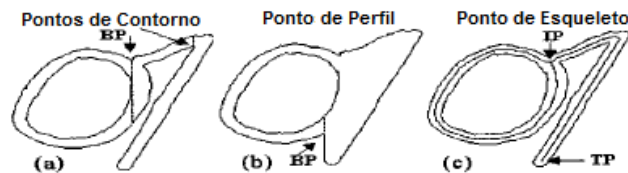


Figura 9: Pontos de (a) contorno, (b) perfil e (c) esqueleto. BP (pontos básicos), IP (ponto de intersecção) e TP (ponto terminal).

O algoritmo busca relações entre todos os pontos encontrados na tentativa de agrupá-los de modo que sejam formadas várias hipóteses de segmentação. Para avaliar as várias hipóteses, foi utilizada uma rede neural treinada com uma base de dados contendo 8500 imagens de dígitos

manuscritos isolados. O melhor experimento de aprendizagem resultou em uma taxa de 99,9% no treinamento e 98,5% no teste.

O sistema proposto foi então testado sobre 900 imagens de dígitos conectados, extraídas de 2000 imagens de cheques bancários brasileiros. Vale lembrar que o algoritmo só se aplica a pares de dígitos conectados. A escolha da melhor hipótese de segmentação foi feita pelo maior valor de produto dos escores dos elementos segmentados que compõem a hipótese. Estes escores são atribuídos pela rede neural. Em uma primeira análise visual dos resultados conclui-se que 99,5% dos dígitos foram corretamente segmentados. Em um segundo momento, o algoritmo foi avaliado com o uso de uma rede neural para classificar os dígitos segmentados. Nesta segunda avaliação, a taxa de acerto foi de 90,8%. Dentre o percentual de erro de 9,2% verificou-se que 1,5% era erro de segmentação, e 7,7% era erro de reconhecimento. Esse erro referente à segmentação pode ser diminuído se a rede neural for treinada com dígitos provenientes da saída de um algoritmo de segmentação, e não apenas dígitos naturalmente isolados.

A técnica de segmentação desenvolvida nesse artigo possui um algoritmo simples e com poucas regras. O algoritmo atingiu bons resultados e realizou uma boa segmentação na maioria dos dígitos conectados, mesmo para aqueles em que os dígitos estavam sobrepostos ou inclinados. Testes realizados por Oliveira et al [OBS05], com o uso de uma base sintética de dígitos conectados, mostram que a adição das características de esqueleto conduziram o método a atingir um desempenho 2,5% maior que o método original proposto por Fenrich [Fen91].

2.4 Chen e Wang

Chen e Wang [CW00] propõem uma nova abordagem para segmentação de cadeias de 2 dígitos conectados. Esta abordagem visa à segmentação tanto de pares de dígitos simplesmente conectados, quanto dígitos com múltiplas conexões. O método combina características tanto do primeiro plano da imagem (dígito propriamente dito) como do plano de fundo. Os pontos de segmentação são obtidos com a análise dos esqueletos da imagem, esqueleto dos dígitos e do fundo. Vários caminhos de segmentação são gerados e os “traços-ligadores” removidos. Finalmente os parâmetros da função de probabilidade geométrica de cada caminho de segmentação são determinados e analisados utilizando uma função de mistura de probabilidade gaussiana, para escolher o melhor caminho de segmentação ou rejeitar a imagem.

Grande parte dos métodos encontrados na literatura apresenta dificuldade para segmentar dígitos com conexão múltipla ou dígitos com conexão simples que possuem uma grande área de toque. Segundo os autores este método conduz a bons resultados nesses casos, se comparado com métodos já existentes.

As conexões entre dígitos são classificadas pelos autores em cinco tipos (fig. 10). De modo geral, os algoritmos que trabalham com o plano de fundo ou plano principal, ambos isoladamente, segmentam com sucesso conexões dos tipos 1 e 2. Porém apresentam dificuldade com conexões dos tipos 3, 4 e 5. Por isso neste método, características de ambos os planos são extraídas, pois se espera com isso, melhorar o desempenho em relação aos demais métodos que

utilizam apenas um dos tipos de característica.

Categoria	Tipo	Tipo de Conexão	Exemplo
Conexão simples	1		59 33
	2		24 02
	3		23 52
	4		40 00
Conexão múltipla	5		78 38

Figura 10: 5 possíveis tipos de conexão, segundo Chen et al [CW00].

Os pontos de característica extraídos dos esqueletos da imagem são: pontos de bifurcação, de curvatura e terminais. O algoritmo combina então essas características extraídas tanto dos dígitos como do fundo, para gerar os possíveis caminhos de segmentação.

É feita uma classificação dos caminhos de segmentação, utilizando-se uma função de mistura de probabilidades Gaussianas. Esta função foi obtida com base em 823 imagens da base NIST SD19. Então se a probabilidade Gaussiana do caminho de segmentação melhor classificado for maior que um valor pré-determinado, este é tido como o melhor caminho de segmentação. Caso contrário, este caminho é rejeitado.

Após a identificação de todos os possíveis caminhos de segmentação é feita a remoção dos “traços-ligadores” quando os mesmos existem. Esses traços são caracterizados por dois caminhos de segmentação com mesmo início e fim, além disso, algumas heurísticas devem ser satisfeitas para que esse tipo de traço seja realmente caracterizado. A figura 11 ilustra uma imagem contendo “traço-ligador” (fig. 11a) , sua identificação através dos caminhos de segmentação com início e fim em comum (fig. 11b) e a imagem final após removido o traço (fig. 11c).

Para testar o desempenho do método foram utilizadas 4500 cadeias de dois dígitos, destas, 4178 foram extraídas da base NIST SD 19, as 322 imagens restantes foram coletadas pelos próprios autores. No experimento foram separadas 832 imagens para determinar a função de mistura de probabilidades Gaussianas, e o restante utilizado para testar o algoritmo de segmentação. Das imagens utilizadas no teste, 8% foram rejeitadas, e se considerarmos somente as imagens aceitas, é obtida uma taxa de acerto de 95,7%. As principais razões causadoras de rejeição foram: quando o centro do caminho de segmentação está muito deslocado em relação ao centro da imagem ou quando a largura de um dos dígitos separados é muito maior que do outro.

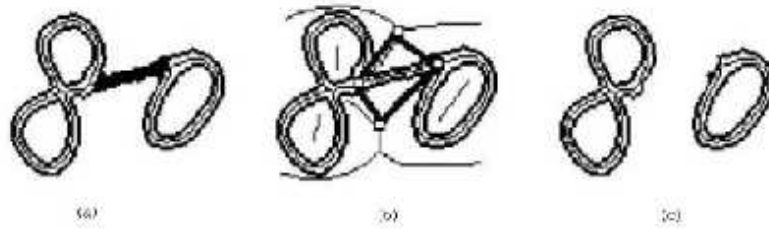


Figura 11: (a) Imagem contendo “traço-ligador”, (b) identificação do traço, (c) imagem após a remoção do mesmo.

