ISRAEL RIOS

Busca por Palavras em Imagens de Documentos: Uma Abordagem Independente de OCR

Dissertação apresentada ao Programa de Pós-Graduação em Informática da Pontifícia Universidade Católica do Paraná como requisito parcial para obtenção do título de Mestre em Informática.

Curitiba – PR
Julho/2007

ISRAEL RIOS

Busca por Palavras em Imagens de Documentos: Uma Abordagem Independente de OCR

Dissertação apresentada ao Programa de Pós-Graduação em Informática da Pontifícia Universidade Católica do Paraná como requisito parcial para obtenção do título de Mestre em Informática.

Área de Concentração: Ciência da Computação.

Orientador: Alceu de Souza Britto Jr., Dr.
Co-orientador: Alessandro Lameiras Koerich, Dr.

Curitiba – PR
Julho/2007

Rios, Israel
Busca por Palavras em Imagens de Documentos: Uma Abordagem Independente de OCR. Curitiba – PR, Julho/2007.

Dissertação - Pontifícia Universidade Católica do Paraná. Programa de Pós-Graduação em Informática.

1. Recuperação de Texto em Imagens de Documentos 2. Comparação Inexata de Características 3. Segmentação de Imagens de Eocumentos I.Pontifícia Universidade Católica do Paraná. Centro de Ciências Exatas e Tecnologia. Programa de Pós-Graduação em Informática II - t

Ao meu Pai e minha Mãe por todo amor, apoio espiritual e financeiro. A minha esposa, Sílvia, pela inspiração e carinho.

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Símbolos

δ	Parâmetro no cálculo da prioridade do retângulo na busca por espaços em branco
α	Valor que multiplica a mediana das distâncias entre componentes conexos em uma linha para efetuar a segmentação de palavras
σ	Atributo de linha ou transição no conjunto de características LRPS
ω	Atributo de posicionamento com relação às linhas de ascendentes e descendentes no conjunto de características LRPS
λ	Limiar de similaridade utilizado no método de comparação de descritores

Lista de Abreviações

OCR	*Optical Character Recognition*
ASCII	*American Standard Code for Information Interchange*
PDF	*Portable Document Format*
TIFF	*Tagged Image File Format*
CC	*Componente Conexo*
HMM	*Hidden Markov Model*
SOM	*Self Organizing Map*
LRPS	*Left-to-Right Primitive String*
LTA	*Line-or-Traversal Attribute*
ADA	*Ascender-and-Descender Attribute*
DTW	*Dynamic Time Warping*
AYV	*Conjunto de características baseado nas características desenvolvidas por Arica e Yarman-Vural*
ULTC	*Upper and Lower profiles and Transitions Count*

Resumo

Hoje em dia, há um grande volume de informação disponível na forma digital, seja em grandes empresas seja em bibliotecas digitais. Grande parte dessa informação é composta de imagens de documentos digitalizados. Devido ao grande volume, existe a necessidade de prover métodos de acesso rápido a essa informação. Entretanto, as ferramentas atuais de indexação e busca não estão preparadas para lidar com esse tipo de dados e o uso de OCR tem se mostrado uma opção cara do ponto de vista computacional. Neste contexto, surge o grupo dos métodos que visam possibilitar a busca por palavras em documentos de imagens sem utilizar OCR, no qual este trabalho se insere. Estes métodos tem como vantagem o menor tempo de execução, além de geralmente serem mais robustos em documentos ruidosos. Neste trabalho será apresentado um método independente de OCR capaz de buscar uma cadeia de caracteres, no formato ASCII, informada pelo usuário em uma imagem de documento digitalizada. O método ainda tem como característica a possibilidade de encontrar palavras dentro de palavras através da utilização de um algoritmo de comparação de descritores capaz de fornecer uma medida de similaridade entre dois descritores, representantes da palavra procurada e da imagem da palavra analisada, e realizar a correspondência parcial entre eles. Além disso, será apresentado o processo de segmentação da página em palavras, as quais são a menor unidade de segmentação, visto que a segmentação a nível de caractere não será utilizada. Três conjuntos de características distintos foram avaliados com o objetivo de encontrar um boa representação da imagem da palavra apenas com características extraídas de colunas e da análise das linhas dos traços dos caracteres. Os resultados dos testes realizados em mais de 800 imagens de documentos mostraram a viabilidade do método, onde foi possível obter taxas de precisão e revocação em torno de 65% e 70%, respectivamente.

Palavras-chave: Recuperação de Texto em Imagens de Documentos, Comparação Inexata de Características, Segmentação de Imagens de Documentos.

Abstract

Nowadays, there is a large volume of information available in digital format, either in large companies either in digital libraries. Most of of this information is composed of scanned document images. Due to the large volume, there is the urgency to provide fast access methods to this information. However, the current tools of indexing and search are not prepared to deal with this type of data and the OCR use has shown itself an expensive option of the computational point of view. In this context, appears the group of methods that aim to make possible the search for words in documents images without using OCR, in which this work is inserted. These methods have as advantage the smaller execution time, beyond generally being more robust in noisy documents. In this work will be presented an OCR free method capable of search a string, in ASCII format, informed by the user in a scanned document image. The method even has as characteristic the possibility to find words inside of words through the use of a describers matching algorithm capable to supply a measure of similarity between two describers, representatives of the searched word and of the analyzed word image, and to carry through the partial correspondence between them. Moreover, the segmentation process of the page into words will be presented, which are the smaller unit of segmentation, since the segmentation at the character level will not be used. Three distinct sets of features had been evaluated with the objective to find a good representation of the word image only with features extracted from columns and from the analysis of the lines of the characters strokes. The results of tests carried out in more than 800 document images had shown the viability of the method, where it was possible to obtain precision rates around 65% and recall rates around 70%.

Keywords: Text Retrieval in Document Images, Inexact Feature Matching, Document Images Segmentation.

Capítulo 1

Introdução

Com o aumento da capacidade de armazenamento e da redução no custo das memórias de computador tem ocorrido um aumento no número de dados armazenados por empresas e por pessoas comuns. Essas informações muitas vezes se encontram na forma de imagem. Da mesma maneira, algumas empresas têm adotado a prática de manter cópias digitais de alguns documentos na forma de imagem, para facilitar o compartilhamento de informações e diminuir o espaço necessário para armazenamento. Porém, os recursos de software existentes para manipular essas imagens de documentos ainda não estão aptos a indexar, de forma consistente e automática, imagens de documentos. Faz-se necessário que existam ferramentas capazes de indexar imagens a partir do conteúdo de maneira automática, para que o armazenamento de documentos na forma digital seja realmente uma alternativa plausível.

Uma alternativa que algumas empresas têm adotado para indexar imagens é utilizar indexação manual, onde o trabalho de colher palavras-chave ou até mesmo transcrever o documento inteiro é feito por pessoas. Apesar de ser uma alternativa que pode apresentar resultados satisfatórios, esta possui um alto custo, pois demanda muitos recursos pessoais e temporais, podendo torná-la inviável.

No intuito de indexar esses documentos de forma automatizada algumas propostas utilizam-se do Reconhecimento Óptico de Caracteres OCR (*Optical Character Recognition*), como na proposta de Kise, Wuotang e Matsumoto (2003). Ou seja, transformar texto digitalizado para texto ASCII (*American Standard Code for Information Interchange*), que é um formato facilmente manipulado por computador, para depois utilizar técnicas de indexação tradicionais para documentos no formato ASCII. O problema é que o OCR ainda apresenta erros no processamento de documentos com formatos complexos e o problema pode piorar quando a imagem em questão não for de boa qualidade. Além

disso, os erros causados durante a conversão da imagem para texto podem impossibilitar a localização de um determinado documento a partir de uma palavra fornecida, visto que a comparação de duas cadeias de caracteres, quando é realizada pela simples comparação binária dos caracteres, retorna uma resposta positiva ou negativa, limitando a busca às palavras que foram corretamente identificadas. Uma alternativa interessante tem sido o uso de estratégias que executam a comparação de dois descritores e retornam o grau de semelhança entre eles ou a probabilidade das palavras que eles representam serem iguais.

Quando se está trabalhando com imagens de documentos, que não podem ser transformadas de forma segura (sem erros na conversão) para o formato textual, uma opção é manter o documento na forma de imagem e utilizar métodos de indexação que sejam capazes de retornar a probabilidade de certa palavra ser igual a outra presente no documento. Assim, pode-se retornar todos os itens que alcançarem uma probabilidade pré-definida. As chances de encontrar o que se procura aumentam, de maneira que mais informação será retornada. Porém, o método deve ter a precisão suficiente para evitar que muita informação indesejada seja retornada para o usuário.

Nesse contexto, surgem os métodos que fazem uso da técnica de *Word Spotting*, destinados a trabalhar com documentos em um nível intermediário, entre o formato de imagem e o formato textual. Essa técnica pode ser descrita como segue:

> *Word Spotting* é uma técnica que permite encontrar todas as ocorrências de uma palavra em um documento (manuscrito ou impresso). Este método é usado quando os OCRs comerciais não podem processar o documento (documentos antigos, manuscritos ...). O princípio é selecionar um modelo (a palavra que se deseja encontrar) e comparar sua forma com a forma de cada palavra no documento. Isto permite acessar a lista de todas as ocorrências de uma palavra no texto e assim nos permite indexá-lo (LEYDIER, 2004).

A técnica de *Word Spotting* basicamente consiste em localizar imagens de palavras que se assemelhem a uma imagem de palavra selecionado do texto. O fato de se ter que selecionar uma imagem extraída do texto que represente a palavra que se deseja encontrar, limita a utilização dessa técnica no contexto de recuperação automatizada de imagens de documentos. Devido a isso, surgem métodos que permitem a recuperação de imagens de documentos através de uma palavra no formato ASCII (ou outra codificação de caracteres conhecida), realizando a conversão dos caracteres para algo que possa ser comparado com as imagens de palavras coletadas do texto.

Alguns trabalhos foram publicados nesse contexto. Entre os métodos desenvolvidos, diferentes princípios foram utilizados para a comparação das palavras. Já se usou a

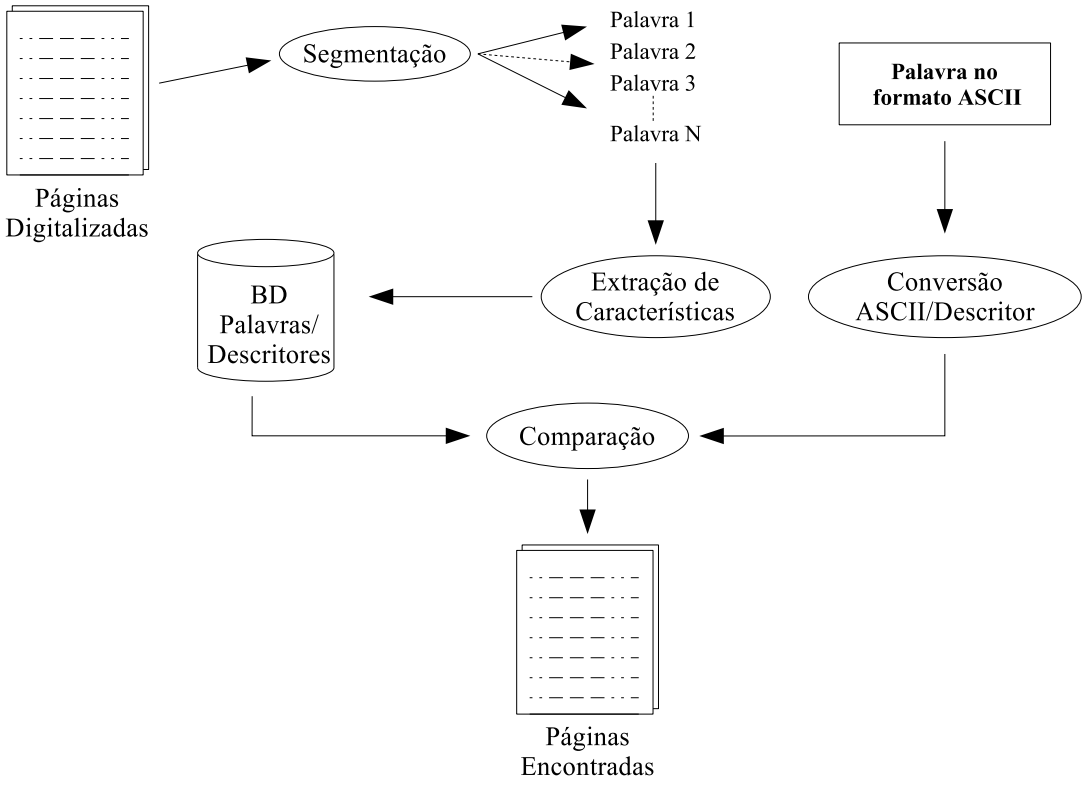

Figura 1.1: Estrutura básica de um sistema para a recuperação de imagens de documentos utilizando uma palavra no formato textual.

relação das linhas e dos traços que formam a palavra (LU; TAN, 2004), as projeções verticais e horizontais da palavra, as linhas superiores e inferiores formadas pela palavra, entre outros. Além disso, Marinai et al. (2004) fizeram uso de informações relativas ao formato do documento para aprimorar a busca do usuário.

Na Figura 1.1, pode-se visualizar a estrutura básica de um sistema que se propõem a realizar a recuperação de imagens de documentos utilizando uma palavra textual. O processo inicia com a segmentação das páginas em palavras. Depois de segmentadas as palavras, as características de cada uma são extraídas gerando descritores que são armazenados em um banco de dados de descritores. Juntamente com a palavra e o descritor são armazenadas informações sobre a posição e o número da página em que ela se encontra. Na etapa de busca o usuário informa a palavra que deseja encontrar no formato ASCII, a qual é transformada em um descritor e comparado com outros descritores existentes no banco de dados. As páginas que contiverem palavras cujos descritores apresentaram similaridade suficiente com o descritor da palavra procurada serão retornadas para o usuário.

A extração de características é um processo fundamental e de grande importância nos sistemas de indexação de documentos digitalizados. Aspectos como caracteres que se tocam, ruídos e invariância a diferentes tipos de fontes, escala, rotação e translação devem

ser levados em conta no momento da especificação do conjunto de características que será utilizado. Neste contexto, o objetivo deste trabalho é desenvolver um método para a localização de palavras em imagens de documentos. Para tanto, a imagem de uma página deve ser segmentada em linhas e a seguir em palavras. A partir de cada palavra no documento será extraído um descritor. A palavra fornecida, no formato ASCII, será convertida para um descritor no mesmo formato do descritor extraído da imagem. Estes descritores serão comparados através da *comparação inexata de características* (LOPRESTI; ZHOU, 1996) (a qual será descrita no Capítulo 2), e uma medida de similaridade entre eles será dada.

Implementada uma versão inicial considerando-se o conjunto de características proposto por Lu e Tan (2004), os esforços serão concentrados no processo de extração de características, visto que este pode ser considerado o mais importante do sistema. Nesse sentido, novos conjuntos de características serão implementados e testados. Finalmente, espera-se, através dessas novas características, melhorar o desempenho do método original.

1.1 Objetivo

O objetivo geral deste projeto é desenvolver um método capaz de encontrar, de forma eficaz, em imagens digitalizadas de documentos impressos palavras que correspondam a palavra que se deseja buscar no formato ASCII. O escopo dos documentos analisados se restringirá a documentos com fontes tradicionais em documentos na língua inglesa tais como Times e Arial.

Alguns objetivos específicos serão importantes para a realização do objetivo principal:

- Criar um banco de dados com imagens de documentos no formato PDF (*Portable Document Format*)[1]. Este banco de dados será utilizado para testes e deverá ser composto por documentos em duas versões: uma no formato textual e uma no formato de imagem. O arquivo no formato textual será utilizado para validar os resultados das buscas efetuadas nos documentos no formato de imagem.

- Desenvolver uma plataforma que possibilite a implementação e avaliação do método proposto. Para a realização de testes e implementação de qualquer método cujo objetivo seja realizar a busca por palavras em imagens de documentos, se faz necessário a implementação de uma plataforma de desenvolvimento. Pretende-se com

[1]Formato desenvolvido para proporcionar o máximo de compatibilidade entre o que se vê na tela e o que se imprime.

essa plataforma realizar tarefas cruciais para qualquer método que se proponha a trabalhar com a análise de uma imagem de documento. Entre essas tarefas pode-se citar:

  - Ler a imagem a partir de diferentes formatos de arquivo, incluindo PDF, TIFF (*Tagged Image File Format*) e outros formatos comumente utilizados para armazenar imagens de documentos. O formato PDF é largamente utilizado na publicação de artigos e outros documentos na Internet. Existem documentos disponibilizados nesse formato que estão no formato texto e podem ser indexados com ferramentas de indexação tradicionais para a indexação de textos. Porém, existem também documentos PDF que consistem em imagens digitalizadas do documento original e que não podem ser indexadas da maneira tradicional. É esse tipo de documento que se deseja tratar;
  - Desenvolver um conjunto de funções para a realização de tarefas atribuídas ao processamento de imagens, tal como binarização, cópia, redimensionamento, entre outros;

- Segmentar o documento ao nível de palavras ou outro nível que seja interessante ao método que se deseja implementar. A segmentação é uma etapa importante do processo de indexação de imagens de documentos. Se esta etapa não obtiver um desempenho satisfatório, comprometerá todo o processo. Assim, figura entre os objetivos deste projeto implementar métodos para realizar essa tarefa de forma eficaz (ver o Capítulo 2);

- Implementar o método inicial, baseado na proposta de Lu e Tan (2004);

- Avaliar diferentes variações do método inicialmente implementado. Novas características devem ser desenvolvidas e testes serão realizados para avaliar o impacto no método.

A realização de cada um destes objetivos é de grande importância para o resultado final. O processo terá inicio como a implementação da plataforma de desenvolvimento e com a criação do banco de dados de testes. A seguir, o método propriamente dito será implementado e o processo de investigação terá inicio, no qual diferentes conjuntos de características serão avaliados. Finalmente, o método será comparado com outros existentes.

1.2 Motivação

Dois pontos principais motivam este projeto:

- O grande volume de documentos disponíveis na Internet e em bibliotecas digitais na forma de imagens. Algumas bibliotecas como a *Making of America* (da Biblioteca da Universidade Cornell), Gallica (da Biblioteca Nacional da França) e a Biblioteca Britânica, possuem métodos precários de indexação de documentos. Algumas fazem uso do OCR e outras indexam manualmente o sumário dos livros. Além disso, artigos científicos também são comumente disponibilizados na forma de imagens incorporadas a documentos PDF;
- A falta de ferramentas capazes de indexar automaticamente e buscar documentos na forma de imagens de maneira eficaz;
- A possibilidade de se avaliar diferentes conjuntos de características na tarefa de buscar palavras em imagens de documentos sem o uso de OCR.

Existe uma crescente demanda em grandes empresas na área de indexação de imagens. Os métodos existentes, baseados em OCR, quando submetidos a documentos de baixa qualidade, têm uma tendência a gerar erros prejudicando a qualidade da busca. Além disso, os métodos baseados em OCR geralmente apresentam um custo computacional maior do que os baseados em outras técnicas que não utilizam OCR, visto que estes adotam um procedimento onde a palavra procurada é aproximada do formato da imagem, evitando assim a necessidade de tentar converter cada imagem de palavra do documento para uma versão textual. Ou seja, ao invés de aproximar a maioria da minoria aproxima-se a minoria da maioria. Porém, ainda há muito para ser feito nessa área, principalmente no desenvolvimento de características e de métodos que não fazem uso do OCR. Os resultados obtidos com técnicas independentes de OCR ainda são inferiores, em termos de precisão e revocação, aos obtidos em métodos que fazem uso do OCR. Por outro lado, a medida que mais trabalhos forem publicados nessa área os números devem ao menos se igualar e os ganhos em tempo de processamento falarão mais auto.