Como pontos positivos do método podemos citar o bom desempenho em alguns casos específicos que dificilmente são segmentados por outros métodos presentes na literatura, como por exemplo, conexões onde a área de toque é muito grande, ou imagens com mais de dois pontos de conexão. Além da boa capacidade de remoção dos “traços-ligadores”. O principal ponto negativo do método é o número muito alto de caminhos de segmentação gerados, os quais aumentam bastante o tempo de processamento e impossibilitam o uso do mesmo em aplicações reais.

2.5 Yu e Yan

Yu e Yan [YY01] desenvolveram um método capaz de segmentar cadeias de dígitos, ou seja, cadeias que podem possuir mais de dois dígitos. O método funciona segmentando a cadeia em várias sub-cadeias de dois dígitos e processando cada uma delas separadamente.

O método utiliza características morfológicas estruturais. São obtidos os pontos de características estruturais na imagem de bordas suavizada. Um ponto estrutural é o ponto que define uma mudança morfológica no contorno. São definidos 16 diferentes padrões de pontos estruturais (fig. 12). A imagem é dividida em quatro regiões. Durante a análise dos pontos estruturais é levado em conta além do padrão do ponto, a região que este se encontra. Também são extraídas características das concavidades da imagem. Com base nestas características foram construídos diversos modelos, os quais são representados por sequências de pontos estruturais. Então com esses modelos são construídas as diversas heurísticas.

Para construir os modelos foram utilizadas 600 imagens da base NIST. O conjunto de teste era constituído de 3287 imagens de cadeias constituídas de dois dígitos, para as quais, a taxa de segmentação alcançada foi de 94,8%. Para as 256 imagens restantes, as quais eram constituídas de três dígitos, a taxa de segmentação correta foi de 84,7%. Para os testes de desempenho foi utilizado um knn otimizado, de modo que se ambos os dígitos segmentados são reconhecidos por este knn, então a segmentação é considerada correta. Os autores enfatizam que podem existir casos onde uma boa segmentação possa ter sido classificada como incorreta devido a um erro de classificação do dígito por parte do knn.

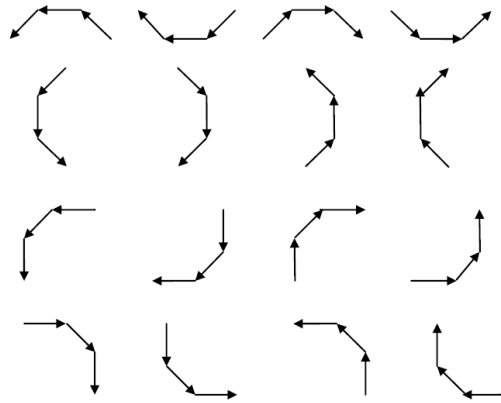


Figura 12: Diferentes padrões estruturais que representam as possíveis mudanças morfológicas no contorno da imagem.

2.6 Pal et al

Pal et al [PB03] propõem um algoritmo de segmentação livre de reconhecimento (não utiliza um classificador para a validação dos pontos de segmentação). Uma grande vantagem desse tipo de algoritmo é o menor tempo de processamento se comparado aos baseados em reconhecimento. Uma particularidade deste método é que ele gera apenas uma hipótese de segmentação.

O algoritmo se baseia na idéia principal de que se dois dígitos se tocam, é formado entre eles um grande espaço, chamado pelos autores de “reservatório” (fig. 13). Os pontos de segmentação tendem a estar na base destes reservatórios. Mais especificamente em pontos de “obstáculos”, ou “nós”, na base. Sendo assim, temos uma redução da área de busca dos pontos, o que contribui para um menor esforço computacional. Como outros pontos positivos deste método, podemos apontar o funcionamento independente da inclinação da imagem, suporte a conexões simples e múltiplas e a não utilização de algoritmos de alto custo computacional, como por exemplo, esqueletização da imagem. Com este algoritmo também é possível classificar as imagens pela região de conexão de acordo com o tipo de reservatório encontrado.

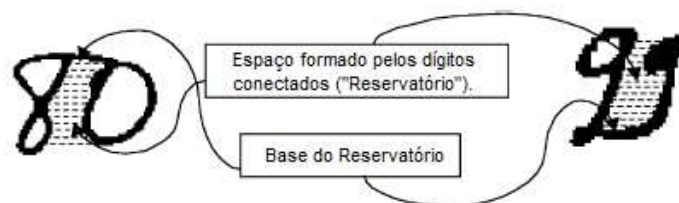


Figura 13: Espaços (“Reservatórios”) criados pela conexão entre os dígitos, e suas respectivas bases.

O algoritmo utiliza como características a quantidade de reservatórios, suas posições

em relação à *Bounding Box*, seus tamanhos e formas, centro de gravidade, posição relativa e padrão morfológico da área de conexão. Também são utilizadas características extraídas dos *loops* (ciclos fechados), são elas: posição, centro de gravidade e a razão $\frac{AlturadoLoop}{AlturadoComponente}$.

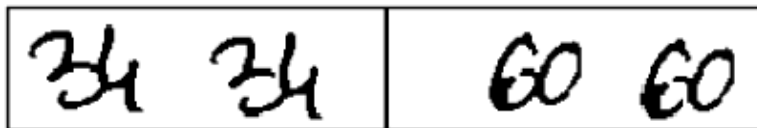


Figura 14: Exemplos de segmentação incorreta gerada pelo método proposto.

Um diferencial deste algoritmo é que antes de tentar segmentar ele classifica os dígitos da imagem em isolados ou conectados. Esta classificação possui uma taxa de acerto de 98,81%. Como problemas deste algoritmo temos o fato de não conseguir segmentar imagens que contenham falha no contorno do dígito em um ponto que constitua a parede de um reservatório e também o fato de suportar apenas cadeias de 2 dígitos. Dois exemplos de segmentação incorreta deste algoritmo podem ser observados na figura 14. Os autores testaram o método sobre uma base de dados contendo 2250 pares de dígitos conectados extraídos de cheques bancários franceses. Neste teste, o algoritmo alcançou 94,8% de acerto com uma taxa de rejeição de 3,4%.

2.7 Elnagar e Alhajj

Elnagar e Alhajj [EA03] propõem um método para segmentação de pares de dígitos simplesmente conectados. O método utiliza características de fundo e contorno juntamente com algumas heurísticas, estas utilizadas para determinar os pontos de características com maior potencial.



Figura 15: Imagens das fases do método de segmentação: imagem original, imagem esqueletizada, imagem segmentada e imagem restaurada.

Antes da extração dos pontos de características a imagem é normalizada quanto à inclinação, linha de base e tamanho. Quanto ao tamanho, a imagem é redimensionada para 30 x 60 pixels, independente do fato de ter dimensões originais maiores ou menores que estas. Após este processamento inicial a imagem é então esqueletizada (fig. 15). O processo de esqueletização da imagem é caro computacionalmente, porém, os autores justificam seu uso argumentando que de posse de uma imagem com traços uniformes (neste caso, traço com um pixel de espessura), a extração de características torna-se mais simples.