1.3 Contribuições

Este trabalho contribui para a comunidade científica de modo que fornece uma análise de diferentes conjuntos de características no âmbito da localização de palavras em imagens

de documentos. Contribui para a sociedade, a medida que se procura desenvolver um método que venha a acelerar a recuperação de texto em imagens de documentos, o que causa um maior acesso a informação armazenada nesses meios e beneficia um número maior pessoas que necessitam desses dados. Por fim, pode-se relacionar aqui contribuições ao meio ambiente, visto que quando se reduz o número de recursos computacional gastos para se realizar uma determinada tarefa reduz-se o consumo de recursos naturais utilizados pela mesma.

1.4 Organização do Documento

No próximo capítulo estão relacionados alguns dos trabalhos publicados na área de processamento e recuperação de texto em imagens de documentos. No Capítulo 3 está detalhado o método desenvolvido e os conjuntos de características utilizados. No Capítulo 4 têm-se os resultados experimentais, a descrição do procedimento utilizado para calcular tais resultados e a análise do desempenho do método como um todo e dos diferentes conjuntos de características empregados. O fechamento do trabalho e as considerações finais são apresentadas no Capítulo 5.

Capítulo 2

Revisão Bibliográfica

Antes de se chegar a uma proposta para abordar o problema da localização de palavras em imagens de documentos, foi necessário realizar uma pesquisa bibliográfica e levantar algumas das técnicas utilizadas em outros trabalhos, relativos a esta área, publicados ao longo dos últimos anos. Entre os trabalhos avaliados dedicou-se maior atenção aos trabalhos na área de recuperação de texto a partir de imagens de documentos sem o uso de OCR. Nesse contexto, serão relacionados a seguir alguns trabalhos que são objeto de estudo dentro do escopo deste trabalho: segmentação de documentos e recuperação de texto em imagens de documentos.

Durante a avaliação dos trabalhos apresentados neste capítulo, encontrou-se com frequência duas métricas: precisão e revocação. Considerando-se o contexto de uma busca em um conjunto de elementos, o número de acertos (X_c) seria o número de vezes em que a busca encontrou o que se estava procurando, o número de ocorrências (X_t) seria o número de elementos encontrados e o número de elementos existentes (X_e) corresponde ao número de elementos que uma busca ideal retornaria. Assim, pode-se definir a precisão como a relação entre o número de acertos e o número de ocorrências e a revocação como a relação entre o número de acertos e o número de elementos existentes. Assim, podemos calcular a precisão P e a revocação R através das equações 2.1 e 2.2, respectivamente.

$$P = \frac{X_c}{X_t} \tag{2.1}$$

$$R = \frac{X_c}{X_e} \tag{2.2}$$

2.1 Segmentação de Documentos

Uma das etapas presentes em praticamente todos os sistemas de processamento de imagens de documentos é a segmentação. Dependendo do método utilizado o objetivo da segmentação pode ser a delimitação das imagens, tabelas, blocos de texto, parágrafos, linhas de texto, palavras, caracteres e outros itens estruturais presentes em um documento. Uma correta segmentação é essencial para o funcionamento do sistema como um todo. Além disso, seu desempenho deve ser adequado para que não afete o desempenho do sistema.

Neste trabalho o foco da segmentação serão as palavras do texto. Porém, uma das formas de segmentar as palavras é primeiramente segmentar o texto em blocos, em linhas e a seguir em palavras. Assim, a segmentação das colunas e das linhas do texto também são importantes para o foco principal.

Uma das primeiras fases na segmentação de um documentos é a detecção de componentes conexos. Um CC (*Componente Conexo*) nada mais é do que um conjunto de pixels pretos (representando parte de um objeto na imagem) dispostos de tal maneira que pode-se chegar a qualquer pixels presente no componente a partir de outro pixel qualquer presente no mesmo componente. Ou seja, todos os pixel presentes em um componente conexo estão conectados de alguma maneira.

Breuel (2002) propôs dois métodos para solução de problemas relativos a segmentação de documentos: identificar espaços retangulares em branco e identificar linhas levando-se em consideração um conjunto de espaços em branco em uma página. O primeiro método pode ser utilizado para detectar colunas em um documento, desde que encontre um retângulo vazio entre duas colunas de texto. Esse método identifica espaços em branco na ordem crescente de área, utilizando uma fila de prioridades onde a prioridade de cada retângulo é dada pela sua área. A cada iteração do método um retângulo é retirado da fila e dividido em quatro partes que são re-inseridas na fila. Quando um retângulo vazio é encontrado ele é classificado como espaço em branco e inserido na lista de CCs para evitar que outros retângulo contendo partes em comum com este sejam detectados. A iteração continua até que os N maiores retângulos vazios sejam encontrados. O segundo método é utilizado para encontrar linhas de texto. Este método busca encontrar linhas que minimizem a distância para a parte inferior central dos componentes conexos da página. As linhas encontradas não devem exceder os limites gerados pela detecção de espaços vazios. Esses dois métodos juntos formam uma poderosa ferramenta de análise da estrutura de documentos.

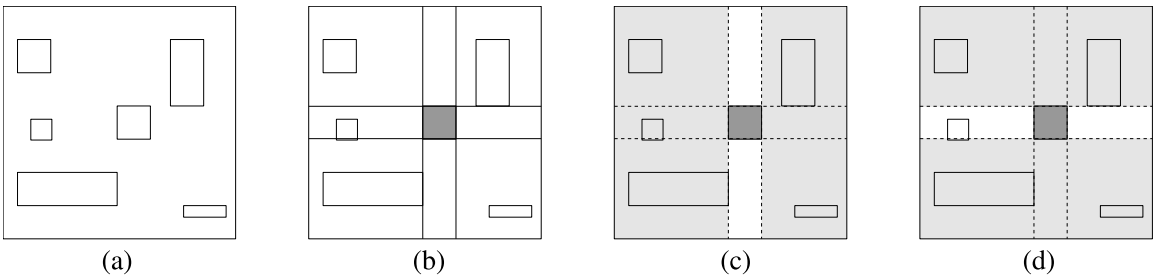

Figura 2.1: Método proposto por Breuel (2002) para detecção de espaços em branco.

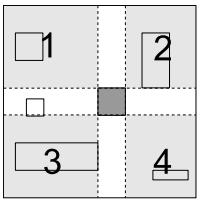

Figura 2.2: Regiões em comum (1, 2, 3 e 4) geradas pela estratégia de divisão utilizada por Breuel.

O algoritmo apresentado por Breuel (2002) para a detecção de espaços é ótimo. Ou seja, apresenta como solução o maior retângulo, que não contém CCs no seu interior, existente no documento. Ao se inserir o retângulo encontrado na lista de CCs do documento, tem-se a capacidade de re-executar o método para obter o segundo maior retângulo e assim por diante. A Figura 2.1 ilustra o processo que está descrito abaixo:

1. Cria-se uma fila de prioridades onde o retângulo que tem maior área fica mais a frente na fila, e insere-se o retângulo inicial (que pode ser correspondente a página a ser analisada). A Figura 2.1(a) representa uma região da imagem onde os retângulos mostrados correspondem aos CCs pertencentes a esta região;

2. Retira-se um retângulo $\Re$ da fila;

3. Se $\Re$ não contiver CCs ele é retornado como resposta. Senão, um componente pertencente a $\Re$ é escolhido para ser o pivô da sua divisão. O melhor componente é aquele que está mais perto do centro do retângulo, conforme mostrado na Figura 2.1(b);

4. Divide-se $\Re$ em quatro, baseando-se nas coordenadas do pivô: a esquerda, direita, acima e abaixo, conforme Figuras 2.1(c) e 2.1(d). Cada sub-retângulo formado é inserido na fila de prioridades;

5. Retorna para o passo 2.

A estratégia de divisão e conquista proposta por Breuel (2002) para identificar espaços vazios é interessante porque garante que o n-ésimo retângulo encontrado será o n-ésimo maior retângulo vazio da imagem, simplificando a tarefa de encontrar os N maiores retângulos vazios. Porém, ela tem a desvantagem de ocasionar o re-processamento de várias áreas da imagem devido a maneira como as divisões são feitas. A divisão de um retângulo como está ilustrado na Figura 2.1 gera regiões em comum, conforme pode-se ver na Figura 2.2, estas regiões serão analisadas mais de uma vez o que torna o algoritmo ineficiente e limita a sua utilização. Tal problema pode ser solucionado dividindo os retângulos sem deixar regiões em comum. Além disso, pode-se dividir os retângulos de maneira a privilegiar aqueles que possuem as maiores relações de $altura/largura$ para favorecer a detecção de colunas. Entretanto, perde-se a garantia de que os N primeiros retângulos encontrados serão os N maiores retângulos da imagem. A modificação necessária é uma das contribuições deste trabalho e será apresentada em detalhes no Capítulo 3.

Wang, Lu e Tan (2003) mostraram que é possível segmentar palavras em uma imagem utilizando diagramas de Voronoi. Primeiro, os componentes conexos são encontrados e a seguir os componentes conexos detectados como ruído e os que provavelmente representam caracteres especiais (“{”,“}”,“(”,“)”,...) são eliminados. O diagrama de Voronoi é calculado para a imagem, formando bordas que separam dois componentes conexos. Cada borda é avaliada segundo um conjunto de restrições que levam em conta a distância da borda aos componentes conexos que ela separa e a relação dessa distância com as distâncias das outras bordas que cercam o componente. Se a borda se enquadrar nas restrições então ela é removida, unindo dois componentes conexos. Este processo continua de modo a formar palavras. Este método tem a vantagem de dispensar a detecção de linhas e segmentar as palavras diretamente. Por outro lado, o cálculo do diagrama de Voronoi é complexo e o processamento envolvido é relativamente custoso.

A segmentação de documentos é uma área bastante desenvolvida e vem sendo objeto de várias pesquisas ao longo dos anos. Sua utilização neste trabalho se da como um pré-requisito para a execução do método de localização de palavras. Desse modo, esta seção não revisa de maneira exaustiva esta área, limitando-se aos trabalhos mais relevantes a nosso objeto de estudo.

2.2 Recuperação de Texto em Imagens de Documentos

Os métodos de recuperação de texto em imagens de documento podem ser divididos em dois grupos: os que fazem uso de OCR e os livres de OCR. No primeiro a imagem do documento é convertida para texto e depois analisada por ferramentas tradicionais de busca em texto. Nos métodos do segundo grupo não se realiza a conversão para texto e a busca é efetuada através da similaridade entre descritores extraídos da imagem e da palavra que se deseja encontrar.

Os métodos livres de OCR tem como vantagem sobre as baseadas em OCR o tempo de execução. Normalmente, os métodos livres de OCR são mais rápidos que os baseados em OCR. Este comportamento é devido ao fato de que o processamento necessário para converter toda a imagem do documento para texto é muito grande. Além disso, métodos livres de OCR são mais indicados para documentos ruidosos, já que a conversão da imagem para texto costuma ser imprecisa nestes casos.

Alguns métodos baseados em OCR utilizam classificadores HMM para a converter imagens de caracteres em caracteres interpretados por computador. Porém, tais classificadores podem ser utilizados mesmo em técnicas que não realizam a conversão da imagem via OCR. No trabalho de Chen, Bloomberg e Wilcox (1996), foi desenvolvido um método para efetuar a busca por palavras e frases em imagens de documentos através do uso de classificadores HMM (*Hidden Markov Model*). Após segmentar o texto em linhas, palavras e caracteres os pixels das colunas existentes entre as linhas de ascendentes e descendentes são utilizados como vetores de características que servem como entrada para a rede HMM. Assim, a única funcionalidade do processo de extração de características é enquadrar uma coluna de pixels em um vetor de tamanho pré-definido. Através da utilização desse método em um subconjunto do banco de imagens de documentos da Universidade de Washington(UW), foram relatadas revocação igual a 94% e taxa de falsos alarmes igual a 0,2% na localização de frases com três palavras e taxas de falsos alarmes igual a 0,5% na localização de palavras individuais. Portanto, pode-se concluir que quanto mais informação fornecida melhores são os resultados obtidos. Um dos pontos fracos do método reside na necessidade de uma etapa de treinamento.

Ainda no contexto de busca por palavras em documentos pode-se relacionar o trabalho de Marinai et al. (2004), onde o formato da página foi utilizado para auxiliar na recuperação de documentos. Nesse trabalho os autores desenvolveram um método para fazer a recuperação de imagens de documentos através da análise do formato da página e das palavras-chave que se deseja encontrar. Na fase de indexação o método gera três

repositórios de informações sobre o documento que se está indexando: árvores MXY[1], formatos de página e palavras. No primeiro repositório são armazenadas as árvores MXY para cada página do documento. Nesse repositório a árvore é armazena de forma completa, representando todas as divisões encontradas na página. No segundo repositório, a árvore MXY é codificada em uma forma mais compacta, rotulando padrões de árvore pré-definidos. Além disso, nesse repositório são armazenadas informações relativas à página como um todo, como a percentagem da página ocupada pelo texto. No terceiro repositório são armazenadas palavras na forma vetorial. Os vetores de palavras são formados pelas posições de cada caractere relativas a um SOM (*Self Organizing Map*), o qual é um tipo especial de rede neural, correspondendo à aglomeração que o caractere pertence. Na etapa de recuperação a palavra procurada é transformada em uma imagem e seus caracteres submetidos ao SOM para formar o vetor de características que será buscado no repositório de palavras. Uma página com o formato desejado pode ser fornecida para aprimorar a busca. Métodos de busca foram desenvolvidos pelos autores, os quais combinaram o formato e as palavras do documento. Através da utilização do repositório de formatos de página, com o método proposto é possível buscar palavras em lugares específicos do documento, como em legendas de imagens ou em início de capítulos. Porém, para documentos ruidosos esse método pode não apresentar bons resultados devido ao fato de segmentar palavras em caracteres, estratégia que tradicionalmente não traz bons resultados para documentos com caracteres que se tocam.

Para cada idioma de escrita dos documentos a localização de palavras apresenta particularidades. Nesse sentido, pode-se relacionar o trabalho de Lu e Tan (2002). Os autores desenvolveram um método para efetuar a busca por palavras ou frases em imagens de documentos chineses, sem a necessidade de analisar o formato do documento. O método se propõe a encontrar palavras na vertical e na horizontal. Numa primeira etapa os componentes conexos são encontrados e alguns componentes conexos próximo são unidos, de acordo com a relação proximidade e distância, para formar caracteres chineses. A seguir um caractere da palavra de pesquisa é utilizado para iniciar a busca. Quando um caractere correspondente for encontrado os caracteres da direita e de baixo da imagem são comparados com os próximos caracteres da palavra de pesquisa. Quando o primeiro caractere pesquisado não for o primeiro caractere da palavra de pesquisa então os caracteres da direita e de cima são comparados com o caractere anterior da palavra de pesquisa e assim sucessivamente. A imagem do caractere é dividida em um grade e a densidade de

[1] Árvores MXY são utilizadas na análise do formato do documento. A raiz da árvore é a própria página do documento, os nós internos são as divisões da página e os nós folhas são as regiões de texto e imagens existentes na página.

cada região é utilizada para comparar os dois caracteres, se houver correspondência então a distancia ponderada de Hausdorff é calculada para finalmente dizer se os caracteres correspondem. O método proposto é relativamente simples, porém, segundo os próprios autores, não apresenta um bom desempenho com imagens ruidosas.

Um dos métodos fortemente relacionados com a pesquisa apresentada neste documento é o proposto por Lu e Tan (2004). Os autores propuseram um método para efetuar a busca por palavras em imagens de documentos baseado em informações estruturais da imagem da palavra. Após a segmentação do documento em palavras, a extração de características de linhas e de passagem das mesmas é realizada. As características extraídas das imagens das palavras são chamadas de LRPS (*Left-to-Right Primitive String*), as quais são formadas através da análise da imagem da palavra da esquerda para a direita.

Cada característica p do conjunto de características é formada por um par de atributos (σ, ω), onde σ é o LTA (*Line-or-Traversal Attribute*) e ω é o ADA (*Ascender-and-Descender Attribute*). Como resultado a imagem de uma palavra é expressa por uma seqüência P de p_i's.

$$P =< p_1 p_2 ... p_n >=< (\sigma_1, \omega_1)(\sigma_2, \omega_2)...(\sigma_n, \omega_n) > \tag{2.3}$$

O atributo ω da característica p pode receber um dos seguintes valores:

'x' se a primitiva[2] está entre linha de ascendentes[3] e a linha de descendentes;

'a' se a primitiva está entre a margem superior e a linha de ascendentes;

'A' se a primitiva está entre a margem superior e a linha de descendentes;

'D' se a primitiva está entre a linha de ascendentes e a margem inferior;

'Q' se a primitiva está entre a margem superior e a margem inferior.

O atributo σ da característica p é formado por características de linhas retas (se houver) ou características de transição. Para gerar este atributo o algoritmo percorre a imagem horizontalmente até encontrar um ponto preto. Ao encontrar este ponto preto ele tenta encontrar a maior seqüência de pontos consecutivos dentro de determinados ângulos. Se a maior seqüência encontrada for maior ou igual a distância entre a linha de

[2]Primitiva é o traço ou o conjunto de pixels que esta sendo analisado dentro da imagem da palavra.

[3]linha imaginária paralela a linha de descendentes que se encontra no topo da maioria das letras minúsculas

ascendentes e a linha de descendentes então a linha que obteve o maior comprimento é retirada da imagem, e o atributo σ recebe um dos seguintes valores:

'l' : linha reta vertical. Se o atributo ω da primitiva for 'x' ou 'D' e se existir um ponto acima da primitiva este valor será mudado para 'i';

'v' : linha reta diagonal da direita para baixo;

'w' : linha reta diagonal da esquerda para baixo. Se o atributo ω da primitiva for 'x' ou 'A' e se existirem duas retas horizontais conectadas as extremidades da linha este valor será mudado para 'z';

'x' : uma linha reta diagonal da esquerda para baixo cruza com uma linha reta diagonal da direita para baixo;

'y' : uma linha reta diagonal da esquerda para baixo encontra o meio de uma linha reta diagonal da direita para baixo;

'Y' : uma linha reta diagonal da esquerda para baixo, uma linha reta diagonal da direita para baixo e uma linha reta vertical se encontram em um ponto;

'k' : uma linha reta diagonal da esquerda para baixo, uma linha reta diagonal da direita para baixo e uma linha reta vertical se tocam no mesmo ponto.

Depois das primitivas de linhas retas serem retiradas da imagem as primitivas restantes são analisadas, coluna a coluna, para contar o número de transições de pixels pretos para pixels brancos (NT); verificar a distância do pixel mais abaixo, com relação a linha de descendentes (D_b); a distância do pixel mais acima, com relação a linha de ascendentes (D_m) e calcular a razão do número de pixels pretos pela distância da linha de descendentes até a linha de ascendentes (k). Se NT = 0 então σ e ω recebem **'&'** que corresponde a um espaço em branco. Se NT = 2 então tem-se $\xi = D_m/D_b$ e o atributo σ pode receber um dos seguintes valores:

$$\sigma = \begin{cases} n & \text{se } k < 0,2 \text{ e } \xi < 0,3 \\ u & \text{se } k < 0,2 \text{ e } \xi < 3 \\ c & \text{se } k < 0,5 \text{ e } \xi < 1,5 \end{cases}$$

Se NT >= 4, σ recebe:

$$\sigma = \begin{cases} o & \text{se NT} = 4 \\ e & \text{se NT} = 6 \\ g & \text{se NT} = 8 \end{cases}$$

Ca	LRPS	Ca	LRPS
a	(o,x)(e,x)(l,x)	A	(w,A)(v,A)
b	(l,A)(o,x)(c,x)	B	(l,A)(e,A)(o,A)
c	(c,x)(o,x)	C	(c,A)(o,A)
d	(c,x)(o,x)(l,A)	D	(l,A)(o,A)(c,A)
e	(c,x)(e,x)(o,x)	E	(l,A)(e,A)
f	(n,x)(l,A)(u,a)	F	(l,A)(o,A)(u,a)
g	(g,D)(e,D)	G	(c,A)(o,A)(e,A)(o,A)
h	(l,A)(n,x)(l,x)	H	(l,A)(n,x)(l,A)
i	(i,A)	I	(l,A)
j	(i,Q)	J	(u,x)(l,A)
k	(k,x)	K	(k,A)
l	(l,A)	L	(l,A)(u,x)
m	(l,x)(n,x)(l,x)(n,x)(l,x)	M	(l,A)(v,A)(w,A)(l,A)
n	(l,x)(n,x)(l,x)	N	(l,A)(v,A)(l,A)
o	(c,x)(o,x)(c,x)	O	(c,A)(o,A)(c,A)
p	(l,D)(o,x)(c,x)	P	(l,A)(o,A)(c,A)
q	(c,x)(o,x)(l,D)	Q	(c,A)(o,A)(e,Q)(o,D)
r	(l,x)(n,x)	R	(l,A)(o,A)(e,A)(o,A)
s	(o,x)(e,x)(o,x)	S	(o,A)(e,A)(o,A)
t	(n,x)(l,A)(o,x)	T	(u,a)(l,A)(u,a)
u	(l,x)(u,x)(l,x)	U	(l,A)(u,x)(l,A)
v	(v,x)(w,x)	V	(v,A)(w,A)
w	(v,x)(w,x)(v,x)(w,x)	W	(v,A)(w,A)(v,A)(w,A)
x	(x,x)	X	(x,A)
y	(y,D)	Y	(Y,A)
z	(z,x)	Z	(z,A)

Tabela 2.1: Seqüência de características LRPS dos caracteres (LU; TAN, 2004).

Quando duas características vizinhas do vetor de características são iguais, apenas uma é deixada no vetor. Além disso, um pré-processamento é feito para remover características decorrentes de fontes com serifa, ou seja, fontes que apresentam pontos adjacentes nas extremidades de algumas letras.

Para efetuar uma pesquisa em um documento de imagem, através do método proposto, a palavra procurada deve ser representada em um vetor de características LRPS. Assim sendo, cada caractere foi mapeado para uma seqüência de características LRPS, possibilitando a conversão de uma cadeia de caracteres apenas pela substituição de cada caractere pela sua seqüência equivalente em LRPS e a adição da característica (&,&) entre cada caractere. Por exemplo, a palavra "health" corresponde a seguinte seqüência de características LRPS: "(l,A)(n,x)(l,x)(&,&)(c,x)(e,x)(o,x)(&,&)(o,x)(e,x)(l,x)(&,&)(l,A)(&,&)(n,x)(l,A)(o,x)(&,&)(l,A)(n,x)(l,x)". A Tabela 2.1 traz a relação completa dos caracteres e suas respectivas seqüências de características.

Para medir a similaridade entre dois vetores de características a técnica de *comparação inexata de características*(LOPRESTI; ZHOU, 1996) foi utilizada. O algoritmo faz uso da técnica de programação dinâmica, a qual utiliza uma tabela para armazenar os dados já calculados pelo algoritmo para evitar o recálculo. O algoritmo compara os dois vetores de características e através de pesos, atribuídos para cada par de características diferentes, chega a uma pontuação, onde pode ser medida a similaridade entre as duas palavras.

O cálculo da pontuação (similaridade) é feito de forma tabular, onde o tamanho dos dois vetores a serem comparados corresponde às dimensões da tabela. Ao final do cálculo a tabela é percorrida em busca do maior valor, que será então normalizado entre 0 e 1 e comparado a um limiar para dizer se os vetores são semelhantes o suficiente.

O pseudocódigo do algoritmo utilizado para comparar dois conjuntos a e b de características LRPS pode ser visualizado no Algoritmo 1. Um traço "$-$" representa a primitiva de espaço inserida na correspondente posição da cadeia de caracteres e V é a tabela (matriz) utilizada para armazenar os resultados. As funções $\epsilon(p_1, p_2)$, $\mu(p_1, p_2)$ e $\nu(p_1, p_2)$ retornam o peso atribuído ao par de características p_1 e p_2, o qual é positivo quando $p_1 = p_2$ e negativo quando $p_1 \neq p_2$. Um mínimo de 0 (zero) é atribuído a $V(i, j)$ para garantir ao algoritmo a capacidade de reinicio.

Algoritmo 1 Cálculo da similaridade utilizando programação dinâmica

{inicialização}
para $i \leftarrow 0$ **até** n **faça**
  $V(i, 0) \leftarrow 0$
fim para
para $j \leftarrow 0$ **até** m **faça**
  $V(0, j) \leftarrow 0$
fim para
{calcula cada elemento levando em conta os resultados anteriores}
para $i \leftarrow 1$ **até** n **faça**
  para $j \leftarrow 1$ **até** m **faça**

$$V(i,j) \leftarrow max \begin{cases} 0 \\ V(i-1, j-1) + \epsilon(a_i, b_j) \\ V(i-1, j) + \mu(a_i, -) \\ V(i, j-1) + \nu(-, b_i) \end{cases}$$

  fim para
fim para

Após a comparação de "unhealthy" e "health" tem-se o resultado ilustrado na Tabela 2.2, na qual se pode ver que a medida de similaridade (nesse exemplo) é igual a 34, ou seja, o maior valor da tabela. Os números em negrito mostram qual dos três vizinhos de $V(i, j)$ obteve maior valor e foi escolhido pelo algoritmo e qual região das palavras

			un										h				e			al					th				y
			(l,x)	(u,x)	(l,x)	(u,x)	(o,x)	(l,x)	(u,x)	(l,x)	(&)	(l,A)	(o,x)	(l,x)	(&)	(c,x)	(e,x)	(o,x)	(&)	(o,x)	(e,x)	(l,x)	(l,A)	(&)	(n,x)	(l,A)	(o,x)	(l,A)	(n,x)	(l,x)	(&)	(y,D)
		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
h	(l,A)	0	0	0	0	0	0	0	0	0	0	**2**	1	0	0	0	0	0	0	0	0	0	2	1	0	2	1	2	1	0	0	0
	(n,x)	0	0	0	0	0	0	0	2	1	0	1	**2**	1	0	0	0	0	0	0	0	0	1	1	3	2	2	1	4	3	2	1
	(l,x)	0	2	1	2	1	0	2	1	4	3	2	1	**4**	3	2	1	0	0	0	0	2	1	0	2	3	2	2	3	6	5	4
	(&)	0	1	1	1	1	0	1	1	3	6	5	4	3	**6**	5	4	3	2	1	0	1	1	3	1	2	2	1	2	5	8	7
	(c,x)	0	0	1	1	1	1	0	1	1	5	4	5	4	5	**8**	7	6	5	4	3	2	1	2	3	2	2	1	1	4	7	6
e	(e,x)	0	0	0	1	1	1	1	0	1	4	3	4	5	4	7	**10**	9	8	7	6	5	4	3	2	1	2	1	1	3	6	5
	(o,x)	0	0	0	0	1	3	2	1	0	3	2	5	4	4	6	9	**12**	11	10	9	8	7	6	5	4	3	2	1	2	5	4
	(&)	0	0	0	0	0	2	2	1	0	2	2	4	4	6	5	8	11	**14**	13	12	11	10	9	8	7	6	5	4	3	4	4
	(o,x)	0	0	0	0	0	2	2	2	1	1	1	4	4	5	6	7	10	13	**16**	15	14	13	12	11	10	9	8	7	6	5	4
a	(e,x)	0	0	0	0	0	1	2	2	2	1	0	3	4	4	5	8	9	12	15	**18**	17	16	15	14	13	12	11	10	9	8	7
	(l,x)	0	2	1	2	1	0	3	2	4	3	2	2	5	4	4	7	8	11	14	17	**20**	19	18	17	16	15	14	13	12	11	10
	(&)	0	1	1	1	1	0	2	2	3	6	5	4	4	7	6	6	7	10	15	16	**19**	19	21	20	19	18	17	16	15	14	13
l	(l,A)	0	0	0	1	0	0	1	1	2	5	8	7	6	6	5	5	6	9	14	15	18	**21**	20	19	22	21	20	19	18	17	16
	(&)	0	0	0	0	0	0	0	0	1	4	7	7	6	8	7	6	5	8	13	14	17	20	**23**	22	21	20	20	19	18	20	19
	(n,x)	0	0	0	0	0	0	0	2	1	3	6	7	7	7	8	7	6	7	12	13	16	19	22	**25**	24	23	22	22	21	20	19
	(l,A)	0	0	0	0	0	0	0	1	2	2	5	6	7	6	7	6	5	6	11	12	15	18	21	24	**27**	26	25	24	23	22	21
t	(o,x)	0	0	0	0	0	2	1	0	1	1	4	7	6	6	6	7	8	7	10	11	14	17	20	23	26	**29**	28	27	28	25	24
	(&)	0	0	0	0	0	1	1	0	0	3	3	6	6	8	7	6	7	7	9	10	13	16	19	22	25	**28**	28	24	26	28	27
	(l,A)	0	0	0	0	0	0	1	0	0	2	5	5	6	7	6	5	6	6	8	9	12	15	18	21	24	27	**30**	29	28	27	26
h	(n,x)	0	0	0	0	0	0	0	3	2	1	4	4	5	6	7	6	5	5	7	8	11	14	17	20	23	26	29	**32**	31	30	29
	(l,x)	0	2	1	2	1	0	2	1	2	1	3	3	6	5	6	7	6	5	6	7	10	13	16	19	22	25	28	31	**34**	33	32

Tabela 2.2: Resultado final da comparação de “unhealthy” e “health” (LU; TAN, 2004).

apresenta maior similaridade.

Ainda no contexto de indexação de imagens de documentos pode-se citar o trabalho de Balasubramanian, Meshesha e Jawahar (2006), no qual um sistema completo foi desenvolvido para realizar a indexação de imagens de documentos impressos em diferentes linguagens indianas. O método desenvolvido mesclou a rotulação de palavras chave com a indexação automática de imagens de palavras. Os autores mesclaram o uso de características locais e globais. Entre as características locais utilizadas pode-se citar o contorno superior e inferior, projeções verticais e horizontais e número de transições. Entre as globais foram utilizados momentos estatísticos (como média e desvio padrão) e momentos baseados na região (como momentos de primeira e segunda magnitude). Esta combinação de características locais e globais é reconhecidamente robusta. Porém, características globais dificultam a implementação de métodos que busquem sub-palavras.

O método de comparação utilizado por Balasubramanian, Meshesha e Jawahar (2006) fez uso do algoritmo DTW (*Dynamic Time Warping*) responsável por alinhar as características extraídas de duas palavras distintas e retornar um valor de similaridade entre elas. Este método é semelhante a *comparação inexata de características* (Algoritmo 1) utilizada por Lu e Tan (2004). Porém, neste método não é possível detectar sub-palavras e a distância entre dois vetores de características a e b, componentes da seqüência de vetores que compõe as palavras comparadas, é dada por $\sum_{k=1}^{N}(a[k]-b[k])^2$.

Após armazenar os descritores das várias palavras existente no banco de imagens, o

método agrupa as palavras semelhantes para que seja feita uma rotulação manual dos grupos. Esta rotulação visa possibilitar a detecção de palavras parecidas como “directed” e “redirected”. Nem todas as grupos são rotulados, somente os maiores. Através dessa técnica em união com a comparação de características utilizando a DTW os autores chegaram a resultados com precisão e revocação próximos de 95%, realizando os testes sobre um banco de imagens desenvolvido pelos próprios autores. Entretanto, o uso da rotulação manual e a impossibilidade de encontrar sub-palavras que não estejam em grupos previamente rotulados são pontos fracos do método.

Algumas vezes não se deseja procurar apenas uma palavra em um documento, mas sim procurar documentos semelhantes a um outro qualquer. Neste contexto, pode-se relacionar o trabalho de Tan et al. (2002), onde os autores propuseram um método para efetuar a recuperação de imagens de documentos utilizando n-gramas[4]. Um dos pontos mais interessantes do método proposto é a independência da linguagem de escrita dos documentos. O processo se inicia com a segmentação do documento em caracteres, mas sem se preocupar com a segmentação de palavras, visto que se deseja independência de linguagem. Após a segmentação, características relativas ao número de transições verticais e horizontais dos caracteres são extraídas e normalizadas em um vetor de tamanho pré-definido. A seguir, os vetores são agrupados em classes de acordo com a semelhança. Definidas as classes os documentos são analisados através do algoritmo n-grama, onde cada caractere é representado pelo centróide da classe a que pertence, com o objetivo de formar um vetor que descreva o documento. A comparação entre duas imagens de documentos se da pelo produto vetorial entre os dois vetores representantes de cada documento. Utilizando essa técnica, os autores relataram resultados interessantes para documentos com o mesmo tipo de fonte, mas o método foi incapaz de tratar pequenas variações na fonte. Entretanto, os autores apresentaram bons resultados com documentos na língua inglesa e chinesa, com taxas de precisão e revocação iguais a 73,9% e 85,7% respectivamente, mostrando a viabilidade do método no tratamento de documentos escritos em diferentes línguas.

Durante anos de pesquisa muitas características foram utilizadas na difícil tarefa de detectar a semelhança entre palavras situadas em uma imagem gerada a partir da digitalização de um documento. Esta tarefa passa a ser ainda mais complexa quando se trata de um documento manuscrito. Características utilizadas na indexação de documentos manuscritos podem ser utilizadas na indexação de documento impressos, apresentando bons resultados. Em contrapartida, documentos impressos geralmente são mais padronizados

[4]N-grama é uma seqüência de n caracteres consecutivos. Uma seqüência de n-gramas é obtida através do deslocamento de uma janela com n-itens de largura sobre do texto, avançando um caractere por vez.

que documentos manuscritos, fato que muitas vezes não permite que técnicas geralmente empregadas no primeiro sejam utilizadas no último.

Usando *DTW* em seu método de comparação de características, Rath e Manmatha (2003) experimentaram várias características e algumas combinações delas para encontrar palavras semelhantes em um subconjunto de manuscritos da coleção de George Washington. Entre as características avaliadas pode-se citar:

- Projeção Horizontal: soma do valor dos pixels de uma coluna;
- Projeção Vertical: soma do valor dos pixels de uma linha[5];
- Contorno Superior/Inferior: distância entre o pixel mais acima ou abaixo da coluna e a margem superior ou inferior da palavra, respectivamente;
- Número de Transições: número de transições preto/branco em uma coluna;
- Variação em Níveis de Cinza: variação das intensidades dos níveis de cinza dos pixels de uma coluna.

Para reduzir o número de palavras que tiveram suas características comparadas, alguns fatores como a razão altura/largura das palavras em questão foi levada em conta para determinar se a etapa de comparação seria necessária. Ao final, foi relatado um melhor desempenho individual da característica de contorno superior e um melhor desempenho em conjunto das características de contorno superior, contorno inferior e número de transições. No primeiro obteve-se 64,29% de precisão e 58,07% de revocação contra 72,56% de precisão e 65,17% de revocação no último. Os autores também realizaram testes com características obtidas através da suavização Gaussiana e da derivação Gaussiana, porém os resultados foram inferiores a característica de contorno superior.

O trabalho de Rath e Manmatha (2003) mostrou que a combinação de características simples pode apresentar bons resultados. Porém, características mais simples tendem a apresentar suas fraquezas quando submetidas a imagens ruidosas e textos impressos com diferentes fontes ou textos manuscritos com diferentes autores, por isso devem ser utilizadas com cuidado utilizando-se técnicas de pré-processamento sempre que possível.

[5]Esta característica somente pode ser utilizada quando não se deseja encontrar palavras que estejam contidas dentro de outras palavras

2.3 Considerações Finais

Boa parte dos trabalhos analisados não permite encontrar sub-palavras, esse ponto será abordado nesse trabalho através da utilização da técnicas proposta por Lu e Tan (2004), visto que é uma funcionalidade que permite aumentar em muito a abrangência das buscas. O ponto comum entre os métodos é a comparação a nível de descritores, a qual é comumente implementada com os algoritmos DTW ou com a comparação inexata de características. Este último, é muito semelhante ao DTW porém permite detectar sub-descritores dentro de um outro descritor, possibilitando a localização de palavras dentro de palavras. Alguns métodos relacionados aqui utilizam classificadores HMM. Estes classificadores são comumente utilizados no reconhecimento de caracteres via OCR e normalmente exigem grande quantidade de recursos computacionais. Entretanto, um dos objetivos deste trabalho é reduzir o tempo de execução gasto para a realização de uma busca. Assim, deve-se evitar o uso de estratégias que reconhecidamente apresentam alto custo computacional.

Concluída a análise de alguns dos métodos pesquisados, será iniciada a descrição da metodologia a ser utilizada. O método a ser implementado e os conjuntos de características que serão avaliados para fazer a busca de palavras em imagens de documentos sem a utilização de OCR estão descritos no próximo capítulo.

Capítulo 3

Metodologia

Neste capítulo é apresentado, de forma detalhada, o método desenvolvido e as técnicas empregadas durante a pesquisa. Onde é possível visualizar a estrutura da solução desenvolvida, a qual encontra-se dividida em pré-processamento, segmentação, extração e comparação de descritores.

O método proposto é formado por quatro estágios distintos. O primeiro estágio consiste em deixar a imagem da página a ser processada em condições mais favoráveis ao seu processamento. Nesta etapa são realizados procedimentos a fim de extrair ruídos e converter a imagem para o formato esperado pelo sistema. No segundo estágio é realizada a segmentação da imagem da página em blocos de texto, em linhas e em palavras. No terceiro estágio é realizada a extração de características, onde são empregados diferentes conjuntos de características para transformar a imagem de uma palavra em um descritor simbólico[1] para que se possa comparar com texto ASCII, o qual também é convertido para um descritor utilizando-se o mesmo conjunto de características. Finalmente, no quarto e último estágio é realizada a comparação dos descritores extraídos da imagem da palavra com o gerado a partir do texto ASCII, a fim de obter uma medida de similaridade que é utilizada para decidir se a imagem da palavra corresponde a palavra procurada. Na Figura 3.1 pode-se visualizar os estágios do método proposto e um exemplo de seu funcionamento.

Tendo como objetivo possibilitar a busca de palavras em imagens de documentos, busca-se a capacidade de fazer a correspondência parcial entre imagens de palavras, ou seja, identificar um subconjunto de características correspondente à palavra procurada dentro de um conjunto maior correspondente a uma palavra derivada da procurada. Para

[1]Conjunto de símbolos ou números que representa a imagem de uma palavra em um determinado conjunto de características.