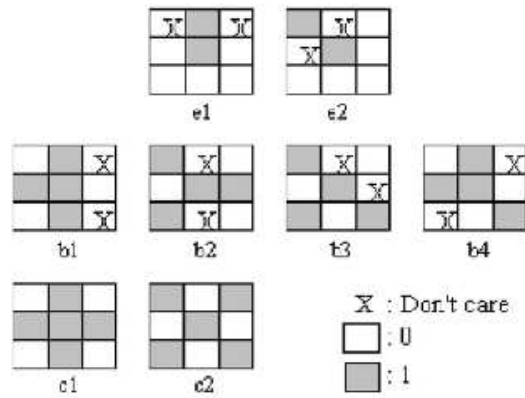


Figura 16: Modelos utilizados na extração de características. Pontos Finais (e1;e2), pontos de ramificação (b1;b2;b3;b4) e pontos de cruzamento (c1;c2)

Para a extração de características são utilizadas máscaras de tamanho 3x3 (fig. 16). Estas máscaras são deslizadas sobre a imagem em todas as suas rotações múltiplas de $\frac{\pi}{2}$. Desta forma são então obtidos os pontos finais, pontos de ramificação e de cruzamento. Outro ponto é encontrado obtendo-se o ponto mais alto do histograma do eixo y. Depois de encontradas as características, é então utilizado um processo de remoção de ruído, também baseado em uma máscara 3x3. Este processo visa à remoção dos pontos de características redundantes.

De modo análogo à Suwa et al [SN04], as conexões são classificadas em alguns tipos. Os quais são representados pelas figuras 17, 18, 19 e 20.



Figura 17: Conexões com ponto em comum.

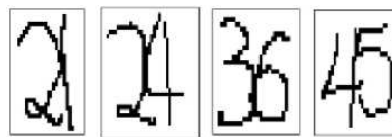


Figura 18: Conexões com segmento em comum.

Os pontos de segmentação são então escolhidos de acordo com algumas heurísticas que basicamente analisam os posicionamentos dos pontos de características encontrados com relação a mais alta colina e o mais profundo vale encontrado na imagem (fig. 21).

No caso específico de conexões suaves (fig. 19), o algoritmo não consegue encontrar os pontos de características e conseqüentemente não determina nenhum caminho de segmentação.

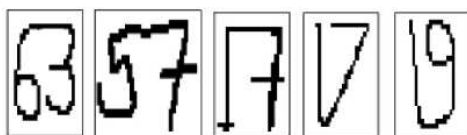


Figura 19: Conexões suaves.



Figura 20: Conexões com “traço-ligador”

Portanto podemos considerar este fato como uma limitação deste método.



Figura 21: Na esquerda temos uma imagem exemplificando um ponto de colina e na direita um ponto de vale. (ambos os pontos estão marcados com um “X” na imagem)

Embora tenha sido desenvolvido para tratar de conexões simples, os autores afirmam que em casos onde as conexões que compõem a conexão múltipla se encontram bem próximas, o método é capaz de realizar a segmentação.

O método foi testado sobre uma base de dados composta por imagens do CD-ROM do CEDAR, base NIST 19 e imagens de propriedade dos autores. Os testes resultaram em uma taxa de acerto de 96%.

2.8 Lei et al

Lei et al [LLDF04] propõem um método baseado no conceito de sobre-segmentação e reconhecimento. Método este que realiza um pré-processamento para normalização do tamanho, suavização do contorno e uma operação morfológica de fechamento, esta visando resolver o problema de pedaços quebrados e ruídos. Antes de iniciar a segmentação da imagem, o método decide se a mesma é de dígito isolado ou conectado, assim como em Pal et al [PB03], através de um limiar aplicado ao resultado de um classificador de dígitos isolados.

Para identificar os possíveis pontos de segmentação é realizada uma análise do contorno interno e externo, divididos em superior e inferior (fig. 22), além de projeções de histograma. O

resultado ótimo da segmentação é determinado pela máxima probabilidade de reconhecimento. Visando a diminuição do número de caminhos de segmentação a serem testados é utilizada uma condição de poda, tendo assim uma conseqüente redução do tempo de processamento.

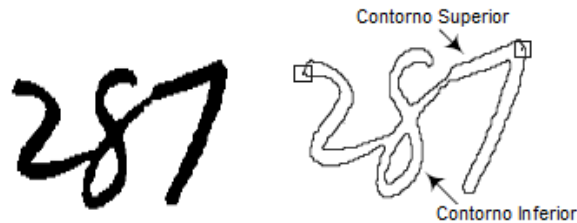


Figura 22: Contorno inferior e superior de uma cadeia de dígitos conectados.

Este método foi testado pelos autores em 3359 exemplos da base NIST SD19, e atingiu uma taxa de acerto de 97,72% com 0% de rejeição, em cadeias de 2 dígitos. Para cadeias de 3 dígitos o acerto foi de 93,33%, para os 525 exemplos testados. Vale ressaltar que as imagens utilizadas sempre possuíam os 3 dígitos conectados, o que pode-se considerar um fato de aumento da complexidade em relação a testes realizados por outros autores, os quais contém cadeias onde em alguns casos apenas 2 dígitos se tocam.

2.9 Suwa e Naoi

Suwa e Naoi [SN04] propõem um algoritmo capaz de segmentar tanto dígitos com conexão simples, quanto dígitos com conexão múltipla. O algoritmo proposto trabalha apenas com pares de dígito e segmenta inclusive casos em que há um “traço ligador” (traço que liga os dois dígitos e não pertence a nenhum dos mesmos), estes sendo considerados casos de maior dificuldade.

Antes da segmentação o algoritmo executa um pré-processamento, o qual realiza remoção de ruídos, suavização e correção de inclinação. Após isso é realizada a esqueletização da imagem (fig. 23b) e identificação dos vértices de graus 1, 3 e 4 (pontos finais, junção em T e pontos de cruzamento, respectivamente). Os vértices são unidos por arestas formando um grafo conectado (fig. 23c). Algumas arestas são eliminadas de acordo com algumas heurísticas.

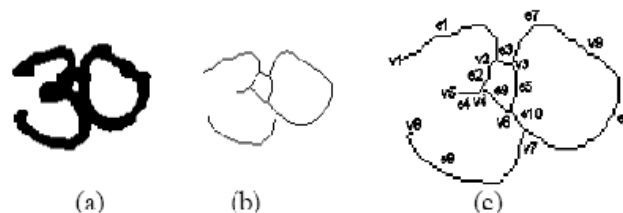


Figura 23: Par de dígitos e suas diferentes representações. (a) Imagem original, (b) Imagem do esqueleto, (c) Representação de grafo no contorno da imagem.

Para realizar a segmentação, é necessário a obtenção de 2 sub-grafos. Então para realizar

o agrupamento dos vértices/arestas que pertencem a um mesmo dígito, são utilizados conceitos de teoria dos grafos e algumas heurísticas. Segundo os autores os pontos de segmentação geralmente são encontrados próximos ao ponto mais baixo do “vale” formado no contorno superior e no mais alto da “colina” encontrada no plano inferior. Esses pontos são encontrados pelo algoritmo com ajuda de dois histogramas verticais.

Os autores classificam as conexões entre dígitos em 4 tipos (fig. 24): Conexão em um ponto, conexão em um segmento, ponto de conexão em “traço ligador” e conexão múltipla. Então o algoritmo identifica primeiramente o tipo da conexão para depois definir as heurísticas a serem utilizadas.