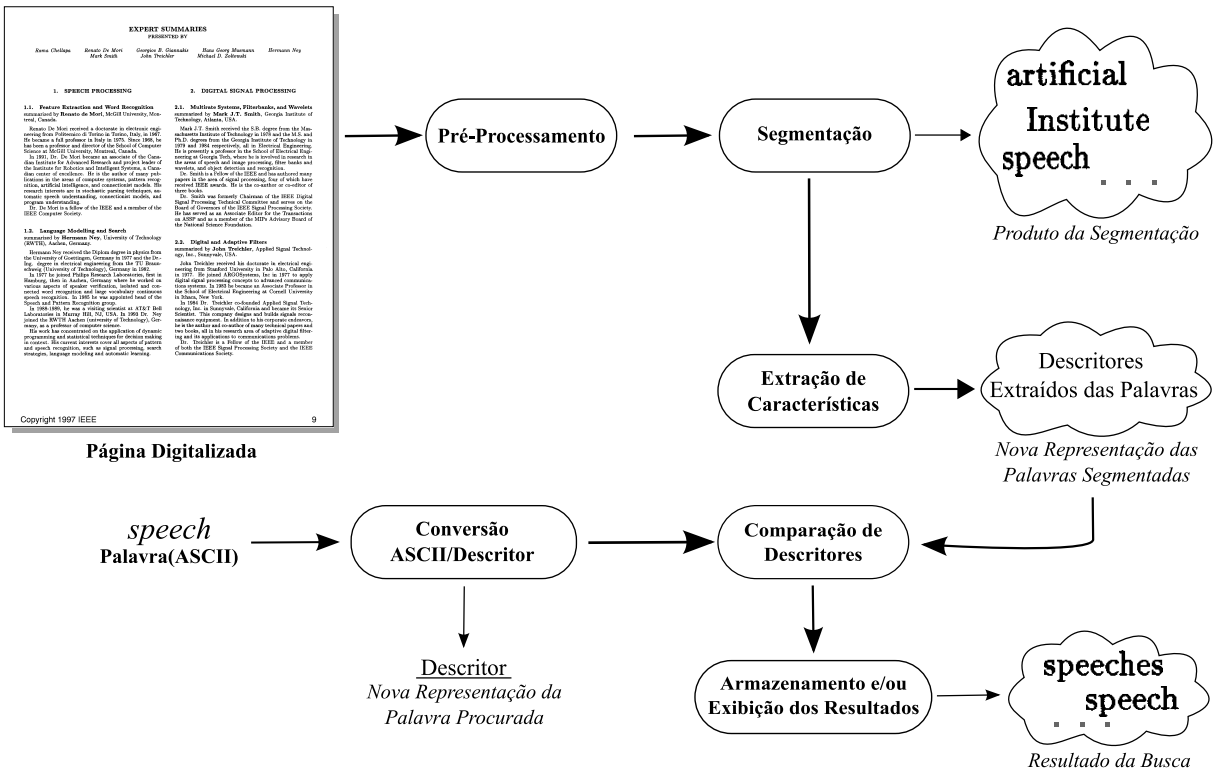

Figura 3.1: Visão geral do método desenvolvido; exemplo de busca da palavra "speech".

tanto, segmenta-se a imagem de uma página em palavras e extrai-se das mesmas um descritor que a represente. A palavra que se deseja encontrar também é convertida em um descritor e submetida juntamente com cada palavra detectada na página ao algoritmo de *comparação inexata de características* (LOPRESTI; ZHOU, 1996), o qual utiliza programação dinâmica para calcular a similaridade entre dois descritores, ou parte deles.

Imagens de documentos comumente apresentam diferenças no formato dos caracteres, seja por diferenças na fonte utilizada ou seja por ruídos. Diante desse problema, o método proposto tem a capacidade de encontrar palavras que apresentem diferenças em determinados itens do descritor. Além disso, trabalhou-se com palavras onde existam caracteres que se tocam, sem a necessidade de uma etapa de segmentação de caracteres.

Após a implementação do conjunto de características proposto por Lu e Tan e da devida avaliação deste, novas características foram avaliadas e comparadas com o método recém implementado. Para tanto, características descritas nos trabalhos de Arica e Yarman-Vural (2000) e Rath e Manmatha (2003) foram implementadas e adequadas ao método de comparação de descritores. Estes novos conjuntos de características serão abordados na Seção 3.4.

3.1 Obtenção das Imagens

O formato de arquivos PDF é utilizado como fonte primaria de obtenção das imagens processadas. Esse formato é utilizado tanto para armazenar texto formatado como imagens de documentos digitalizados e vem se tornando um dos formatos mais populares para a apresentação de documentos digitais. Várias empresas e organizações estão utilizando este formato, por exemplo, o IEEE realizou a digitalização de vários artigos que não estavam mais disponíveis no formato digital e armazenou as imagens no formato PDF. Outro exemplo é a distribuição dos anais da conferência ICASSP'97 que contêm duas versões de cada artigo nela publicado, sendo uma digital e outra digitalizada a partir da impressão da primeira, ambas no formato PDF.

A aquisição das imagens analisadas é realizada através da extração de imagens de arquivos no formato PDF. Esta etapa é realizada com o auxilio do software Xpdf (2005), o qual sofreu pequenas modificações para ser integrado ao sistema desenvolvido. Assim, é possível extrair as imagens diretamente do sistema. O formato PDF é utilizado devido ao fato de que documentos digitalizados são comumente armazenados neste formato por conveniência e organização.

3.2 Pré-Processamento

Na etapa de pré-processamento a imagem analisada é convertida para o formato binário (preto e branco), descartando a informação de níveis de cinza ou cor presente na imagem, este processo é chamado de binarização. O método utilizado para efetuar a binarização da imagem é o desenvolvido por Otsu (1978), devido ao seu reconhecido desempenho e adaptabilidade a vários tipos de imagens. A binarização da imagem é utilizada para facilitar e acelerar o processamento da imagem, visto que após binarizar a imagem tem-se apenas dois valores para representar um pixel, um representando o branco e outro representando o preto. Ou seja, dois níveis são suficientes para representar imagens de documentos onde se está interessado apenas no texto, sendo que um nível representa o texto e o outro representa o fundo da imagem.

Outra técnica utilizada é a suavização de contornos apresentada por Suen et al. (1992). Este processo consiste em corrigir imperfeições causadas nos caracteres do texto durante os processos de digitalização e binarização. Estas imperfeições prejudicam o desempenho do processo de extração de características, de modo que alteram as bordas internas e

?	1	?
1	0	1
?	1	?

(a)

?	1	?
1	0	1
0	0	1

(b)

1	1	0
1	0	0
1	1	0

(c)

0	0	?
0	1	1
0	0	1

(d)

0	0	0
0	1	1
0	0	?

(e)

0 - Fundo
1 - Primeiro Plano
? - Não considera

Figura 3.2: Mascaras utilizadas para realizar a suavização de contornos. Em (a), (b) e (c) o pixel central é mudado para 1, em (d) e (e) para 0.

Figura 3.3: Resultado da suavização de contornos em um caractere colhido de um dos documentos analisados.

externas do traçado dos caracteres. Utilizou-se as máscaras ilustradas na Figura 3.2 e as suas rotações em 90°, 180° e 270°. Estas máscaras são "deslizadas" sobre a matriz de zeros e uns correspondente à imagem binarizada e quando todos os valores da mascara coincidem com a posição atual na matriz da imagem o pixel central é mudado de zero para um ou vice versa. O símbolo "?" na representação da máscara indica que este valor da matriz é ignorado. Este processo é repetido por três vezes ou até que não haja alterações na imagem.

Na Figura 3.3 pode-se visualizar a imagem de um caractere com ruídos causados no processo de digitalização e o resultado obtido após a aplicação da suavização. Pode-se visualizar uma redução das imperfeições no contorno do caractere, realizada através do preenchimento de lacunas e da exclusão de pixels. Porém, em alguns casos este processo pode gerar deformações na forma do caractere que ao contrário de ajudar acabam por prejudicar o desempenho do método. Assim, a utilização da técnica deve ser avaliada com cuidado.

O pré-processamento utilizado é mínimo, visto que este projeto não pretende adentrar nas áreas de filtragem de imagens ruidosas, correção de inclinação e outras correções aplicadas a imagem de documentos. Além disso, os softwares mais recentes utilizados no processo de digitalização submetem os documentos a algumas correções após a captura da imagem, o que torna as imagens praticamente prontas para o processamento. Porém, o desempenho do método está diretamente ligado a qualidade das imagens. Ainda que haja a preocupação de se utilizar características robustas perante a imagens ruidosas, é muito difícil trabalhar com certos tipos de ruídos (alguns dos quais estão relacionados na

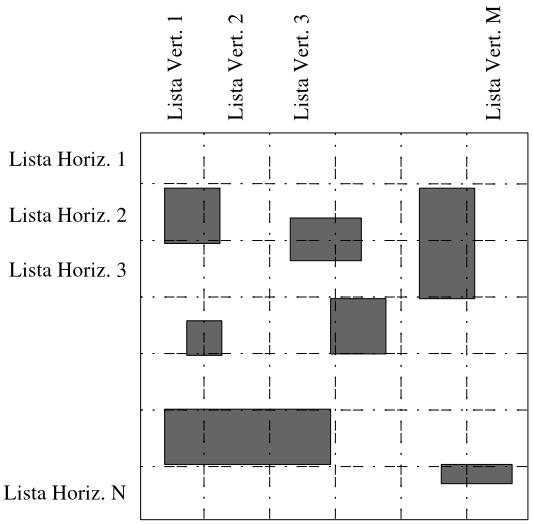

Figura 3.4: Divisão da página em listas verticais e horizontais. Blocos representam os componentes conexos detectados (imagens, tabelas, caracteres, etc.).

Seção 4.5 do Capítulo 4). Assim, a qualidade dos equipamentos utilizados na impressão e digitalização dos documentos são fatores incidentes nos resultados de precisão e revocação do método proposto.

3.3 Segmentação

A etapa inicial do processo de segmentação consiste em detectar os componentes conexos (CCs) da imagem. Um CC nada mais é do que uma área retangular que engloba um conjunto de pixels pretos (podem ser brancos dependendo do que se está procurando) onde pode-se chegar a qualquer pixel partindo de outro pixel qualquer, pertencente ao mesmo conjunto, sem passar por um pixel branco. Cada CC encontrado é submetido a um filtro que avalia sua forma (relação *altura vs largura*) e seu preenchimento (relação de *pixels pretos vs brancos*), para eliminar possíveis linhas de tabelas, figuras gráficas e ruídos presentes na imagem analisada.

Cada CC que passar pelo filtro de forma e preenchimento tem sua posição e dimensões armazenadas. Devido a grande quantidade de CCs que podem ser encontrados em uma página, deve-se ter o cuidado de usar uma estrutura que possibilite acessá-los de forma eficiente. Assim, resolveu-se utilizar a estratégia de dividir a página em várias listas horizontais e verticais (50 pontos de largura para cada lista) onde cada lista contém os CCs que fazem parte de uma determinada zona da página. As regras dessa estrutura, a qual pode ser visualizada na Figura 3.4, são as seguintes:

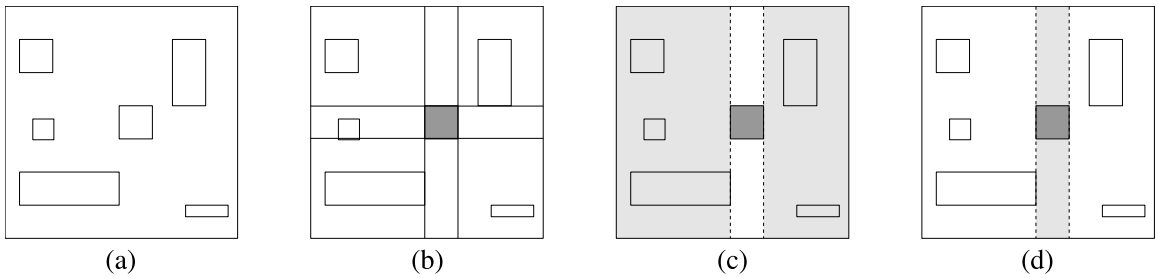

Figura 3.5: Alteração do método de detecção de espaços para evitar recálculo. (a) Imagem com os componentes conexos delimitados; (b) Seleção do pivô; (c) Criação dos sub-retângulos esquerdo e direito; (d) Criação dos sub-retângulos superior e inferior.

1. Um CC faz parte de pelo menos uma lista vertical e uma horizontal;
2. CCs que fazem parte de mais de uma zona devem fazer parte de mais de uma lista;
3. Se o retângulo de um CC intercepta outro eles devem ser unidos em um único CC que é re-inserido na lista.

A próxima etapa no processo de segmentação utilizado é a detecção das linhas. Para capacitar o sistema a segmentar as linhas de documentos que apresentem mais de uma coluna de texto e diferentes formatos é utilizada uma variação do método proposto por Breuel (2002), o qual busca espaços em branco na página a ser segmentada. Espaços em branco normalmente são encontrados entre colunas de texto facilitando a delimitação das linhas.

Com o intuito de evitar o recálculo uma variação do algoritmo proposto por Breuel (2002) foi desenvolvida e está ilustrada na Figura 3.5. A partir desse aperfeiçoamento foi possível diminuir drasticamente o tempo de execução. Essa variação não garante que a resposta seja o maior retângulo vazio da imagem, mas tem a característica de retornar retângulos com maior relação $altura/largura$ primeiro. Esse resultado é atrativo quando se deseja detectar colunas no texto. Ao invés da área do retângulo a seguinte função é utilizada para definir a prioridade Q de cada retângulo:

$$Q = largura(\delta + altura) \tag{3.1}$$

Quando δ for maior que zero os retângulos com menor relação $altura/largura$ serão levemente beneficiados na fila, fato esse que ajuda a diminuir o problema dos retângulos serem tão finos que se encaixem nos espaços entre as palavras.

Após detectar os espaços em brancos na imagem é possível realizar com mais precisão

Figura 3.6: Posição das linhas de apoio em uma imagem de palavra.

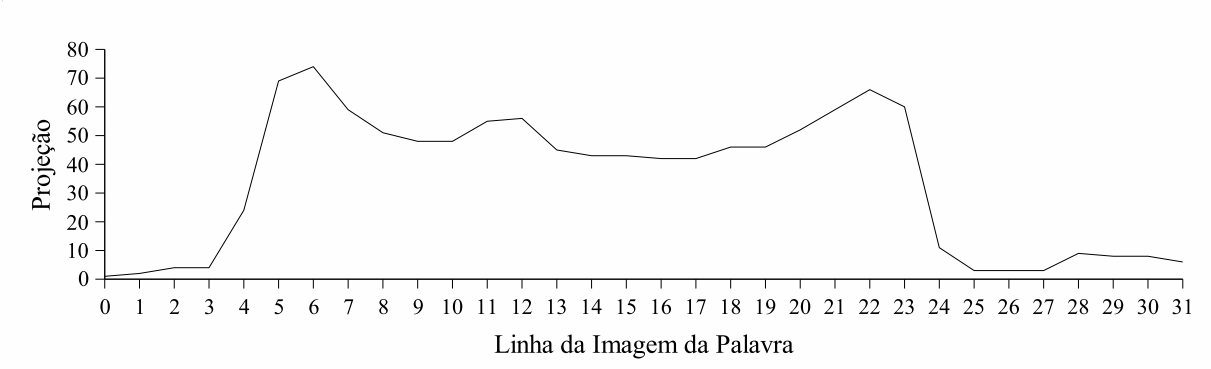

Figura 3.7: Projeção horizontal da imagem da palavra “system”.

a tarefa de detecção de linhas, tem-se apenas que unir os CCs adjacentes enquanto um espaço em branco não for encontrado.

Alguns dos métodos de extração de características utilizados fazem uso das linhas dos caracteres ascendentes e descendentes[2], as quais estão ilustradas na Figura 3.6. Para determiná-las em uma linha de texto é possível utilizar a projeção do histograma horizontal. Esta técnica consiste em contar o número de pixels pretos em cada linha de pixels da imagem. A seguir, as duas linhas onde esta contagem é máxima são encontradas e a partir delas delimita-se as linhas de ascendentes e descendentes. Por exemplo, na Figura 3.7 têm-se a projeção horizontal da imagem da palavra “system”, onde as linhas de ascendentes e descendentes estão posicionadas nas linhas 6 e 22, respectivamente. Para utilização deste método é fundamental que as palavras em uma linhas do documento estejam alinhadas horizontalmente, existem vários métodos existentes para corrigir a inclinação de documentos, porém nenhum deles foi utilizado neste trabalho visto que os documentos utilizados não apresentam inclinação.

A última etapa necessária na segmentação da página é a segmentação das palavras, visto que não se pretende trabalhar com segmentação ao nível de caractere, devido a problemas causados por caracteres que se tocam. Dado que M é igual a mediana das distâncias entre CCs adjacentes em uma determinada linha, pode-se determinar se dois

[2]Normalmente as linhas de *ascendentes* e *descendentes* são um produto do algoritmo de extração das linhas do texto.

CCs pertencem a mesma palavra verificando se a distâncias entre eles é menor ou igual a αM, onde α é um fator definido experimentalmente. A mediana foi utilizada por fornecer, na maioria dos casos, a distância mais frequente entre caracteres adjacentes em uma linha de texto, a média das distâncias não foi utilizada por estar sujeita a distorções causadas por um espaço muito grande existente na linha de texto analisada.

Finalizado o pré-processamento e a segmentação do texto em linhas e palavras obtém-se um ambiente pronto para receber os métodos de extração e comparação de descritores, os quais obrigatoriamente devem fornecer as seguintes funcionalidades:

- Converter uma palavra ASCII para um descritor χ;
- Converter a imagem de uma palavra para um descritor β;
- Comparar χ com todos os sub-descritores contidas em β e fornecer o maior valor de similaridade encontrado.

3.4 Extração de Características

Conforme citado anteriormente o método utilizado para fazer a busca por palavras trabalha a nível de descritores formado com características extraídas da imagem, evitando a conversão da imagem para o formato textual via OCR. Ou seja, a palavra procurada é convertida para um descritor e comparada com o descritor extraído das palavras existentes na imagem processada. Assim, o processo de extração e comparação de descritores pode ser considerado o mais importante deste processo, já que lhe foi dada a responsabilidade de aproximar descritores gerados a partir da imagem da palavra dos gerados a partir de uma cadeia de caracteres ASCII.

A definição das características que foram extraídas das imagens das palavras foi uma das tarefas mais difíceis do método proposto. Características propostas em trabalhos na área de reconhecimento de caracteres foram adaptadas ao método. Porém, durante a avaliação de novas características alguns pontos foram priorizados:

- Tolerância a caracteres que se tocam. É necessário que haja uma preocupação com palavras que apresentem caracteres que se tocam em um ou mais pontos, visto que este efeito pode ser causado pela fonte utilizada ou por baixa qualidade da imagem digitalizada. Além dos caracteres que se tocam, deve-se prever os casos onde haja

sobreposição de caracteres, mais comum em seqüências como “VA” e “To” onde parte da primeira letra fica sobre a segunda;

- Invariância ao tamanho e a posição da palavra. Uma característica deve ser a mesma para o mesmo caractere em diferentes tamanhos de fonte e em diferentes posições da palavra(início, meio ou fim);

- Invariância ao tipo de fonte. A total invariância ao tipo de fonte é um desafio grandioso e um problema que ainda não tem solução. Porém, deve-se buscar características que tolerem algumas modificações na forma dos caracteres causadas pela fonte utilizada, como a serifa que é apresentada nas extremidades de alguns caracteres em certos tipos de fonte;

- Robustez a ruídos. Imagens digitalizadas não são perfeitas e os ruídos são uma constante que afeta diretamente o desempenho de qualquer sistema que se propõe a analisar imagem de texto. Por isso procurou-se características o mais robustas possíveis;

- Representação particionada da imagem. Devido ao fato de que se deseja encontrar palavras dentro de palavras, é necessário que se tenha características que possibilitem a comparação de partes da palavra (geralmente colunas de pixels), de forma que se possa identificar uma palavra no corpo de outra palavra que tenha comprimento maior ou igual a ela.

A seguir serão apresentados os conjuntos de características analisados. Entre eles podemos destacar os conjuntos LRPS (*Left-to-Right Primitive String*), AYV (*Conjunto de características baseado nas características desenvolvidas por Arica e Yarman-Vural*) e ULTC (*Upper and Lower profiles and Transitions Count*).

3.4.1 Conjunto de Características LRPS

O primeiro conjunto de características implementado e avaliado foi o LRPS (descrito na Seção 2.2 do Capítulo 2) proposto por Lu e Tan (2004). Cada característica deste conjunto é formada por um par de atributos (σ, ω), onde σ é o *Line-or-Traversal Attribute* (LTA) e ω é o *Ascender-and-Descender Attribute* (ADA). A extração do primeiro é dividida em duas partes: análise de linhas retas e análise das transições preto/branco que não pertencem a uma linha reta. O segundo atributo é obtido através da análise da posição

da primitiva(linha reta ou transição) com relação as linhas de apoio(ascendentes e descendentes) detectadas na imagem da palavra. Estas características levam em consideração aspectos estruturais da imagem da palavra e são invariantes a escala e translação. Além disso, estas características apresentam certa invariância à fonte utilizada para imprimir o documento, dado que levam em conta aspectos relativos a fontes com ou sem serifa.