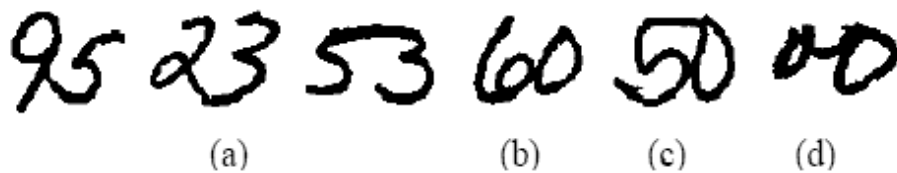


Figura 24: Tipos de conexão: (a) Conexão em um ponto, (b) Conexão em um segmento, (c) Conexão múltipla, (d) Conexão com “traço-ligador”.

O desempenho do algoritmo foi avaliado nos testes realizados sobre 2000 pares de dígitos conectados, extraídos da base NIST 19. Outras 1000 imagens procedentes da mesma base foram utilizadas para construção das heurísticas. O algoritmo atingiu nos testes um desempenho de 88,7% de acerto com taxa de rejeição de 2,45%. As imagens segmentadas produzidas pelo algoritmo possuem forma mais natural do que as imagens geradas por algoritmos que utilizam uma linha reta para separar os dígitos.

2.10 Sadri et al

Sadri et al [SSB07] propõem um método de segmentação e reconhecimento de strings numerais de tamanho variável, o qual utiliza um algoritmo genético para encontrar o melhor caminho de segmentação. Assim como em Lei et al [LLDF04], o método também é baseado no conceito de sobre-segmentação.

Antes da segmentação da imagem, a mesma passa por um pré-processamento constituído de suavização, remoção de ruídos e correção de inclinação. Na fase de segmentação o método trabalha com uma análise de componentes conectados (CCs). Desta forma os CCs podem ser classificados em três tipos: pedaços de dígitos quebrados ou “traços-ligadores”, dígitos isolados e dígitos conectados. Dentre estes, apenas o terceiro tipo necessita de segmentação. Por isso, o método realiza esta classificação analisando informação contextual (basicamente tamanho relativo do componente em relação à cadeia).

Feita a classificação dos CCs, o método então vai buscar os caminhos de segmentação para todos os CCs classificados como dígitos conectados. Para isto são utilizadas características

tanto do plano frontal (CC), como do fundo da imagem. As características do primeiro tipo (plano frontal) são os pontos de intersecção extraídos do esqueleto da imagem. Os autores propõem uma nova maneira de extrair características do plano de fundo, a qual não considera todos os pixels brancos da imagem. Primeiramente é obtida a projeção do perfil da imagem, logo após é então obtido o esqueleto desta imagem da projeção. Segundo os autores pelo fato deste método cobrir apenas a parte essencial do fundo da imagem, os pontos encontrados são mais estáveis e informativos, além de ser encontrado um menor número de características de fundo, o que propicia uma maior simplicidade e menor custo computacional para avaliar estas características. Todas as etapas do método proposto para extração das características do plano de fundo são demonstradas pela figura 25.

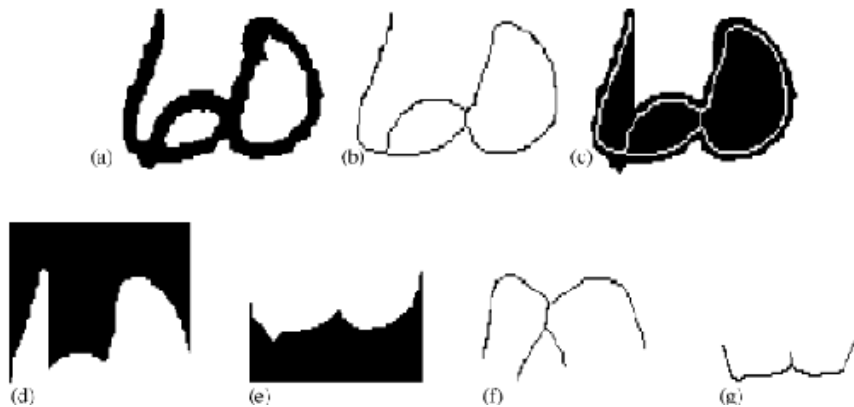


Figura 25: (a) Imagem pré-processada, (b) Esqueleto do plano frontal, (c) Plano de fundo (pixels brancos fora do objeto composto por pixels pretos), (d) projeção do perfil superior, (e) projeção do perfil inferior, (f) esqueleto superior do plano de fundo, (g) esqueleto inferior do plano de fundo.

As características encontradas são então combinadas e os possíveis caminhos de segmentação gerados. Se existirem n caminhos de segmentação, então existirão $2n$ hipóteses de segmentação para a cadeia de dígitos em questão. Nos casos onde $n \leq 5$ (até 32 hipóteses), são avaliadas todas as hipóteses através de uma busca exaustiva. Porém se $n > 5$ então é feita a busca com auxílio de um algoritmo genético, o qual nem sempre encontra a solução ótima, porém encontra uma solução próxima a esta com custo computacional menor do que da busca exaustiva.

Para a avaliação das hipóteses de segmentação é proposta a combinação de um reconhecedor de dígitos manuscritos isolados com informações contextuais, informações estas que compõem um novo conjunto de valores, chamado de escore da segmentação. Apesar de a maioria dos algoritmos encontrados na literatura utilizarem apenas um reconhecedor de dígitos isolados, optou-se por utilizar também este escore de segmentação para prevenir alguns erros comuns (fig. 26) encontrados em sistemas de reconhecimento de cadeias manuscritas. Este escore expressa o grau de confiança de que determinado CC pertence ou não a cadeia de dígitos em questão e é formado basicamente por duas partes: posição relativa e razão de aspecto.

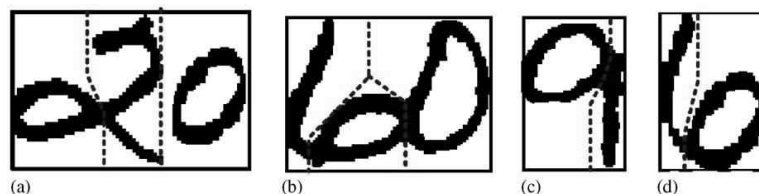


Figura 26: Casos onde provavelmente um reconhecedor de dígitos isolados cometeria erro sem a ajuda de informações de contexto. Por exemplo, (a) poderia ser facilmente reconhecido como 020, (b) como 101, (c) como 01 e (d) como 10.

Então para determinar se determinado CC é ou não válido dentro da cadeia analisa-se o mínimo entre o escore da segmentação e o escore do reconhecimento, sobre este mínimo aplica-se um limiar. Se o dígito for maior que um limiar (determinado empiricamente) então este é considerado válido.