Após a implementação e avaliação do conjunto de características LRPS iniciou-se a avaliação de novas características. As características avaliadas foram baseadas em trabalhos recentes nas áreas de reconhecimento de caracteres impressos, manuscritos e indexação de imagens de documentos.

3.4.2 Conjunto de Características LRPS Modificado

O método de extração de características utilizado para gerar o conjunto de características LRPS foi modificado para tentar deixá-lo mais robusto. Em uma das modificações foi inserida a condição que determinava que uma característica deveria ser detectada em duas colunas vizinhas para ser inserida no descritor. A modificação consistiu em ignorar características isoladas nos descritores extraídos da imagem da palavra. Ou seja, se V representa o descritor de uma palavra então o atributo LTA de V[i] deveria ser igual ao de V[i-1] para V[i] ser considerado. Esta variação do conjunto de características tem como objetivo diminuir o impacto causado por ruídos. Uma característica obtida em uma coluna da imagem da palavra com ruído normalmente é diferente da coluna anterior e da coluna posterior. Por outro lado, características extraídas de colunas de imagens de caracteres com pouco ou nenhum ruído normalmente se repetem uma ou mais vezes devido a resolução da imagem. Esta variação do conjunto de características LRPS foi desenvolvida durante a pesquisa realizada neste trabalho e será referenciada como LRPS Modificado no restante do documento.

3.4.3 Conjunto de Características AYV

A idéia da divisão das linhas e colunas da imagem de um caractere em regiões, proposta por Arica e Yarman-Vural (2000), é aqui utilizada para a criação de novas características. A idéia foi utilizada para o desenvolvimento de características empregadas no reconhecimento de caracteres manuscritos. Estas características não puderam ser empregadas integralmente ao método aqui proposto, devido ao fato de terem sido desenvolvidas para caracteres isolados e não para palavras. Além disso, alguns procedimentos como

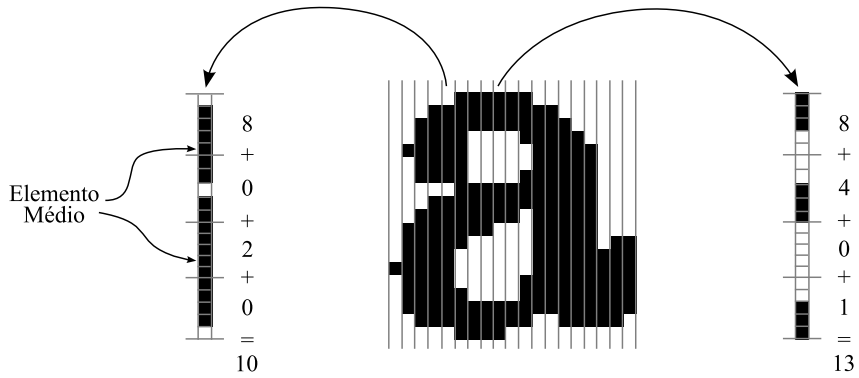

Figura 3.8: Divisão das colunas em regiões e o cálculo da característica das colunas 5 e 9 do caractere “a”.

a normalização da imagem da palavra em uma janela de tamanho pré-fixado não foram utilizados porque podem provocar distorções na estrutura da palavra.

O processo de extração de características baseado na proposta de Arica e Yarman-Vural está ilustrado na Figura 3.8. Neste processo cada coluna da palavra é analisada da seguinte maneira:

- A coluna é dividida em quatro regiões iguais (a diferença de 1 pixel no tamanho das regiões pode ocorrer devido a divisão inteira);
- Cada região recebe o valor 2^i, onde $i = \textit{índice\_da\_região} - 1$;
- Cada seqüência consecutiva de pixels pretos tem seu elemento médio M encontrado e o valor da região a que M pertence é somado ao resultado final.

A soma do valor das regiões que contêm um elemento médio corresponde ao valor da característica atribuída à coluna analisada. Esta característica é então inserida no descritor da palavra somente se ela for diferente da característica anteriormente inserida.

Arica e Yarman-Vural utilizaram a característica descrita acima e mais três características que seguem a mesma lógica porém analisando as diagonais e linhas verticais da imagem de um caractere para efetuar o reconhecimento de caracteres manuscritos. Obtendo como resultado taxas de reconhecimento de caracteres manuscritos em torno de 90% utilizando apenas duas das quatro características propostas por eles. Perante a estes resultados, decidiu-se avaliar a utilização da característica descrita acima no método proposto neste trabalho. Apesar de se utilizar apenas uma característica, acredita-se que a mesma seja suficiente para uma boa representação de uma coluna da imagem de um caractere e conseqüentemente suficiente para a representação da imagem da palavra como

um todo. Outro fator que motiva a sua utilização é a a facilidade de implementação, visto que permite uma rápida avaliação da mesma.

3.4.4 Conjunto de Características ULTC

O contorno externo de uma palavra pode dizer muito a respeito dela. Pelo menos é o que mostra o estudo realizado por Rath e Manmatha (2003) onde os contornos superiores e inferiores da imagem da palavra foram utilizados como características para a indexação de documentos manuscritos. Assim, adaptou-se tais características para o sistema desenvolvido e o método de comparação. Além disso, utilizou-se a característica de número de transições preto/branco para fornecer informações a respeito da forma interna da imagem da palavra. Buscou-se com a união dessas três características, contorno superior, inferior e número de transições preto/branco, obter uma representação consistente da palavra analisada.

A característica de contorno superior de uma coluna é obtida através do cálculo da distância existente entre a borda superior da palavra e o pixel mais próximo dela. Da mesma maneira, a característica de contorno inferior é obtida através do cálculo da distância existente entre a borda inferior da palavra e o pixel mais próximo dela. A Figura 3.10 e a Figura 3.11 ilustram de forma gráfica o valor das características de contorno superior e inferior para todas as colunas da imagem da palavra “problem”(Figura 3.9). Ambas as características são normalizadas entre 0 e 1. Para facilitar a visualização do gráfico o valor da característica de contorno superior foi invertido.

A característica de número de transições é obtida através da contagem do número de traços em uma coluna. Ou seja, para cada transição de um pixel representante do fundo da imagem(geralmente branco) para um pixel representante dos objetos da imagem(geralmente preto) o número de transições é incrementado. Assim, considerou-se um número máximo de quatro transições em uma coluna, sendo o valor da característica o número de transições divido pelo máximo permitido, gerando um valor entre 0 e 1. A Figura 3.12 apresenta de forma gráfica o valor atribuído para esta característica em todas as colunas da palavra “problem”.

No conjunto de características formado pelas características de contorno superior, contorno inferior e número de transições cada coluna da imagem da palavra foi representada por um vetor composto de três características com valores entre 0 e 1. Este conjunto de características será referenciado no resto do documento como ULTC.

Figura 3.9: Imagem da palavra “problem” extraída de um dos documentos analisados.

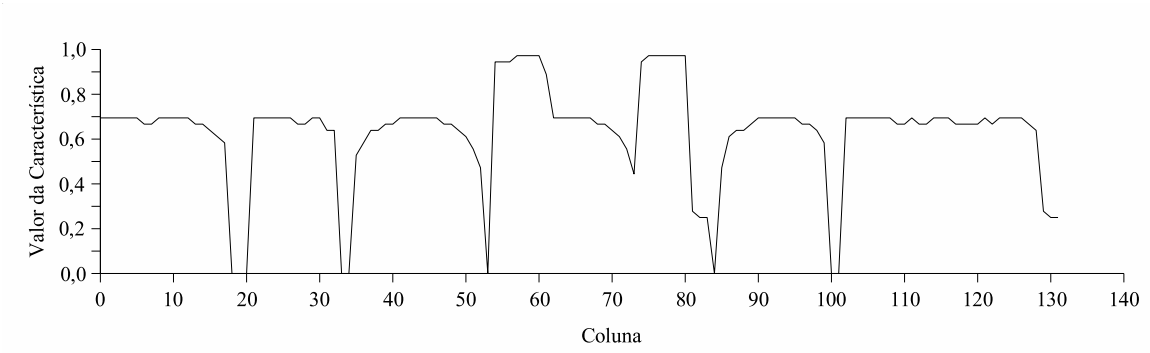

Figura 3.10: Característica de contorno superior(invertida) ao longo da palavra “problem”.

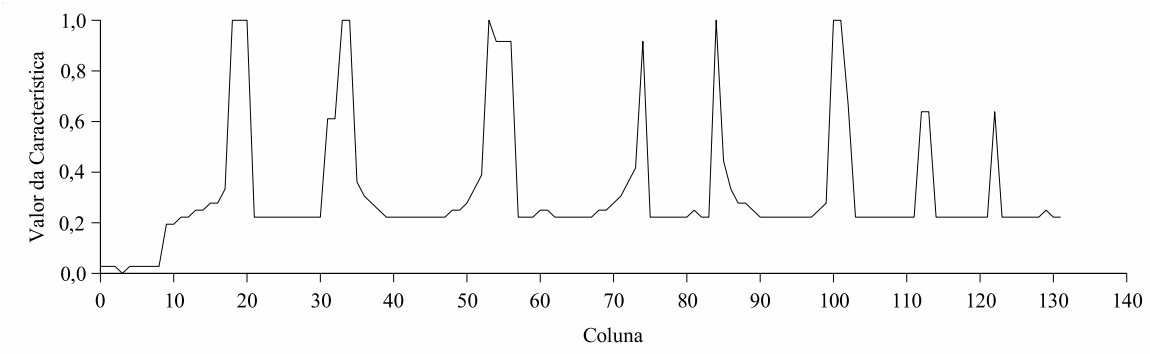

Figura 3.11: Característica de contorno inferior ao longo da palavra “problem”.

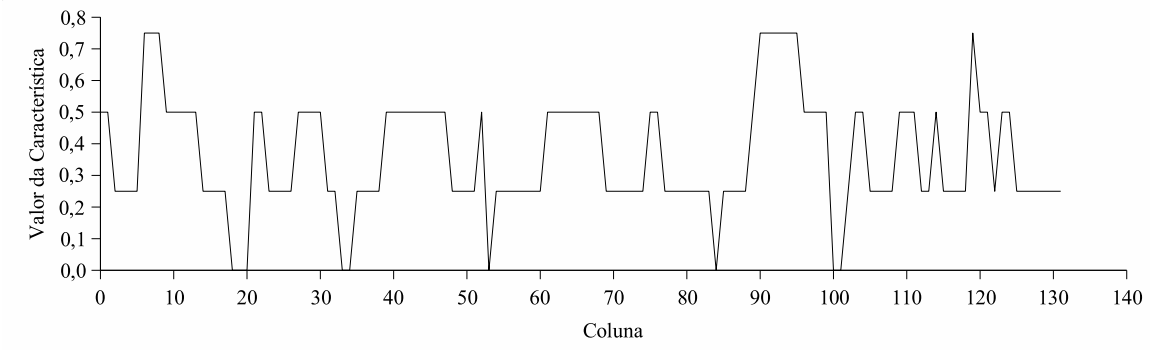

Figura 3.12: Número de transições(normalizado) ao longo da palavra “problem”.

3.4.5 Conjunto de Características ULTC Modificado

Durante a pesquisa implementou-se um novo conjunto de características, formado pelo conjunto ULTC e uma característica correspondente ao número de pixels pretos da coluna. Esta última característica foi a mais simples das implementadas e consiste em contar o número de pixels pretos em uma coluna e normaliza-lo entre 0 e 1. O objetivo dessa modificação é aumentar a precisão do conjunto de características ULTC, visto que duas das três características que fazem parte do conjunto dizem respeito a forma externa da imagem palavra, contando apenas com a características de número de transições para representar a forma interna da imagem da palavra. A nível de caractere, a forma externa pode não ser suficiente para distinguir caracteres como "o" e "a". Assim, faz-se a inclusão de uma característica que representa a forma interna da palavra e que juntamente com a característica de número de transições espera-se obter um aumento da precisão do método. Este conjunto de características será referenciado no restante do documento como ULTC Modificado.

3.5 Conversão ASCII/Descritor

A conversão de uma cadeia de caracteres ASCII para um descritor pode ser feita através de uma tabela gerada manualmente. Por exemplo, no conjunto de características LRPS a conversão utilizada pelos autores do método fez uso de uma tabela (ilustrada na Tabela 2.1 do Capítulo 2) gerada manualmente a partir da observação de cada caractere. Porém, neste projeto avaliou-se também a criação desta tabela de forma automática a partir da extração de características de uma imagem com os modelos de caractere, a qual contêm todos os caracteres do alfabeto inglês, minúsculos e maiúsculos, no tamanho de 48 pixels. Assim, torna-se mais rápido o processo de criação da tabela sempre que o método for modificado. Além disso, essa metodologia permite que diferentes tabelas sejam criadas a partir de diferentes modelos de caracteres criados com fontes distintas. Todos os modelos foram criados a partir de um editor de imagens comum, sendo estes sintéticos para que não se tornem específicos para um determinado conjunto de documentos.

A conversão de uma cadeia de caracteres P para o descritor correspondente é realizada da seguinte maneira: para cada caractere C em P inclui-se no descritor os valores correspondentes as características de C presentes na tabela de conversão gerada manualmente ou automaticamente. Além disso, entre as características correspondentes a dois caracteres vizinhos em P insere-se as características correspondentes a representação de

Caracteres ASCII	**t**		**o**		**p**
Descritor	(n,x)(l,A)(o,x)	(&)	(c,x)(o,x)(c,x)	(&)	(l,D)(o,x)(c,x)

Tabela 3.1: Descritor que representa a palavra *top* utilizando o conjunto de características LRPS.

uma coluna em branco, para simular os espaços entre caracteres existentes na imagem da palavra.

Na Tabela 3.1 está exemplificada a representação de um descritor gerado pelo conjunto de características LRPS a partir da cadeia de caracteres "top". No conjunto de características LRPS o simbolo "&" representa um espaço em branco entre dois caracteres. Durante a conversão de uma cadeia de caracteres para um descritor o símbolo "&" é inserido entre o primeiro e o último símbolo de cada caractere adjacente. Neste conjunto de característica os símbolos que representam os caracteres estão em uma tabela pré-definida, a qual está apresentada na Tabela 2.1 (Capítulo 2, Secção 2.2). Os símbolos nada mais são do que representações de características que podem ser extraídas da imagem da palavra.

Quando utiliza-se o conjunto de características LRPS, a tabela de conversão é mantida como no método original, gerada manualmente. Porém, nos outros conjuntos de características implementados utiliza-se tabelas geradas automaticamente através de imagens com modelos de caractere. Durante a geração da tabela de conversão as imagens são segmentadas de forma a isolar as palavras, assim como uma página do documento também é segmentada. Cada imagem de palavra segmentada é submetida ao método de extração de características em questão para gerar o descritor da palavra. Este descritor, por sua vez, é processado para identificar características correspondentes as colunas em branco, existentes entre caracteres da imagem, estas características servem como delimitadores de caractere utilizados para saber quais características correspondem a determinado caractere. Com a posse das características de todos os caracteres, basta organizá-las em uma tabela, que é posteriormente utilizada para conversão das palavras chave fornecidas ao sistema.

Os modelos de caractere são gerados a partir de duas fontes: *Times New Roman* e *Sans*. A primeira fonte contêm caracteres com serifa e é comumente encontrada em documentos. A segunda não contêm serifa e é utilizada para possibilitar a detecção de palavras em fontes sem a mesma.

O conjunto de características ULTC faz uso de duas tabelas de conversão ao mesmo tempo. Assim, a estrutura de dados utilizada para armazenar o descritor de uma palavra chave tem que ser duplicado, para que armazene dois descritores: com e sem serifa. Além

disso, o método de comparação pode ser acionado duas vezes para cada par de palavras comparado (se a primeira comparação retornar um valor de similaridade maior ou igual ao limiar definido então apenas uma chamada será realizada).

3.6 Comparação de Descritores

O processo de comparação de descritores é responsável por grande parte do tempo de processamento do sistema. Seu desempenho é O(nmp), onde p corresponde ao total de palavras no documento e n m correspondem ao tamanho do descritor gerado a partir da palavra que se deseja encontrar e da imagem da palavra analisada no momento, respectivamente. Dessa maneira qualquer aumento, mesmo que pequeno, no tamanho dos descritores acarreta em perda de performance, visto que é necessário acionar o método para praticamente todas as palavras existentes nos documentos analisados. Dado dois descritores, representando a imagem de uma palavra no documento e a cadeia de caracteres que se deseja encontrar, a função desse método é retornar a similaridade entre estes dois descritores na forma de um valor entre 0 e 1. A partir da definição de um valor mínimo para a similaridade é possível dizer se estes dois descritores correspondem a mesma cadeia de caracteres.

O método desenvolvido durante a pesquisa é baseado no método utilizado por Lu e Tan (2004). Utilizou-se este método devido a sua capacidade de fazer a correspondência parcial entre dois descritores. Ou seja, ser capaz de detectar uma sub-palavra dentro de outra, como “feliz” dentro da cadeia de caracteres “infeliz”. Procurou-se manter esta qualidade do método de comparação de descritores durante o processo de implementação de novas características.

Cada palavra encontrada no documento em que a relação *altura/largura* é menor ou igual a da palavra fornecida, é submetida a comparação com esta. Assim, quando deseja-se procurar por uma sub-palavra, todas as palavras maiores que a sub-palavra são analisadas. Esta filtragem é necessária para agilizar a busca, visto que um documento pode ter uma quantidade relativamente grande de palavras.

O método de comparação utilizado nos conjuntos de características LRPS e AYV deve sofrer alterações para se adequar as características implementadas no conjunto ULTC. Neste último conjunto de características, os valores das características obtidas para cada coluna não são mapeados para valores discretos e sim armazenados em um vetor de características. Assim, cada coluna possui o seu próprio vetor de características que é

comparado ao de outra coluna utilizando-se a distância D calculada através da equação 3.2. Esta metodologia tem como aspecto positivo o fornecimento intrínseco de uma métrica para julgar a distância entre duas colunas, fato esse que não ocorre nos outros dois métodos onde a distância entre características deve ser tabelada. Porém, a implementação de um algoritmo de comparação que trabalhe com a identificação de sub-palavras torna-se mais complexa, porque agora é preciso identificar a distância máxima aceitável entre dois vetores de características.

$$D_{ULTC} = \sum_{k=1}^{N} (a[k] - b[k])^2 \tag{3.2}$$

onde N corresponde ao número de características extraídas por coluna, ou seja, 3 para o conjunto ULTC e 4 para o ULTC Modificado, e a e b correspondem aos vetores de características das colunas comparadas.

A comparação de dois vetores de características, representantes de duas colunas da imagem do caractere e extraídos utilizando o conjunto de características ULTC, precisa de um valor que indique qual a distância máxima positiva entre eles. Assim, quando a distância for maior que este limiar a pontuação somada no algoritmo de comparação de descritores será negativa. Quanto maior for a distância menor será a pontuação atribuída ao par de colunas analisado.

A comparação de dois descritores é feita através da técnica de *comparação inexata de características*. Através dessa técnica, analisa-se os descritores gerados pelo processo de extração de características, de modo a fornecer um valor de similaridade entre eles. Um dos pontos chaves deste método é a definição da pontuação atribuída a cada coluna comparada. No conjunto de características ULTC esta pontuação é obtida pelo cálculo da distância, descrita na equação 3.2, entre os vetores de características de duas colunas, visto que cada coluna é representada por um vetor de três características com valores entre 0 e 1. Porém, em alguns conjuntos de características essa abordagem não pode ser aplicada, devido ao fato do método de extração de características gerar valores discretos que não tem relação de distância. Nestes casos, a definição dessa pontuação é fixada no método.