Foram realizados dois testes do algoritmo, utilizando dois métodos de reconhecimento diferentes. Um deles utiliza uma rede neural MLP e o outro um máquina de vetores de suporte (SVM). Antes dos testes do sistema como um todo, foram realizados testes do módulo de segmentação, o qual, sobre uma base de dados contendo 5000 imagens de pares de dígitos conectados da base NIST, obteve uma taxa de acerto de 96,5%. Já para as 1800 imagens extraídas da base NIST NSTRING SD19, imagens estas de cadeias de tamanho 2, 3, 4, 5, 6 e 10, a taxa de acerto foi de 98,04%. Chegou-se a estes números através de avaliação visual dos resultados da segmentação. Porém estes números não podem ser comparados aos métodos encontrados na literatura, pois os autores consideraram como acerto todos os casos onde todos os caminhos de segmentação necessários estavam presentes dentre os vários caminhos encontrados pela sobre-segmentação da imagem. Ou seja, estas taxas de acerto podem diminuir caso o algoritmo que decide quais caminhos de segmentação dentre os encontrados deve utilizar, não seja livre de erros.

Destes primeiros testes foram extraídos os CCs que não constituem um dígito, para auxiliar na determinação de alguns parâmetros do cálculo do escore de segmentação e também foram utilizados no treinamento da rede neural e do SVM.

Os testes finais foram então feitos, a versão utilizando MLP obteve uma taxa de acerto (segmentação e reconhecimento) de 95,28% e a que utilizou SVM alcançou 96,42%. Os testes também mostraram que houve uma melhora de 7,85% para o MLP e 6,35% para o SVM, referente à utilização do escore de segmentação.

Podemos considerar como contribuições significativas deste método o uso do escore de segmentação, o qual trouxe melhoria considerável no desempenho do algoritmo. O método diferenciado de análise do plano de fundo, o qual analisa apenas os pixels considerados essenciais, reduzindo assim a quantidade de pixels a serem analisados e obtendo com isso um menor custo computacional. Devemos citar também o bom desempenho alcançado na segmentação de conexões que possuem “traços-ligadores”, os quais são responsáveis por boa parte dos erros em

grande parte dos métodos encontrados na literatura. O uso de algoritmo genético também é um diferencial do método e é o principal ponto comentado no artigo, porém não demos destaque aos detalhes do AG, pois seu uso não faz sentido na segmentação de cadeias numéricas contendo apenas dois dígitos, caso este que é o foco do nosso projeto.

2.11 Comparação dos Métodos

Tabela 1: Tabela Comparativa dos Métodos.

Métodos	Base de Dados	Quantidade	Acerto	Rejeição
Fujisawa et al	Imagens próprias	920	95,0%	5,0%
Shi e Govindaraju	CEDAR CD-Rom	1966	80%	-
Oliveira et al	Cheques Brasileiros	900	99,5%	0%
Cheng e Wang	NIST/próprias	4500	96%	7,8%
Yu e Yan	NIST SD19	3287	94,8%	-
Pal et al	Cheques Franceses	2250	94,8%	3,4%
Elnagar e Alhajajj	NIST/CEDAR/próprias	Não informada	96%	-
Lei et al	NIST SD19	3359	97,72%	0%
Suwa e Naoi	NIST SD19	2000	88,7%	2,45%
Sadri et al	NIST SD19	5000	96,5%	-

A tabela (tab. 1) exibida acima estabelece uma comparação entre os métodos descritos nesta seção. Os desempenhos exibidos são os reportados pelos autores. Lembrando que não podemos tirar conclusões concretas apenas olhando para a tabela abaixo, devido principalmente a dois fatores. O primeiro é que os métodos não foram testados com as mesmas imagens, então os métodos podem ter sido submetidos à imagens com diferentes graus de dificuldade. O outro fator é que as formas de avaliação também não foram as mesmas, alguns foram avaliados visualmente, outros com redes MLP e outros ainda com SVM. Portanto apesar desta tabela ser uma forma de comparação, não é possível tirar conclusões justas somente com base nela. Esperamos ao fim deste projeto, poder realizar uma comparação mais precisa.

3 Método Proposto

Devido ao fato de existirem diversos métodos para segmentação de dígitos manuscritos e mesmo vários deles alcançando bom desempenho, nenhum chega a atingir o desempenho necessário em todos os tipos de conexão, para que possa ser utilizado na solução de problemas reais.

Este projeto tem como objetivo realizar uma análise detalhada de alguns dos principais algoritmos de segmentação encontrados na literatura, visando definir quais são os casos de sucesso e insucesso de cada um. Desta forma podemos saber quais métodos obtém melhores resultados para cada um dos tipos de conexão (a descrição dos tipos de conexão utilizados encontra-se nos parágrafos a seguir).

De posse dos resultados das análises propostas, seria possível estudar a construção de um método que possa combinar algoritmos de segmentação e utilizá-los com o conceito de voto. Poderíamos também, decidir quais algoritmos participariam da votação, de acordo com o tipo de conexão encontrado na imagem de entrada. Com um método como este que estamos vislumbrando, seria possível melhorar o desempenho obtido na segmentação através do aproveitamento dos pontos positivos de cada algoritmo descartando seu uso nos casos onde é sabido que seu desempenho é baixo.

Optamos por focar nosso trabalho na segmentação de pares de dígitos conectados. Esta decisão foi baseada nos dados mostrados por Wang et al [WGS00], onde os autores analisaram uma base de dados contendo 200.000 cadeias de dígitos correspondentes à códigos de endereçamento postal, números de rua, números de apartamentos e números de caixas postais. Com esta análise concluíram que 85% dos casos de conexão encontrados correspondem à pares de dígitos conectados.

Para realizar a avaliação dos algoritmos, será utilizada uma base sintética de dígitos conectados proposta por Oliveira et al [OBS05]. Dizemos que é uma base “sintética”, pois a mesma foi gerada automaticamente com base em imagens de dígitos isolados da base NIST. O algoritmo utilizado para a construção desta base seleciona 2 dígitos isolados de mesmo escritor (a informação de escritor está disponível na base NIST) e os aproxima até que aconteça um toque. A base contém 273.452 pares de dígitos conectados, alguns exemplos podem ser vistos na figura 27.

Neste trabalho utilizaremos uma parte desta base de dados, o que corresponde a aproximadamente 79.500 imagens (não será utilizada a base de dados inteira, pois não julgamos necessária quantidade tão grande de imagens). Todas as imagens desta base possuem como informação adicional as coordenadas dos pontos de conexão, facilitando assim a posterior avaliação do desempenho dos algoritmos de segmentação.

Para que seja possível a análise dos casos de acerto e erro de cada algoritmo de segmentação de acordo com o tipo de conexão existente na imagem, precisamos classificar a base de dados de acordo com o tipo de conexão. Para isso nos baseamos na classificação proposta por Chen e Wang [CW00] (fig. 10). Porém, devido ao fato de nossa base de dados possuir



Figura 27: Exemplos de imagens geradas pelo algoritmo de geração automática

pouquíssimos casos de conexão por traço-ligador (tipo 4 na classificação de Cheng e Wang, optamos por não considerar este como um tipo de classificação e classificar as imagens que pertenceriam a este tipo, como sendo do tipo 1 ou 3, dependendo do caso. Um exemplo dessa classificação alternativa para os casos de dígitos conectados por traço ligador é mostrado na figura 28.



Figura 28: Exemplo de imagem contendo traço-ligador, classificada alternativamente como tipo 1.