No caso do conjunto de características LRPS atribui-se a mesma pontuação utilizada pelos autores do método. O valor D atribuído as diferentes possibilidade de comparação de dois pares de atributos a e b está relacionado na equação 3.3, onde σ corresponde ao atributo de linhas e transições (LTA) e ω corresponde ao atributo do posicionamento relativo as linhas de ascendentes e descendentes (ADA). Vale relembrar que o símbolo &

do conjunto LRPS simboliza uma coluna da imagem sem pixels pretos.

$$D_{LRPS} = \begin{cases} 2 & \text{se } \sigma_a = \sigma_b \text{ e } \omega_a = \omega_b \\ 0 & \text{se } \sigma_a = \sigma_b \text{ e } \omega_a \neq \omega_b \text{ ou } \sigma_a \neq \sigma_b \text{ e } \omega_a = \omega_b \\ -1 & \text{se } \sigma_a \neq \sigma_b \text{ e } \omega_a \neq \omega_b \text{ e } \sigma_a = \& \text{ ou } \sigma_b = \& \\ -2 & \text{se } \sigma_a \neq \sigma_b \text{ e } \omega_a \neq \omega_b \end{cases} \tag{3.3}$$

Pontuação semelhante à do conjunto LRPS é utilizada quando da comparação das características do conjunto AYV. Porém, no conjunto AYV se tem apenas um atributo para cada coluna. O valor D atribuído nas diferentes possibilidades de comparação de duas características a e b do conjunto AYV está relacionado na equação 3.4, onde o ν representa a característica extraída por este conjunto e $\nu = 0$ é verdadeiro quando ν representa um coluna da imagem sem pixels pretos.

$$D_{AYV} = \begin{cases} 2 & \text{se } \nu_a = \nu_b \\ 0 & \text{se } \nu_a \neq \nu_b \text{ e } \nu_a = 0 \text{ ou } \nu_b = 0 \\ -2 & \text{se } \nu_a \neq \nu_b \end{cases} \tag{3.4}$$

O método de comparação de descritores utilizado no conjunto de características LRPS pode ser visualizado no Algoritmo 1(Capítulo 2). Uma variação desse algoritmo é utilizada para comparar os descritores gerados com as características do conjunto AYV. Além disso, este algoritmo segue a mesma linha de raciocínio do *DTW*, utilizado na comparação dos descritores gerados com o conjunto ULTC.

Após comparar os dois descritores, sendo um gerado a partir da imagem da palavra extraída do documento e o outro a partir da conversão dos caracteres ASCII da palavra, obtém-se uma medida de similaridade entre eles. Esta medida de similaridade S é comparada a um limiar λ, quando $S \geq \lambda$ considera-se que a palavra procurada esta contida na palavra correspondente ao descritor analisado. Assim, a palavra é marcada como coincidente e tem sua posição, página, documento e valor de similaridade armazenados para posterior conferência. O valor do limiar λ é atribuído experimentalmente, visando um equilíbrio entre precisão e revocação. Durante os testes observou-se que o valor de λ é diretamente proporcional à precisão e inversamente proporcional à revocação do método.

3.7 Considerações Finais

O pré-processamento é mínimo dado que os documentos analisados não apresentam inclinação e ruídos em excesso. Porém, no futuro pode haver a necessidade de implementar novos métodos de filtragem e correção da imagem da página.

A segmentação utilizada é simples, porém eficaz, visto que apenas se deseja obter as linhas e palavras do texto sem se preocupar com características estruturais, como delimitação de parágrafos e seções.

Três conjuntos de características foram propostos aqui, cada um possui características distintas. No conjunto de características LRPS realiza-se uma análise de características estruturais da imagem da palavra e este é o conjunto de características mais complexo dos três implementados. Espera-se obter uma maior precisão com este conjunto, devido ao grande número de análises realizadas nas linhas dos traçoes dos caracteres e nas transições pertencentes a imagem da palavra. O conjunto AYV é o mais simples, porém a idéia da divisão em regiões das colunas da imagem da palavra é interessante e espera-se com ela obter uma boa representação de cada padrão de coluna presente na imagem. Finalmente, o conjunto ULTC apresenta característica que visam a forma externa da imagem da palavra e espera-se com isso obter certa robustez a ruídos.

Através da avaliação dos diferentes conjuntos de características implementados é possível determinar qual obteve melhor desempenho perante um banco de dados de testes utilizado. O processo de avaliação de cada método está descrito no próximo capítulo juntamente com os resultados experimentais e a análise dos mesmos.

Capítulo 4

Resultados Experimentais

Neste capítulo é apresentada a metodologia utilizada para avaliar o desempenho do método proposto, uma análise de cada conjunto de características implementado assim como uma análise global dos resultados.

4.1 Banco de Imagens de Documentos

A realização dos testes dependeu da utilização de imagens de documentos que apresentassem uma versão textual com a informação da posição de cada palavra na imagem, para que a conferência das palavras encontradas fosse feita a partir da verificação da posição. Porém, não foi possível encontrar um banco de dados de imagens de documentos que se enquadrasse nessa restrição. Assim, foi necessária a criação de tal banco de dados. Este procedimento foi realizado através de um conjunto de artigos científicos digitalizados e armazenados no formato PDF. Ao todo foram utilizados 865 artigos publicados na ICASSP'97 (IEEE. . . , 1997), os quais foram copiados de um CD fornecido pela conferência. Cada documento contém entre duas e quatro páginas e é formado por texto em colunas podendo conter imagens, tabelas e fórmulas matemáticas. Os documentos foram digitalizados a uma resolução de 300dpi, o que é considerado suficiente para a maioria dos métodos existentes na área de processamento de imagens. Além disso, não foram observados documentos com problemas de inclinação do texto. A Figura 4.1 apresenta quatro páginas de um exemplar pertencente ao banco de imagens de documentos utilizado para realizar os testes.

A qualidade dos documentos existentes no banco de dados de testes varia de documento para documento. Em alguns documentos podem ser observados traços mais grossos

A DIGITAL PROCESSING SYSTEM FOR SOURCE LOCATION AND SOUND CAPTURE BY LARGE MICROPHONE ARRAYS *

Harvey F. Silverman[1] *William R. Patterson III*[1] *James L. Flanagan*[2] *Daniel Rabinkin*[2]

[1]LEMS, Division of Engineering, Box D, Brown University, Providence, RI 02912
[2]CAIP Center, Rutgers University, CoRE Building, Piscataway, NJ 08855-1390

ABSTRACT

The Huge Microphone Array(HMA) project started in February 1994 to design, construct, and test a real-time 512-microphone array system and to develop algorithms for use on it. Analysis of known algorithms showed that signal-processing performance of over 6 Gigaflops would be required; at the same time, there was a need for "portability", i.e., fitting into a small van. These tradeoffs and many others have led to a unique design in both hardware and software. This paper presents the design and its justifications. Performance data for a few important algorithms relative to usage of processing-capability, response latency, and difficulty of programming are discussed.

1. INTRODUCTION

The scope of the Huge Microphone Array (HMA) project goes substantially beyond the capabilities of earlier systems. No all-digital, real-time, intelligent, 512-microphone-array system has ever been designed, and the number of microphones is even larger than the 400 used in the analog system built at AT&T Bell Laboratories in the mid 1980's [1]. The system is nearing completion and will be initially used in three different sites. The reader is referred to [2] for a more detailed description, [3] for a recent general reference, [4] for the application of matched filtering, and to [5, 6] for location determination.

2. CRITERIA FOR DESIGN

A typical hierarchy for processing the signals from a microphone array is shown in Figure 2. Many of the algorithms to be implemented operate on frequency-domain data. Thus, the first processing after A/D conversion is the accumulation of short time segments, conversion to the frequency domain, and coding for transmission to other processors. The need to minimize cabling and noise problems, led to an early design decision to place the ADCs for groups of sixteen microphones in small boxes, called *microphone modules*, mounted near the array itself, remote from the console. Each box connects by a dual optical-fiber cable to the central console. As a result, only 32 duplex-fiber cables connect to the console, rather than 512 long, and potentially noisy, microphone cables (see Figure 1).

The sampling rate, frame length and the frame advance

*RESEARCH SUPPORTED BY NSF GRANT MIP-9314625

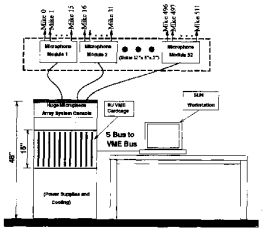

Figure 1. The Packaging of the HMA

must be selected to make this remote microphone-module design feasible. Sampling at 30kHz, a frame length of 51.2ms (1024 samples), and a frame advance of 25.6ms(512 samples) were chosen empirically as suitable for the algorithms and the architecture.

Figure 2 suggests that data from one microphone will usually be used by more than one processor. To meet this need and to minimize acquisition latency, the data transmission system simultaneously writes the data into local buffer memories on each processing board in the console. Any processors on the board can share one such frame buffer. Data transfers in and out of this memory are fairly fast. (17.92MW/s). Thus all microphone data are available to all processors, and data selection time is minimal.

Packetization in the microphone modules introduces the first delay(**latency**) between the input acoustic signal and any output acoustic signal. Summing the nominal delays given in Figure 2 indicates that over 125ms of latency is evident. This calculation also assumes that the architecture does not impose further delay because of the use of multiple processors and the need to pass data between processors within the single frames of delay shown. It is unfortunate that the time needed to produce an improved audio signal is

so significant. While an expected latency of from 100-500ms is likely to be acceptable for teleconferencing, or even for input to a recognizer, this delay would not be acceptable in direct applications such as audience pickup or direct public address. In real systems there is an additional source of latency should it be necessary to split the computation of any of the blocks of Figure 2 among several processors.

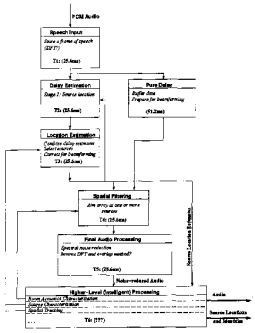

Figure 2. A Typical Data-Flow Structure in a Microphone-Array System (with Nominal Delay Times)

Frame-by-frame processing has another implication. In all the above it has been assumed that there is no delay due to **communications**. Each frame (here 25.6ms) must include time for the processor to receive its input and to return its output, during which the DSP microprocessors cannot do calculations. For computational efficiency, it must be possible to transfer one or more full frames of transformed audio signal in a time that is short compared to the frame interval. The HMA design uses high-speed serial links that can connect multiple pairs of boards simultaneously. The link rate is 2.38MW/s of 32-bit words. Transfers between processors on a single board are at 8MW/s.

Writing **software** for multiprocessor real-time systems is very difficult so a great deal of systems software is needed. The decision was made to make the hardware "smarter" and develop a *load and go* form of an operating system. The main hardware feature that affects the operating system is that data flow is not controlled by the DSP processors, but, instead, a set of supervisory processors sets up and starts DMA transfers independently of the DSP programs.

The HMA(Figure 3) meets the following design criteria:

1. At least 6Gflops/s of DSP computing performance is needed to do matched filtering in real-time for 512 microphones implying about 109 ADSP21020 processors.
2. Large, fast local memories are needed; at least 128K words each of data and program memory per processor.
3. Control channels are required; a low speed uplink to command the modules and a fast path to and from the workstation for booting the system, recording data, and controlling the system at a high level.
4. To minimize latency, there must be few limitations on the scheduling of data transfers.

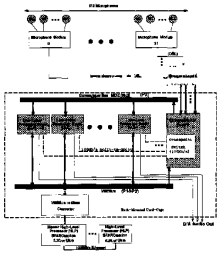

Figure 3. HMA System Block Diagram

3. HMA HARDWARE

The **Microphone Module** for the HMA is shown in Figure 4. Each module does, 1) the preamplification and biasing for 16 electret microphones, 2) analog-to-digital conversion of the microphone signals, 3) DFT transformation and compression or coding of the data, and 4) packetization and transmission of the data to the console unit. One should note that the module circuit board may also be populated for use as a personal computer attachment. In this mode, the system has the capability of a single ADSP21020. 33MHz signal processor with up to 16 microphones of input. This second mode of operation has proven to be quite valuable inside and outside of the University in a package called the **Brown Megamike**[7].

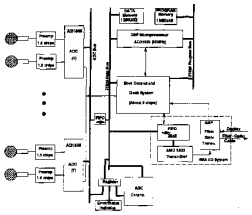

Figure 4. Block Diagram of Module

The **Commutator** system is the traffic manager for the microphone data. Its principle function is to communicate with the microphone modules and to multiplex their data outputs onto the commutator bus. Multiplexing is implemented with two, large Xilinx FPGA's. The commutator bus itself is a VME P1 backplane that has been modified to easily handle the one-way, 64-bit 17.92MHz transfers. The single board fits a standard VME Bus 9U rack (366.7mm high and 277mm deep) and also hosts an SGS-Thompson crosspoint switch that allows transfers between Low-Level Processing(LLP) boards at 2.38MW/s. With 12 LLP boards, this device can support up to 6 pairs communicating at the same time.

The **Low-Level Processing(LLP)** system consists of twelve boards (Figure 5). Each LLP board hosts eight DSP processing systems on daughter boards that mount on the rear. The daughter boards (7.2cmx11.9cm) are ten layers and contain an ADSP-21020 DSP processor, 10Mb of fast SRAM and support logic. The mother/daughter arrangement reduces the complexity of the LLP board and provides processor-system interchangeability.

The dataflow control on the LLP board is the HMA's most idiosyncratic feature. The ADSP-21020 has independent ports for its two memories allowing simultaneous I/O; the design uses a separate DMA controller and a supervisor processor to manage these data transfers. After powerup, the high-level processor (a SPARCstation) loads the supervisor static RAM with routing tables that the processor uses to control the DMA communication system. When the supervisor starts a DMA transfer that effects one of the ADSP21020 processors, the DMA controller requests both buses of the target processor, strobes its program memory in or out, and, simultaneously, loads microphone data into its data memory. The DSP processor programs have no influence on what data they receive or when, which implies that programs may be generic; exact duplicates may be used on the set of processors doing the same task but on different data.

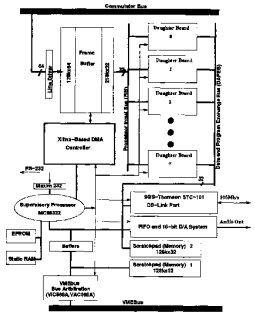

Figure 5. Block Diagram of Low-Level Processing Board

Each LLP board also has two scratchpad memories. Memory 1 is tightly coupled to the VME Bus port and is used as a mailbox for data entering or leaving that port. The VMEbus interface offers full bus master capability. Memory 2 is used for transfers between DSP processors on the same board at a net rate of 8MW/s. The timing of a frame is indicated in Figure 6. All the processors on all the boards run on the data of Frame $n-1$ (or earlier if buffered) while the data for Frame n is being loaded into the frame buffer. Any time an ADSP21020 DSP processor is not the target of a data transfer, it can do calculations. Since the transfer of data from a microphone takes only 28.2μs of the 25.6ms frame time, computational efficiency is very high.

4. SYSTEM SOFTWARE

The operating system for HMA has two major components. The first is an interactive set of tools (Application Tools for the HMA (ATHMA)) that develops a set of files that describe an application. These files contain 1) a program library (all the binary files for programs to run on LLP processors – one program from the library may be run on many LLP processors), 2) a mapping of programs to processors, 3) routing tables to be loaded into the memory of the supervisory processors to control DMA data transfers, 4) parameters to be passed to the DSPs at run-time, and 5) optionally, a program to run on the workstation after the HMA starts. The details of ATHMA are beyond the scope

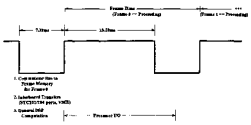

Figure 6. Typical Frame Timing on the LLP Boards

of this paper, but the reader is referred to [2]. The second component is a loader that uses the output files from ATHMA as input, loads the appropriate memories and registers of the HMA, starts the table execution on the supervisors, and, if needed, starts any required workstation programming.

5. SOME NUMBERS FOR A TYPICAL APPLICATION

An important application, locationing and beamforming, has been implemented on the Brown Megamike system for 16 microphones, giving the actual times for the implementation of this application on the 21020 DSP microprocessor [7]. In extrapolating to 512 microphones the following assumptions have been made:

- Brandstein's LI estimator [8, 5], which has a significant computational advantage for microphones grouped as *orthogonal quads* or sets of four microphones arranged as two pairs having the lines between the pairs intersect at right angles.
- The 512 microphones are spread in sub-arrays around the room and not just on a single wall. Experiments suggest that four orthogonal quads per sub-array is sufficient for accurately locating sources.
- The LI estimator, which computes from bearing lines, obtains its best estimates from the intersections of bearing lines from different sub-arrays.
- All microphones contribute to the beamformed signal.

Using the known times for all the algorithms, 36 processors are required (3 LLP boards) for this relatively simple application. Sixteen processors do time-delay estimation, four do the locationing and one computes the final position estimate. Two processors are used to check and correct the delay estimates. Twelve processors are needed for the beamforming in a three-deep tree, and one collects the final output. The latency is kept to under 150ms by the architecture, suitable for teleconferencing and other applications.

6. CONCLUSION

The design of a system that supports the use of an array of 512 microphones in real time has been described. One-hundred twenty-eight fast, floating-point signal processors give real-time performance for moderately complex algorithms. A straightforward example has illustrated the reasons for the decisions. After the hardware and software system is functional, the next step is to develop a set of useful applications. These include software for recording a database from 512 microphones, a practical matched-filtering and/or inverse-filtering beamformer, improved adaptive beamformers, talker-characterized trackers, and systems with advanced methods of noise-cancellation and reverberation reduction.

A project of this magnitude always involves many people. The authors would especially like to acknowledge the contributions of John Adcock, Michael Brandstein and Joseph DiBiase to the system and software design.

REFERENCES

[1] J. L. Flanagan, J. D. Johnson, R. Zahn, and G. W. Elko. Computer-steered microphone arrays for sound transduction in large rooms. *J. Acoust. Soc. Amer.*, 78(5):1508–1518, November 1985.

[2] H. F. Silverman, W. R. Patterson III, and J. L. Flanagan. The huge microphone array (HMA). LEMS Technical Report 156, LEMS, Division of Engineering, Brown University, Providence, RI 02912, November 1996.

[3] J. L. Flanagan and H. F. Silverman. Material for third biennial roundtable of microphone-array technology. LEMS Technical Report 151, LEMS, Division of Engineering, Brown University, Providence, RI 02912, October 1996.

[4] E. Jan, P. Svaizer, and J. Flanagan. Matched-filter processing of microphone array for spatial volume selectivity. In *Proceedings of ISCAS95*, pages 1460–1463. IEEE, 1995.

[5] M. S. Brandstein, J. E. Adcock, J. H. DiBiase, and H. F. Silverman. A closed-form method for finding source locations from microphone-array time-delay estimates. In *Proceedings of ICASSP-1995*, pages 3019–3022. Detroit, Michigan, May 1995.

[6] M. S. Brandstein, J. E. Adcock, and H. F. Silverman. A localization error criterion for microphone-array sensor placement. In *Proceedings of ICASSP-1996*, pages 901–904, Atlanta, GA, May 1996.

[7] J. E. Kassler. On the implementation of a real-time microphone-array audio source location algorithm with applications to video teleconferencing. Master's thesis, Brown University, Providence, RI, May 1996.

[8] M. S. Brandstein, J. E. Adcock, and H. F. Silverman. A practical time-delay estimator for localizing speech sources with a microphone array. *Computer, Speech and Language*, 9(2):153–169, April 1995.

Figura 4.1: Quatro páginas de um documento pertencente ao banco de imagens de documentos utilizado.