Realizamos a classificação da base de dados de acordo com os 4 possíveis tipos (fig. 29), em duas etapas. Uma primeira etapa automática classifica as imagens uma a uma de acordo com a localização do ponto de segmentação. Essa localização podendo ser: superior, central ou inferior. Para isso foram mapeadas manualmente cada uma das 100 classes (00 à 99), informando assim ao algoritmo qual tipo de conexão deveria ser atribuído para cada uma das possíveis posições do ponto de conexão em cada classe de dígitos. Por exemplo, de modo geral uma conexão de classe 02 é do tipo 1 quando a conexão acontece na parte superior da imagem, e de tipo 2 quando acontece na parte central ou inferior da imagem.

Porém apesar desta classificação automática realizada ter classificado corretamente a maioria das imagens, muitas não se encaixam nessas regras básicas baseadas na posição do ponto de toque e foram erroneamente classificadas. Para resolver este tipo de problema foi





Categoria	Tipo	Tipo de Conexão	Exemplo
Conexão simples	1		59 33
	2		24 02
	3		23 52
Conexão múltipla	4		78 38

Figura 29: Tipos de conexão utilizados para a classificação da nossa base de dados.

realizada então uma segunda etapa. Etapa esta manual, na qual todas as imagens passaram por uma avaliação visual para correção dos casos de classificação incorreta. Nesta segunda etapa também foram removidas da base de dados algumas imagens que não condizem com a realidade, ou seja, imagens com conexões que nunca seriam vistas em casos naturais (fig.30), mas que aparecem na base devido ao fato das imagens terem sido geradas artificialmente. Também foram removidas da base de dados, todas as imagens pertencentes à classe 11, pois conexão entre dois dígitos “1” só existe quando o mesmo é escrito “com perna” (fig. 31), o qual não é o caso dos dígitos “1” presentes em nossa base.

Após esta remoção das imagens indesejadas, obtivemos uma base de dados contendo 79.466 imagens. A distribuição das imagens de acordo com o tipo de conexão pode ser visualizado no gráfico presente na figura 32. Podemos notar que os tipos 2 (53,6%) e 1 (34,8%) são os mais presentes em nossa base.



Figura 30: Exemplos de imagens removidas da base de dados. Classe 16 e 10 respectivamente.



Figura 31: Dígito “1”, com e sem “perna”, respectivamente.

Após implementados os algoritmos de segmentação, precisamos avaliar cada um deles afim de obter a taxa de acerto dos mesmos. Para esta avaliação, cada um dos algoritmos vai ser testado para todas as 79.466 imagens presentes em nossa base de dados.

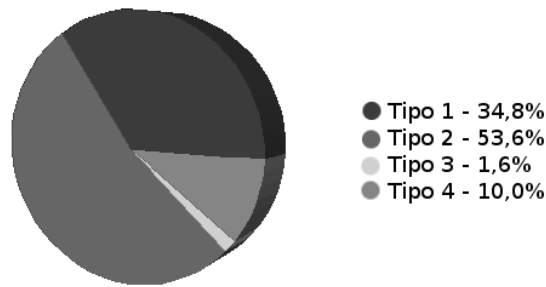


Figura 32: Gráfico da distribuição da Base de Dados de acordo com os tipos.

Como para cada imagem da base, possuímos as coordenadas dos pontos ótimos de segmentação, parece óbvio que a verificação dos pontos de segmentação gerados pelos algoritmos seja feita através de uma comparação com os pontos ótimos presentes na base de dados. Porém, dependendo da forma que esta comparação seja feita, podemos enfrentar alguns problemas. Por exemplo, digamos que para uma determinada imagem, os pontos fornecidos pela base de dados como pontos ótimos de segmentação, sejam em número de 6 e que um determinado algoritmo encontre para esta mesma imagem, apenas 2 pontos de segmentação. Apesar do algoritmo ter gerado apenas $\frac{1}{3}$ dos pontos de segmentação ideais, não podemos afirmar que o algoritmo não obteve sucesso na segmentação. Esta situação é representada pela figura 33. Percebemos então que apesar das duas hipóteses de segmentação mostradas por esta figura possuírem pontos de segmentação muito parecidos, elas conduziram a resultados diferentes, a primeira conduzindo ao resultado “38” e a segunda ao resultado “33”, que neste caso é o correto.

Um detalhe que deve ser percebido é que a hipótese de segmentação que conduziria ao erro possui seus pontos de segmentação mais próximos dos pontos ótimos do que a outra hipótese que obteve um resultado correto.

Para solucionar este problema da comparação dos pontos encontrados por cada algoritmo, com os pontos ótimos de segmentação, utilizaremos um método que além da comparação do posicionamento dos pontos de segmentação encontrados, avalia também a quantidade de cortes realizados. Onde estes “cortes” serão analisados através da contagem de transições branco/preto que ocorrem no(s) caminho(s) de segmentação gerado(s). Desta maneira, se analisarmos novamente a imagem 33, percebemos que tanto com os caminhos ótimos de segmentação como no caminho obtido na segunda hipótese de segmentação, teríamos três cortes, enquanto na primeira hipótese teríamos apenas um corte. Dessa maneira podemos dizer que a primeira hipótese não segmenta corretamente o par de dígitos, pois o número de cortes difere do número obtido com os caminhos ótimos de segmentação.

Para termos certeza de que não teremos qualquer tipo de erro na avaliação dos pontos encontrados pelos algoritmos, sendo estes decorrentes de quaisquer outros casos que possam configurar exceções e não sejam corretamente classificados pela regra descrita nos parágrafos anteriores, avaliaremos também os pontos encontrados com o uso de uma rede neural.

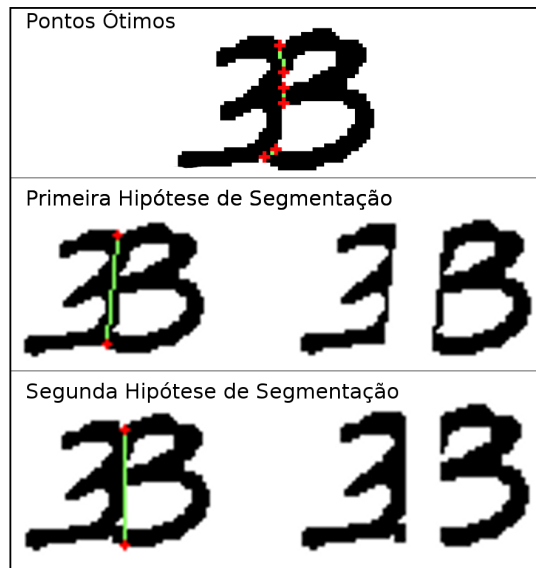


Figura 33: Imagens da classe 33, seus caminhos ótimos de segmentação e duas hipóteses. Sendo a primeira hipótese incorreta e a segunda correta.

Para avaliar os pontos de segmentação encontrados por cada algoritmo, utilizando a rede neural, verificaremos a quantidade de pontos de segmentação encontrados. Caso os pontos encontrados pelo algoritmo em questão estejam em maior quantidade do que os pontos ótimos de segmentação, os primeiros serão comparados com os pontos ótimos. Desta maneira os pontos encontrados, mais próximos aos pontos ótimos de segmentação serão mantidos, e os mais distantes, descartados. Caso o algoritmo encontre menor número de pontos de segmentação do que os pontos ótimos, então esta comparação não é realizada e todos os pontos encontrados serão utilizados.

Para calcular a distância entre os pontos encontrados e os pontos ótimos, será utilizada a equação da distância euclidiana (eq. 1), a qual resultará na distância em pixels entre os dois pontos. Sendo x_1 e y_1 as coordenadas x e y de um dos pontos e x_2 e y_2 as coordenadas x e y do outro ponto.

$$Dist.Euclidiana = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (1)$$

Selecionados os pontos à serem utilizados, segmenta-se a imagem. Obtidas as duas imagens supostamente uma correspondendo ao dígito da esquerda e outra contendo o dígito da direita. Então essas imagens obtidas são fornecidas como entrada para o classificador e o resultado fornecido pelo mesmo é comparado com a informação real dos dígitos (também fornecida pela base de dados). Nos casos onde os dois dígitos forem reconhecidos corretamente pela rede neural, dizemos que o algoritmo de segmentação obteve sucesso. Todos os demais casos são considerados casos de erro de segmentação.

O classificador citado no parágrafo anterior foi proposto por Oliveira [Oli03], o mesmo se baseia em uma rede neural MLP. Vale lembrar que durante a criação da base de dados sintética, só foram utilizados dígitos isolados que foram classificados com sucesso por este classificador. Este fato incrementa o nível de confiança que teremos no resultado dos testes que serão realizados sobre todos os algoritmos que serão implementados. Isso se deve ao fato de sabermos que caso o algoritmo segmente a imagem com um caminho ótimo de segmentação, ela será com certeza classificada corretamente pelo classificador. Com isso saberemos que imagens que forem consideradas erroneamente segmentadas, dificilmente o foram devido à erro de classificação (problema frequentemente encontrado nas avaliações de desempenho de métodos existentes na literatura).

Utilizaremos então, para todos os métodos implementados, as duas formas de avaliação dos caminhos de segmentação. Desta maneira teremos uma análise mais precisa dos resultados obtidos. Apenas após utilizarmos as duas maneiras de avaliação e verificarmos os resultados alcançados de maneira visual, poderemos dizer qual delas é mais indicada, ou até mesmo utilizarmos uma combinação das duas avaliações. A princípio pretendemos utilizar as duas maneiras para todos os algoritmos e apresentar os resultados de cada algoritmo com cada uma das avaliações.

Todos os algoritmos citados na seção 2 serão implementados e seus resultados comparados em relação ao desempenho obtido na taxa de acerto, tanto com o uso da comparação direta dos pontos como com o uso do classificador (rede neural). Os algoritmos também serão comparados com relação à quantidade de pontos de segmentação encontrados, a qual é um dos fatores utilizados para determinar o custo computacional dos algoritmos.

4 Resultados

Nesta seção são descritos os resultados esperados ao longo do projeto bem como alguns resultados já obtidos até o presente momento.

4.1 Resultados Esperados

Os algoritmos propostos na seção 3 serão implementados e seu desempenho testado com 79.466 imagens da base de dados sintética de pares de dígitos conectados. Precisamos então avaliar, para cada uma destas imagens, se os pontos de segmentação gerados por cada algoritmo segmentam corretamente o par de dígitos conectados em questão. Como já citado, nossa base de dados possui a informação de quais são os pontos ótimos de segmentação.

Para avaliar se os pontos de segmentação gerados por cada método segmentam corretamente o par de dígitos conectados, utilizaremos os dois métodos de avaliação propostos na seção 3. O primeiro sendo a comparação direta com os pontos ótimos, sendo esta comparação em relação ao posicionamento dos mesmos, bem como com a quantidade de “cortes” realizados na imagem pelo(s) caminho(s) de segmentação gerado(s). O segundo método à ser utilizado é a rede neural proposta por Oliveira et al [Oli03].

Além da comparação global dos algoritmos, iremos compará-los quanto ao desempenho específico para cada tipo de imagem de entrada, conforme a classificação proposta na seção 3, a qual é exemplificada pela figura 29.

Pretendemos comparar também os algoritmos quanto ao número de pontos de segmentação encontrados. Logicamente o número considerado é o fornecido pelo algoritmo, antes de se descartar os pontos mais distantes dos pontos ótimos como no método descrito acima. Esta análise da quantidade de pontos de segmentação gerados também além de ser feita globalmente, será realizada separadamente para cada um dos tipos (fig. 29) de conexão existentes. Esta comparação da quantidade de pontos de segmentação gerados é um bom parâmetro para comparar o custo computacional dos algoritmos. Logicamente para se ter uma comparação completa e justa deste custo entre todos os algoritmos, teríamos que analisar todo o método e algoritmos intermediários utilizados. Porém como este não é o foco do projeto vamos nos restringir a compará-los de acordo com a quantidade de pontos gerada e alguma análise superficial da complexidade do algoritmo de forma geral.

Pretendemos com isso então, chegar à um modelo de sistema, no qual teríamos um aumento significativo nas taxas de acerto, em relação aos métodos existentes. Este sistema seria composto por um classificador o qual determinaria qual o tipo de conexão presente na imagem de entrada, logo após teríamos um conjunto de métodos. Desta forma dependendo da classificação da imagem de entrada, seria decidido para qual ou quais dos algoritmos a mesma deveria ser segmentada. Com esse modelo conseguiríamos aumentar o desempenho geral do sistema, aproveitando das “especialidades” de cada algoritmo utilizado. Visto que cada algoritmo só seria utilizado para os tipos de imagem no qual possui melhor desempenho.

4.2 Resultados Obtidos

Já realizamos a implementação do algoritmo proposto por Oliveira et al [OLBS00]. Também já foi realizado parte dos testes deste algoritmo seguindo o método proposto na seção 3 e obtidos os resultados esperados segundo o descrito na seção 4.1. O desempenho global deste método de segmentação, de acordo com a avaliação realizada com o uso da rede neural, ficou em 78,95% de acerto. O desempenho para cada tipo de conexão é exibido pelo gráfico presente na figura 34. Em média o algoritmo gerou 4,1 pontos de segmentação para cada imagem processada. A quantidade média de pontos de segmentação gerados por tipo de conexão é exibida pelo gráfico presente na figura 35.

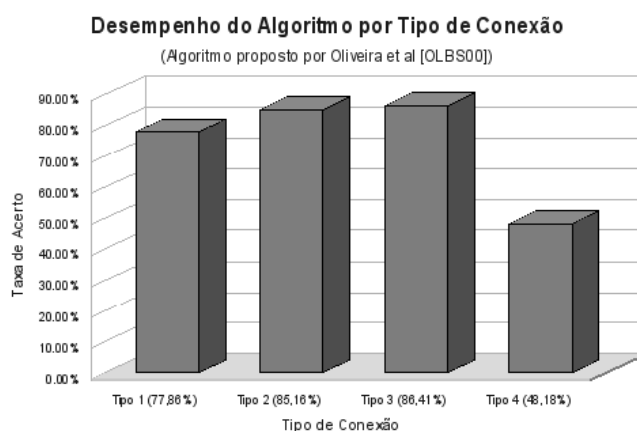


Figura 34: Desempenho obtido pelo método para cada um dos tipos de conexão (fig. 29).