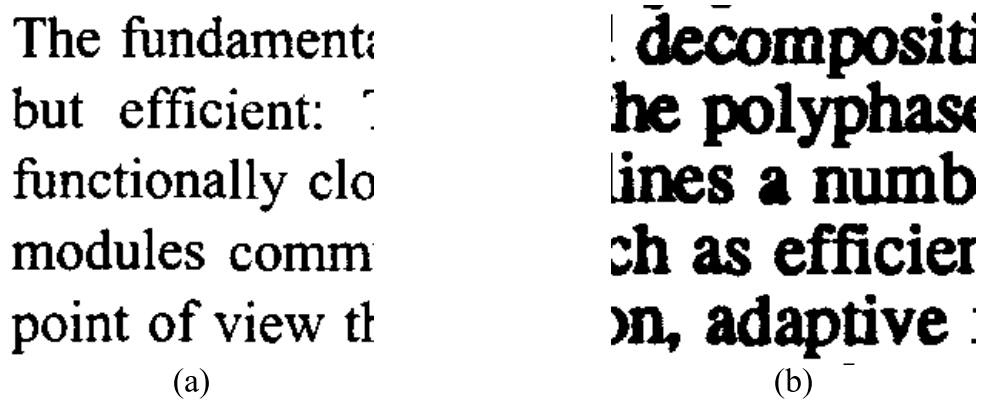

Figura 4.2: Parte de dois documentos pertencentes ao banco de imagens de documentos utilizado: (b) apresenta traços mais grossos que (a).

em alguns caracteres, fato que ocasiona união de caracteres vizinhos e deformação de alguns caracteres. Este problema está ilustrado na Figura 4.2, onde (b) apresenta traços mais grossos, caracteres unidos e deformados. Por outro lado, a Figura 4.2(a) apresenta um trecho de imagem com traços mais finos e algumas vezes não contínuos causando outros tipos de deformação. Este tipo de variação é comum em documentos digitalizados e por isso foi levada em consideração durante o desenvolvimento do método proposto.

4.2 Protocolo Experimental

Os documentos utilizados para criar o banco de dados de testes também contém uma versão PDF no formato texto, fornecidos no mesmo CD das versões digitalizadas. Inicialmente se pretendia utilizar esta versão textual para extrair as palavras e suas respectivas posições para gerar o índice de palavras. Porém, as posições das palavras na versão textual não correspondem, com o devido grau de precisão, às posições das mesmas palavras na versão digitalizada, composta de imagens. Assim, não é possível comparar a posição das palavras encontradas nos testes com a posição das palavras existentes no banco de dados de testes. Devido a isso, não se pode utilizar esta versão textual dos documentos e houve a necessidade de utilizar o software Acrobat (2006) para gerar os documentos na versão textual.

Utilizando-se o OCR embutido no software comercial Acrobat (2006), converteu-se as imagens de documentos digitalizados em documentos texto com o mesmo layout, os quais foram utilizados para fazer a avaliação dos métodos implementados. O software Xpdf (2005) foi utilizado para extrair o texto contido nestes documentos convertidos juntamente com a posição de cada palavra na página, de forma a gerar um índice contendo

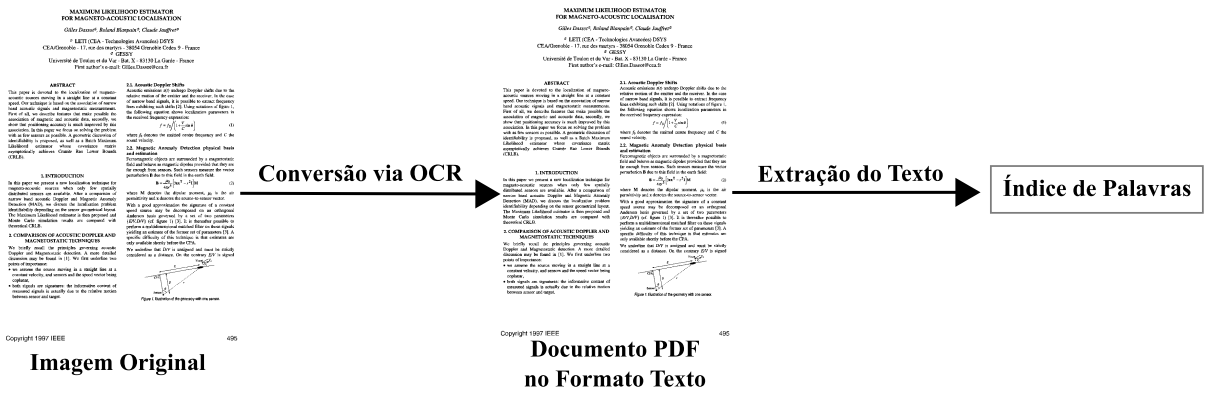

Figura 4.3: Processo de criação do banco de dados de testes.

a palavra no formato ASCII, a posição, o número da página e o nome do documento. Este processo está ilustrado na Figura 4.3. Durante a avaliação dos resultados busca-se no índice a palavra existente na página e posição em que uma determinada palavra foi encontrada na imagem e compara-se as palavras, contabilizando assim o número de palavras corretamente encontradas e o número total de palavras encontradas. Além disso, uma imagem de cada palavra encontrada durante os experimentos foi salva para conferência visual.

O processo de conversão via OCR falhou para aproximadamente 1% das imagens de documentos. Estas imagens não puderam ser convertidas para texto e deste modo foram removidas do banco de imagens. Porém, isto não causa um grande impacto no número de amostras disponíveis.

Para avaliar o método proposto foram utilizadas inicialmente duas medidas: precisão e revocação. A precisão é a relação entre o número de palavras corretamente encontradas e o número total de palavras encontradas durante o teste. A revocação é a relação entre o número de palavras encontradas e o número de ocorrências das palavras procuradas na fração do banco de dados de testes utilizada no teste em questão. As taxas de precisão P e revocação R foram obtidas de acordo com as equações 4.1 e 4.2.

$$P = \frac{X_c}{X_t} \tag{4.1}$$

$$R = \frac{X_c}{X_e} \tag{4.2}$$

onde, X_c corresponde ao número de palavras corretamente encontradas nas imagens de documentos analisadas, X_t corresponde ao número total de palavras encontradas nas imagens de documentos analisadas durante a realização do teste e X_e corresponde ao número de ocorrências, no índice gerado a partir dos arquivos no formato texto, das palavras pro-

curadas. O valor ideal para P e R seria 1, o que representa taxas de precisão e revocação iguais a 100% que por sua vez representa um resultado onde nenhuma palavra foi identificada incorretamente e todos as palavras existentes nos documentos foram encontradas.

Dizer que um conjunto de características ou a utilização de um limiar diferente apresentou um resultado superior a outro não é uma tarefa trivial. Na realidade, o peso atribuído à precisão e revocação pode variar dependendo da utilização do método em questão. Porém, para efeito de comparação entre os resultados fornecidos pelos diferentes conjuntos de características utilizados, deve-se encontrar uma métrica que determine o quanto um método foi melhor ou pior que outro. Nesse contexto, a métrica F_1(YANG; LIU, 1999) foi utilizada para a obtenção de um valor que represente o resultado de um teste. Tal métrica está definida na equação 4.3.

$$F_1 = \frac{2RP}{R+P} \tag{4.3}$$

A eleição das palavras que foram utilizadas nos testes foi feita através da seleção das cinqüenta palavras mais freqüentes encontradas nos documentos presentes no banco de dados. A experiência demonstrou que a avaliação dos métodos se torna mais difícil quando utilizamos palavras pequenas. Devido a isso, inclui-se uma restrição na seleção das palavras mais freqüentes para garantir que apenas palavras com pelo menos cinco caracteres fossem selecionadas. Além disso, algumas palavras comumente encontradas em documentos normalmente chamadas de *stop words*, como conectores, não foram consideradas. Na maioria dos casos estas palavras não são úteis para identificar um documento e raramente são realizadas buscas por elas, visto que não trazem informação alguma. A listagem completa das palavras desconsideradas durante a seleção das palavras mais freqüentes esta disponível no Anexo A. A Tabela 4.1 relaciona o resultado da seleção juntamente com o número de ocorrências de cada palavra no índice gerado a partir dos documentos no formato texto. O número de ocorrências é relativo a todos os documentos existentes no banco de dados de testes.

O banco de imagens de documentos foi divido em duas partes. A primeira parte é formado pelos primeiros 50 documentos deste banco, selecionados em ordem alfabética, e foi utilizada para a definição do limiar de similaridade e do limiar que define a distância máxima positiva entre duas colunas extraídas com o conjunto de características ULTC. A definição destes parâmetros foi realizada através da realização de testes com os 50 documentos selecionados, da observação das palavras obtidas e das suas respectivas pontuações de comparação com as palavras procuradas. A primeira parte do banco de imagens de

Palavra	**Ocorrências**	**Palavra**	**Ocorrências**
speech	5364	shown	2366
signal	5312	models	2315
using	5223	channel	2305
algorithm	4954	different	2269
model	4544	problem	2251
Figure	4512	input	2244
noise	3969	proposed	2242
filter	3723	approach	2185
Copyright	3425	paper	2146
number	3378	signals	2114
function	3285	linear	2066
results	3262	obtained	1895
method	3164	estimation	1805
based	3108	algorithms	1799
performance	2890	information	1795
image	2855	Speech	1726
error	2855	filters	1711
vector	2748	coding	1707
given	2741	output	1704
order	2679	coefficients	1678
parameters	2496	analysis	1670
matrix	2428	sequence	1638
training	2405	Table	1620
recognition	2386	Signal	1616
frequency	2384	values	1614

Tabela 4.1: Palavras, com o respectivo número de ocorrências, utilizadas na realização dos testes.

documentos também foi utilizada para validação do método proposto. A segunda parte é formada de 815 documentos e foi utilizada para a realização dos testes. Assim, cada palavra da Tabela 4.1 e os 815 documentos existentes no banco de dados de testes foram utilizadas na avaliação do método e de cada conjunto de características implementado. Isto gerou um grande volume de dados e permitiu uma boa avaliação do sistema.

O procedimento descrito acima foi realizado para todos os conjuntos de características implementados, assim como para as suas variações. Seguindo este protocolo tem-se a capacidade de avaliar os métodos propostos através da execução de testes com as 815 imagens de documentos destinadas a este propósito e da utilização da métrica F_1 para identificar os melhores resultados. Através da análise dos resultados, buscando identificar os problemas que impossibilitam melhores resultados, é possível identificar as falhas existentes em cada um dos conjuntos de características avaliados.

Método	**Iterações**
Original	21393
Otimizado	643
Ganho	**96,99%**

Tabela 4.2: Número de iterações necessárias para detectar os 40 primeiros espaços vazios em uma página.

4.3 Segmentação

A etapa de segmentação consiste na segmentação de componentes conexos seguida da segmentação de blocos de texto, linhas e palavras. A segmentação de palavras em uma linha é realizada utilizando-se a mediana M das distâncias entre caracteres vizinhos presentes na linha analisada. Dois componentes conexos são considerados pertencentes a mesma palavra se a distância horizontal entre eles for menor que αM. O fator α foi definido através de experimentos, realizados sobre os 50 documentos selecionados para ajustar os parâmetros do método, e da análise visual da segmentação das imagens dos documentos, chegando-se a bons resultados com $\alpha = 2$.

Durante o desenvolvimento dos métodos de segmentação de linhas e palavras foi possível realizar uma melhoria no método proposto por Breuel (2002) para fazer a detecção de espaços em vazios na imagem. A detecção de espaços vazios é importante no contexto de detecção de colunas e linhas. Um espaço vazio pode representar a separação entre duas colunas do texto ou simplesmente a separação entre uma figura ou tabela de um bloco de texto. O método original, proposto por Breuel, visa a detecção dos espaços vazios na imagem ordenados por área. Porém, a maneira como a página é dividida a cada iteração do sistema causa muitos recálculos. Tendo em vista este problema, realizou-se uma modificação do algoritmo para que evitasse por completo o recálculo. Os ganhos em tempo de execução podem ser visualizados através do número de iterações do algoritmo para detectar os primeiros 40 espaços na imagem de uma página. A Tabela 4.3 relaciona o número de iterações do método original e do método otimizado, assim como o ganho obtido com a alteração no método. Porém, é interessante lembrar que o método alterado não garante o retângulo de maior área, apesar de retornar retângulos grandes o suficiente para a correta segmentação da página.

4.4 Experimentos Realizados

Os experimentos foram realizados com os três conjuntos de características propostos e com os dois conjuntos gerados a partir da modificação dos conjuntos LRPS e ULTC. Além disso, para aumentar o desempenho do método testado uma etapa de suavização de contornos foi adicionada ao sistema. A utilização da suavização trouxe benefícios para alguns dos conjuntos de características avaliados, aumentando as taxas de revocação. Porém, em outros conjuntos não houve melhora. Estes experimentos foram realizados sobre os 815 documentos dedicados a este propósito, não pertencentes ao grupo de documentos selecionados para ajuste de parâmetros.

Em cada um dos conjuntos de características avaliou-se diferentes valores para o limiar de similaridade. Este limiar corresponde ao valor mínimo aceitável da pontuação obtida após a comparação de dois descritores. Esta pontuação tem mínimo em 0 e máximo em 1, sendo que 1 somente pode ser obtido quando o descritor da palavra procurada existe integralmente(sem inserções e remoções) dentro do descritor extraído da imagem da palavra analisada. Assim, quando maior for a pontuação menor o número de modificações que precisam ser realizadas no descritor da palavra procurada para que ele se iguale a porção possivelmente correspondente a ele dentro do descritor extraído da imagem da palavra.

Antes de iniciar a análise de cada um dos conjuntos de características implementados vale a pena sumarizar as siglas adotadas para os conjuntos de características descritos nas próximas seções:

- LRPS: Características estruturais baseadas na detecção de linhas retas, análise de transições e do posicionamento relativo às linhas de ascendentes e descendentes;
- AYV: Divisão da coluna em regiões e atribuição de múltiplos de dois a cada região que contêm o pixel central de uma seqüência consecutiva de pixels pretos;
- ULTC: Características de contorno superior e inferior e número de transições;
- ULTC II: Conjunto ULTC com a adição da características de contagem de pixels pretos.

4.4.1 LRPS

As primeiras características implementadas e testadas foram as propostas por Lu e Tan (2004), chamadas de LRPS. Estas características apresentaram um grau de dificuldade relativamente alto na implementação, devido à detecção de linhas retas e à detecção de relacionamentos entre linhas que se interceptam. Apesar dos autores relatarem resultados de precisão e revocação acima de 90%, o máximo que se obteve nos testes foi 61,37% e 53,11% para precisão e revocação, respectivamente. A causa de valores abaixo do esperado pode estar relacionada a dificuldade de se delimitar linhas retas e encontrar relações entre elas em imagens ruidosas. Além disso, não foi possível utilizar o mesmo banco de imagens utilizado pelos autores, visto que na maioria dos testes realizados pelos autores foi utilizado um banco de imagens próprio que não foi possível obtê-lo. Os autores também utilizaram um banco de imagens de documentos comercial chamado *UW*, o qual só está disponível para compra.

Testou-se o conjunto de características LRPS na sua forma original, como proposto pelos autores. Porém, devido ao baixo desempenho obtido decidiu-se realizar uma modificação que visou diminuir os impactos causados por ruídos. Esta modificação consistiu em descartar características isoladas, ou seja, diferente da característica extraída anteriormente e posteriormente a ela.

A Tabela 4.3 ilustra os resultados obtidos com o conjunto de características LRPS e LRPS Modificado, aonde é possível observar que a medida em que o limiar de similaridade aumenta também aumenta a precisão, porém decai a taxa de revocação. Este comportamento é característica do método e, através da métrica F_1, deve-se encontrar um equilíbrio entre as taxas de precisão e revocação de forma a maximizar os resultados. Outro ponto a ser observado é o ganho nas taxas de revocação proporcionado pelo conjunto de características LRPS Modificado, o qual é maior proporcionalmente a queda nas taxas de precisão quando comparado ao conjunto de características LRPS. Assim, o desempenho do conjunto LRPS Modificado foi considerado superior ao LRPS original. Os itens em negrito na Tabela 4.3 representam os melhores resultados, julgados com o auxílio da métrica F_1, com e sem a suavização de contornos.

4.4.2 AYV

O segundo conjunto de características implementado e avaliado foi um subconjunto das características propostas por Arica e Yarman-Vural (2000), referenciado neste do-

Limiar Similaridade		0,60	0,65	0,70	0,75	0,80	0,85
LRPS	**Precisão(%)**	20,27	35,68	55,54	71,31	89,00	95,23
(Original)	**Revocação(%)**	65,46	55,92	45,40	35,40	22,58	12,63
	F_1	0,310	0,436	0,500	0,473	0,360	0,223
LRPS Modificado	**Precisão(%)**	16,55	30,08	48,91	**65,01**	86,63	95,08
	Revocação(%)	74,83	67,31	58,38	**48,03**	34,12	21,05
	F_1	0,271	0,416	0,532	**0,553**	0,490	0,345
LRPS	**Precisão(%)**	— —	— —	— —	69,15	87,63	— —
(com suavização)	**Revocação(%)**	— —	— —	— —	42,98	28,93	— —
	F_1	— —	— —	— —	0,530	0,435	— —
LRPS Modificado	**Precisão(%)**	— —	— —	— —	**61,37**	83,46	— —
(com suavização)	**Revocação(%)**	— —	— —	— —	**53,11**	38,73	— —
	F_1	— —	— —	— —	**0,569**	0,529	— —

Tabela 4.3: Estatísticas de desempenho do conjunto de características LRPS original e modificado sobre os 815 documentos digitalizados.

cumento por AYV. Apenas as características relativas às colunas da imagem foram implementadas, visto que as características propostas pelos autores englobam a análise da imagem da palavra na diagonal e na horizontal. Porém, esta análise não pode ser adaptada ao método porque não permitiria a busca por "sub-palavras".

O conjunto de características AYV é relativamente simples e basea-se por completo na análise das transições. O resultado obtido nos testes foi o menor entre todos os conjuntos de características testados. Porém, não se chegou a testar a combinação desse conjunto de características com outras técnicas, o que poderia dar respostas mais concretas sobre o desempenho das características.

As taxas de precisão do conjunto de características AYV, obtidas após a realização dos testes e apresentadas na Tabela 4.4, são semelhantes as obtidas com o conjunto de características LRPS. Porém, as taxas de revocação estão bem abaixo do obtido com o conjunto de características LRPS. Uma análise mais profunda dos resultados com este conjunto, mostrou que as características extraídas são sensíveis a variações pequenas na forma dos caracteres e a ruídos, devido a isso obteve-se valores inferiores nas taxas de revocação.

Pode-se observar na Tabela 4.4 que o uso da suavização de contornos aumentou a precisão porém provocou uma queda proporcionalmente maior na revocação. Através da utilização da métrica F_1 foi possível verificar que o uso da suavização provocou uma queda de 0,73% no desempenho do conjunto AYV. Ou seja, a suavização de contornos não trouxe benefícios ao sistema quando utilizada juntamente com este conjunto de características.

Limiar Similaridade		0,60	0,65	0,70	0,75	0,80	0,85
AYV	**Precisão(%)**	10,38	19,18	**38,92**	61,86	85,87	94,62
	Revocação(%)	39,63	31,77	**23,12**	15,11	6,85	2,55
	F_1	0,165	0,239	**0,290**	0,243	0,127	0,050
AYV	**Precisão(%)**	— —	20,06	**40,03**	62,93	— —	— —
(com suavização)	**Revocação(%)**	— —	30,74	**21,84**	14,09	— —	— —
	F_1	— —	0,243	**0,283**	0,230	— —	— —

Tabela 4.4: Estatísticas de desempenho do conjunto de características AYV sobre os 815 documentos digitalizados.

4.4.3 ULTC

As características de contorno superior e inferior e número de transições preto/branco formam o terceiro conjunto de características avaliado. Estas características são relativamente simples e apresentam um boa resistência a ruídos. No entanto, tais características não distinguem suficientemente alguns caracteres, impondo um limite relativamente baixo a precisão ou revocação do método, dependendo do valor do limiar de similaridade.