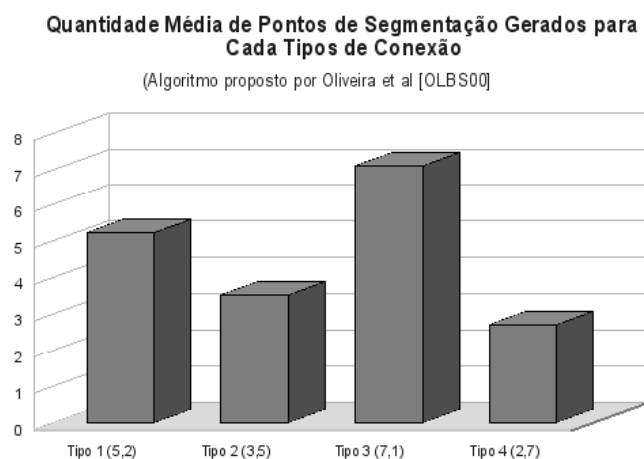


Figura 35: Quantidade média de pontos de segmentação gerados pelo algoritmo, para cada tipo de conexão (fig. 29).

Observando os gráficos acima, contendo os resultados obtidos, podemos concluir que o

desempenho global do algoritmo foi prejudicado pela baixa taxa de acerto em imagens contendo conexão múltipla (Tipo 4). Olhando para o número de pontos de segmentação encontrados pelo algoritmo para cada tipo de conexão (fig. 35) podemos perceber também o porquê deste desempenho mais baixo. Pois o tipo 4 (conexão múltipla) dentre todos os tipos propostos é o que necessita de mais pontos de segmentação, porém o algoritmo fez justamente o contrário. O menor número médio de pontos de segmentação encontrado foi para este tipo de conexão.

Ainda precisamos avaliar o desempenho deste algoritmo com a comparação direta dos pontos encontrados com os pontos ótimos de segmentação, conforme descrito na seção 3.

5 Cronograma

As próximas etapas do desenvolvimento deste projeto irão ocorrer conforme o cronograma apresentado abaixo (tab. 2). Dez algoritmos de segmentação encontrados na literatura serão implementados, testados sobre a base de dados sintética e os resultados obtidos analisados de acordo com o desempenho para cada tipo de conexão. De posse dos resultados pretendemos escrever um artigo e submetê-lo para o ICDAR 2009 (*Tenth International Conference on Document Analysis and Recognition*), conferência essa que tem 12 de janeiro de 2009 como *deadline* para envio de artigos.

Tabela 2: Cronograma descritivo das atividades do projeto.

	Abr 2007	Mai 2007	Jun 2007	Jul 2007	Ago 2007	Set 2007	Out 2007	Nov 2007	Dez 2007	Jan 2008	Fev 2008
Implem. Algor. 1	•	•									
Escrever PDM	•	•									
Defesa do PDM			•								
Implem. Algor. 2			•	•							
Implem. Algor. 3			•	•							
Implem. Algor. 4				•	•						
Implem. Algor. 5				•	•						
Implem. Algor. 6					•	•					
Implem. Algor. 7					•	•					
Implem. Algor. 8						•	•				
Implem. Algor. 9						•	•				
Implem. Algor. 10							•	•			
Anal. Resultados.								•	•		
Escrever Artigo								•	•	•	
Subm. Art. ICDAR										•	
Dissertação									•	•	•
Defesa Dissert.											•

Referências

- [ASBS03] F. Bortolozzi A. S. Britto, R. Sabourin and C. Y. Suen. The recognition of handwritten numeral strings using a two-stage hmm-based method. *IJDAR*, 5(2-3):102–117, 2003.
- [CL96] Richard G. Casey and Eric Lecolinet. A survey of methods and strategies in character segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(7):690–706, 1996.
- [CO99] Soon-Man Choi and Il-Seok Oh. A segmentation-free recognition of two touching numerals using neural network. In *ICDAR '99: Proceedings of the Fifth International Conference on Document Analysis and Recognition*, page 253, Washington, DC, USA, 1999. IEEE Computer Society.
- [CW00] Yi-Kai Chen and Jhing-Fa Wang. Segmentation of single- or multiple-touching handwritten numeral string using background and foreground analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11):1304–1317, 2000.
- [EA03] Ashraf Elnagar and Reda Alhajj. Segmentation of connected handwritten numeral strings. *Pattern Recognition*, 36(3):625–634, March 2003.
- [Fen91] Richard K. Fenrich. Segmentation of automatically located handwritten numeric strings. In *2nd IWFHR*, pages 33–44, 1991.
- [HFK92] Y. Nakano H. Fujisawa and K. Kurino. Segmentation methods for character recognition: from segmentation to document structure analysis. volume 80, pages 1072–1092. IEEE Computer Society, 1992.
- [LLDF04] Yun Lei, C. S. Liu, X. Q. Ding, and Qiang Fu. A recognition based system for segmentation of touching handwritten numeral strings. *International Workshop on Frontiers in Handwriting Recognition*, 00:294–299, 2004.
- [NA98] F T Yarman-Vural N Arica. A new scheme for off-line handwritten connected digit recognition. In *ICPR '98: Proceedings of the 14th International Conference on Pattern Recognition*, volume 2, page 1127, Washington, DC, USA, 1998. IEEE Computer Society.
- [OBS05] Luiz E. Soares Oliveira, Alceu S. Jr. Britto, and Robert Sabourin. A synthetic database to assess segmentation algorithms. In *ICDAR 2005*. IEEE Computer Society, 2005.
- [OLBS00] Luiz E. Soares Oliveira, Edouard Lethelier, Flávio Bortolozzi, and Robert Sabourin. A new segmentation approach for handwritten digits. In *ICPR*, pages 2323–2326, 2000.

- [Oli03] Luiz E. Soares Oliveira. *Automatic Recognition of Handwritten Numerical Strings*. PhD thesis, University of Quebec, ETS, Montreal, QC, Canada, Julho 2003.
- [PB03] U. Pal and A. Bela. Water reservoir based approach for touching numeral segmentation. *Pattern Recognition Letters*, 24(1):261–272, January 2003.
- [SG97] Zhixin Shi and Venu Govindaraju. Segmentation and recognition of connected handwritten numeral strings. *Pattern Recognition*, 30(9):1501–1504, 1997.
- [SN04] Misako Suwa and Satoshi Naoi. Segmentation of handwritten numerals by graph representation. *International Workshop on Frontiers in Handwriting Recognition*, 00:334–339, 2004.
- [SSB07] Javad Sadri, Ching Y. Suen, and Tien D. Bui. A genetic framework using contextual knowledge for segmentation and recognition of handwritten numeral strings. *Pattern Recognition*, 40(3):898–919, 2007.
- [WGS00] Xian Wang, Venu Govindaraju, and Sargur N. Srihari. Holistic recognition of handwritten character pairs. *Pattern Recognition*, 33(12):1967–1973, 2000.
- [YY01] Donggang Yu and Hong Yan. Separation of touching handwritten multi-numeral strings based on morphological structural features. *Pattern Recognition*, 34(3):587–599, 2001.