Quando este conjunto de características foi utilizado realizou-se testes com uma variação do mesmo que utilizou uma tabela de vetores de características em que todos os outros vetores de características extraídos da imagem deveriam se adequar. Ou seja, um vetor calculado a partir de uma coluna da imagem deveria ser substituído pelo vetor na tabela que apresentasse menor distância para ele. Tentou-se com isso diminuir os ruídos e aumentar a revocação. Porém os resultados mostraram que nem a precisão nem a revocação obtiveram ganho com essa técnica.

Este conjunto de características utilizou duas tabelas de conversão de caracteres para características ao mesmo tempo. Uma tabela foi gerada a partir da imagem de caracteres em uma fonte com serifa e a outra com uma fonte sem serifa. A utilização de duas tabelas de conversão abriu a possibilidade de tratar de igual maneira caracteres impressos em fontes com e sem serifa sem a necessidade de uma etapa de remoção de serifas. Isto proporcionou resultados melhores que os do conjunto de características LRPS, no qual se efetua a remoção das serifas em uma etapa pós-extração de características.

A comparação de dois vetores de características, descritores de duas colunas, extraídos utilizando este conjunto de características precisa de um valor que indique qual a distância máxima positiva entre eles. Experimentalmente chegou-se ao valor de 0,07. A definição deste limiar afeta o método de maneira semelhante ao limiar de similaridade. Ou seja, quando menor o valor do limiar maior a precisão do método proposto.

Limiar Similaridade		0,60	0,65	0,70	0,75	0,80	0,85
ULTC	**Precisão(%)**	27,16	46,61	**65,89**	81,42	91,98	96,78
	Revocação(%)	86,32	80,48	**70,35**	55,63	36,95	19,01
	F_1	0,413	0,590	**0,680**	0,661	0.537	0,318
ULTC (com suavização)	**Precisão(%)**	— —	— —	**64,84**	80,28	90,98	— —
	Revocação(%)	— —	— —	**71,63**	56,79	38,17	— —
	F_1	— —	— —	**0,681**	0,665	0,538	— —

Tabela 4.5: Estatísticas de desempenho do conjunto de características ULTC sobre os 815 documentos digitalizados.

A Tabela 4.5 relaciona as taxas de precisão e revocação para os diferentes limiares de similaridade avaliados. Os itens em negrito destacam os melhores resultados de acordo com a equação 4.3.

Após o uso da suavização de contornos o sistema apresentou uma melhora de apenas 0,02% na métrica F_1. Isto mostra que as características do conjunto ULTC são menos sensíveis a ruídos do que as do conjunto LRPS onde os ganhos com o uso da suavização foram de 1,69% na métrica F_1. Ou seja, a suavização quase não trouxe benefícios ao conjunto ULTC. Quando se leva em conta que houve um aumento de 15% no tempo de processamento em virtude da suavização, pode-se considerar desnecessário o seu uso em associação com o este conjunto de características.

4.4.4 ULTC Modificado

Numa tentativa de melhorar a precisão do sistema uma quarta característica foi adicionada ao conjunto de características ULTC, a qual mediu a porcentagem de pixels pretos em uma coluna.

Conforme ilustrada na Tabela 4.6, o resultado dos testes mostrou um ganho na precisão. Porém, perdeu-se proporcionalmente em revocação, quando comparado ao conjunto ULTC. Além disso, obteve-se aumento de 16% no tempo de processamento. O uso da suavização de contornos não trouxe benefícios ao sistema quando utilizada juntamente com que este conjunto de características.

A inclusão de mais uma característica ao conjunto ULTC trouxe, como era esperado, um aumento da precisão e ao mesmo tempo proporcionou uma queda proporcionalmente maior na revocação. Este fato torna o conjunto ULTC Modificado inferior ao ULTC, visto que o desempenho do primeiro pode ser obtido no segundo quando se aumenta o valor do limiar de similaridade. Sendo assim, pode-se concluir que a implementação da

Limiar Similaridade		0,65	0,70	0,75	0,80	0,85
ULTC Modificado	**Precisão(%)**	**61,99**	77,42	88,11	94,88	98,30
	Revocação(%)	**68,53**	52,72	34,25	18,03	7,61
	F_1	**0,651**	0,627	0,493	0,303	0,141
ULTC Modificado	**Precisão(%)**	**60,96**	76,36	86,97	94,28	— —
(com suavização)	**Revocação(%)**	**68,40**	52,07	33,80	18,23	— —
	F_1	**0,645**	0,619	0,487	0,306	— —

Tabela 4.6: Estatísticas de desempenho do conjunto de características ULTC Modificado sobre os 815 documentos digitalizados.

Características	**Precisão**	**Revocação**	F_1	**Limiar Similaridade**
AYV	38,92	23,12	0,290	0,70
LRPS	61,37	53,11	0,569	0,75
ULTC Modificado	61,99	68,53	0,645	0,65
ULTC	**65,89**	**70,35**	**0,681**	0,70

Tabela 4.7: Estatísticas de desempenho com os melhores resultados de cada conjunto de características proposto.

característica extra não trouxe benefícios ao sistema.

Na Tabela 4.7 estão relacionadas as características implementadas e o melhor desempenho obtido com cada um dos conjuntos de características. Os valores do conjunto LRPS são relativos ao conjunto LRPS modificado (LRPS Modificado) e com a suavização de contornos.

Os melhores resultados obtidos, de acordo com a métrica F_1, na busca pelas cinqüenta palavras mais freqüentes nos oitocentos e quinze artigos integrantes da base de testes foi obtido com o uso de características de contorno superior e inferior e número de transições preto/branco, ou seja, o conjunto ULTC. Estas características são relativamente simples e apresentam um boa resistência a ruídos. No entanto, tais características não distinguem suficientemente alguns caracteres, impondo um limite relativamente baixo a precisão ou revocação do método, dependendo do valor do limiar de similaridade.

4.5 Análise de Erros

Cada um dos conjuntos de características implementados possui suas falhas. O conjunto de características AYV tem como maior falha a sensibilidade a ruídos. O conjunto LRPS apresenta um alto nível de complexidade e a detecção de linhas retas nem sempre pode ser realizada da forma como os autores propuseram. O conjunto ULTC, por sua vez,

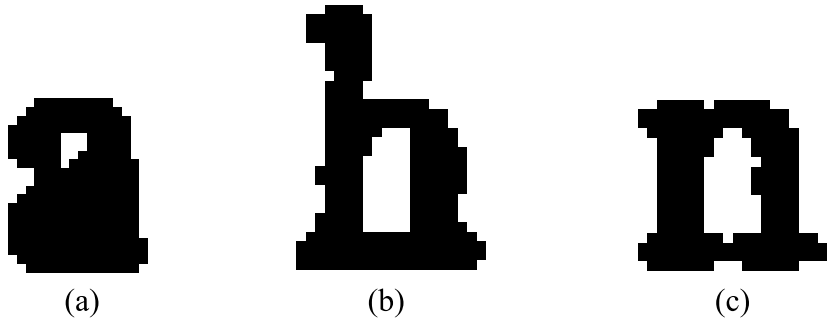

Figura 4.4: Deformações nos caracteres “a”(a), “h”(b) e “n”(c) causadas por ruídos.

não distingue suficientemente alguns caracteres como o “u” e o “n”.

Além das fraquezas encontradas em cada conjunto de características ainda existem os problemas causados por ruídos. Na Figura 4.4 estão ilustrados três caracteres que tiveram suas formas alteradas por más condições de impressão e/ou digitalização. Na Figura 4.4(a) o caractere “a” tem seu interior preenchido eliminando o *loop* próprio deste caractere, esse tipo de deformação afeta características baseadas em transições mas não afeta características relativas ao contorno externo. Nas Figuras 4.4(b) e 4.4(c) o caractere “h” e o caractere “n” tem a parte inferior ligada com um traço eliminando a concavidade própria desses caracteres, esse tipo de deformação afeta todas as características propostas neste trabalho. Estas deformações são um desafio a ser superado, tratá-las na etapa de pré-processamento seria um procedimento bastante agressivo podendo causar imperfeições a caracteres sem deformações. Provavelmente, o lugar mais adequado para tratá-las seria na extração de características, porém distinguir uma característica extraída de um caractere normal de um defeituoso não é um tarefa fácil e possivelmente causaria a inserção de heurísticas no método.

Durante a análise dos resultados ficou claro que o conjunto de características LRPS apresentou um desempenho superior no processamento de palavras escritas na grafia itálica. Esta superioridade pode ser atribuída a utilização das características estruturais relativas a extração de linhas retas, as quais apresentam robustez com relação a presença ou não de fontes itálicas. A diferença causada pelo uso da grafia itálica é mais visível em caracteres com ascendentes e descendentes e em caracteres que apresentam linhas retas. Conjuntos de característica dependentes da análise da transição não se comportam bem na presença de itálicos, e os conjuntos ULTC e AYV praticamente não identificaram palavras na grafia itálica. Uma das soluções para este problema seria a correção de inclinação da imagem da palavra após a segmentação da mesma. Outra solução seria utilizar mais uma tabela de conversão de caractere para descritor, nesse caso a tabela seria gerada a

partir de uma imagem com modelos de caractere criados com uma fonte itálica.

Após avaliar diferentes conjuntos de características foi possível diagnosticar que as características se dividem em dois grupos. Em um grupo tem-se as características que apresentam boa invariância a ruídos e toleram pequenas variações na forma dos caracteres. O segundo grupo é formado pelas características que são mais sensíveis a ruídos, porém são mais precisas que as do primeiro grupo. Essa divisão é visível entre os conjuntos de características avaliados. Por exemplo, o conjunto de características AYV possui basicamente uma característica, a qual é sensível a imperfeições na forma dos caracteres e produz baixas taxas de revocação. Do outro lado, tem-se as características de contorno utilizadas no conjunto de características ULTC, as quais tem baixa precisão porém apresentam boas taxas de revocação.

Na Figura 4.5 estão ilustrados alguns dos problemas comumente encontrados no processo de detecção de linhas retas do conjunto de características LRPS. Estes problemas afetam a desempenho do método, de modo que ocasionam a incorreta identificação das características na imagem. Na Figura 4.5(a) tem-se o problema causado pela escolha da linha de varredura. A linha de varredura é utilizada como ponto de partida para detectar os traços existentes na imagem e quando esta linha coincide com uma reta horizontal não é possível detectar as linhas verticais que possivelmente cortem esta linha horizontal. No exemplo da figura, deveriam ser detectadas as duas linhas retas na diagonal presentes na imagem do caractere “A”, mas como o ponto médio fica no centro da linha horizontal presente neste caractere nenhuma delas pode ser encontrada. Na Figura 4.5(b) e (c) estão ilustrados casos onde uma linha reta pode ser encontrada em posições não desejadas. Na imagem do caractere “S” foi detectada uma linha reta na diagonal, mas o conjunto de características não prevê uma reta neste caractere. Na imagem do caractere “R” foram encontradas duas retas, mas somente a da esquerda está correta.

A inclusão de mais uma característica ao conjunto ULTC, conforme mostrado no conjunto ULTC Modificado, mostrou que a precisão pode ser melhorada, mas com o custo da queda na revocação. O equilíbrio entre essas duas taxas é um questão fundamental, visto que uma tende a crescer em detrimento da outra. Isto é natural, levando-se em conta que quanto mais preciso é um conjunto de características mais sensível a ruídos também é.

A utilização da suavização de contornos, além de corrigir imperfeições na imagem, pode gerar imperfeições em alguns casos. Este comportamento pode ser observado nos resultados do conjunto de características AYV, onde o desempenho do sistema apresentou

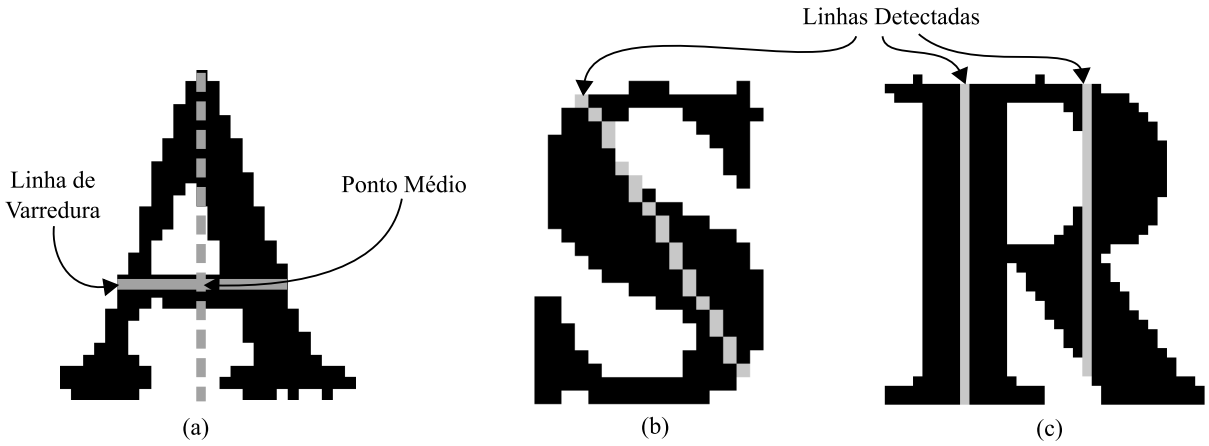

Figura 4.5: Problemas na detecção de linhas retas no algoritmo utilizado no conjunto de características LRPS. (a) linha de varredura coincide com um traço horizontal no caractere “A”. (b) linha incorretamente detectada no caractere “S”. (c) a linha da direita é incorretamente detectada no caractere “R”.

piora após a inclusão da suavização de contornos. Porém, vale destacar, que no conjunto de características LRPS a suavização trouxe visível melhora ao sistema. Assim, o uso da suavização de contornos deve ser o menos agressiva possível para que não cause mais imperfeições do que as existentes.

Capítulo 5

Conclusão

Neste trabalho foi apresentado um método capaz de realizar a busca por palavras em imagens de documentos impressos sem recorrer ao OCR, mantendo o foco na utilização de novos conjuntos características compostos por características presentes nos trabalhos revisados. Foram apresentados três conjuntos de características utilizados para gerar descritores representantes de imagens de palavras. Também foi apresentada a forma de comparar estes descritores seguindo a técnica de comparação inexata de características, de maneira que houve a necessidade de adaptar o procedimento de comparação a cada conjunto avaliado. Entretanto, para realizar a extração das características de imagens de palavras é necessário segmentar as palavras existentes em uma imagem de página. Assim, também foi apresentado o procedimento de segmentação utilizado pelo método e descritas a etapas necessárias para se chegar a segmentação das palavras presentes na imagem de uma página.

Entre as contribuições deste projeto pode-se destacar: as otimizações no processo de segmentação, a criação de um banco de dados de testes e a avaliação de novos conjuntos de características. A contribuição no processo de segmentação está relacionada a economia de tempo de processamento proporcionada pela otimização do método de detecção de espaços em branco na imagem. A criação de um banco de dados de testes representa um recurso que permite que trabalhos futuros realizem testes em um banco de dados com 865 imagens de documentos e índice de palavras com informação de posicionamento das palavras contidas nos documentos. A avaliação de novos conjuntos de característica é importante para a ciência da computação, fornecendo resultados e análises que contribuirão com trabalhos realizados na mesma área de pesquisa desse trabalho e em áreas relacionadas.

A avaliação do método e dos conjuntos de características propostos envolveu a rea-

lização de vários testes com diferentes valores para o limiar de similaridade. Depois de realizados os testes, foi possível observar que quanto maior o limiar de similaridade(valor mínimo necessário para considerar dois descritores como sendo relativos a mesma palavra) menor a revocação e maior a precisão do método. Também, ficou visível algumas fraquezas dos conjuntos de características apresentados, como a baixa robustez perante a ruídos.

Os resultados apresentados mostram a viabilidade do método e deixam expostos alguns pontos que podem ser melhorados no futuro. Uma das necessidades a serem supridas no futuro é possibilidade de buscar frases e não apenas palavras em imagens de documentos. Outro ponto a considerar é a combinação de características para gerar novos conjuntos. Além disso, pode-se combinar o método com OCR. O uso de OCR é computacionalmente intensivo quando se deseja analisar uma página inteira. Porém, seria possível em um trabalho futuro realizar uma filtragem das palavras que mais se assemelham com a palavra procurada e depois submeter o resultado da filtragem a conversão para texto por OCR, aumentando o nível de precisão do método e ao mesmo tempo reduzir o tempo de processamento dos métodos baseados em OCR.

Referências Bibliográficas

ACROBAT. Adobe, 2006. Disponível em: <http://www.adobe.com>.

ARICA, N.; YARMAN-VURAL, F. T. One-dimensional representation of two-dimensional information for HMM based handwriting recognition. *Pattern Recognition Letters*, v. 21, n. 6-7, p. 583–592, jun. 2000.

BALASUBRAMANIAN, A.; MESHESHA, M.; JAWAHAR, C. V. Retrieval from document image collections. In: *Workshop on Document Analysis Systems*. Nelson, Nova Zelândia: Springer, 2006. p. 1–12.

BREUEL, T. M. Two geometric algorithms for layout analysis. In: *DAS '02: Proceedings of the 5th International Workshop on Document Analysis Systems V*. London, UK: Springer-Verlag, 2002. p. 188–199. ISBN 3-540-44068-2.

CHEN, F. R.; BLOOMBERG, D. S.; WILCOX, L. D. Detection and location of multicharacter sequences in lines of imaged text. *Journal of Electronic Imaging*, v. 5, n. 1, p. 37–49, jan. 1996.

IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP"97).

KISE, K.; WUOTANG, Y.; MATSUMOTO, K. Document image retrieval based on 2d density distributions of terms with pseudo relevance feedback. In: *ICDAR '03: Proceedings of the Seventh International Conference on Document Analysis and Recognition*. Washington, DC, USA: IEEE Computer Society, 2003. p. 488. ISBN 0-7695-1960-1.

LEYDIER, Y. *Word Spotting*. 2004. Disponível em: <http://liris.cnrs.fr/yann.leydier/ws.html>. Acesso em: 13 de fev. de 2006.

LOPRESTI, D. P.; ZHOU, J. Retrieval strategies for noisy text. In: *Proceedings of the Fifth Annual Symposium on Document Analysis and Information Retrieval*. Las Vegas, NV: Information Science Research Institute, University of Nevada, Las Vegas (Ed.), 1996. p. 225–269.

LU, Y.; TAN, C. L. Word spotting in chinese document images without layout analysis. In: *16th International Conference on Pattern Recognition (ICPR)*. Quebec, Canada: IEEE Computer Society, 2002. p. 57–60.

LU, Y.; TAN, C. L. Information retrieval in document image databases. *IEEE Trans. Knowl. Data Eng.*, v. 16, n. 11, p. 1398–1410, 2004.

MARINAI, S. et al. A general system for the retrieval of document images from digital libraries. In: *DIAL*. California, EUA: IEEE Computer Society, 2004. p. 150–173. ISBN 0-7695-2088-X.

OTSU, N. A threshold selection method from gray-level histogram. *IEEE Trans. Systems, Man, and Cybernetics*, v. 8, p. 62–66, 1978.

RATH, T. M.; MANMATHA, R. Features for word spotting in historical manuscripts. In: *ICDAR*. [S.l.]: IEEE Computer Society, 2003. p. 218–222. ISBN 0-7695-1960-1.

SUEN, C. Y. et al. Computer recognition of unconstrained handwritten numerals. In: *Proc. IEEE*. [S.l.]: IEEE Computer Society, 1992. v. 7, n. 80, p. 1162–1180.

TAN, C. L. et al. Imaged document text retrieval without OCR. *IEEE Trans. Pattern Anal. Mach. Intell*, v. 24, n. 6, p. 838–844, 2002. Disponível em: <http://computer.org/tpami/tp2002/i0838abs.htm>.

WANG, Z.; LU, Y.; TAN, C. L. Word extraction using area voronoi diagram. In: *2003 Conference on Computer Vision and Pattern Recognition Workshop*. Wisconsin, EUA: cvprw, 2003. v. 3, p. 31.

XPDF: A pdf viewer for X. Glyph & Cog, 2005. Disponível em: <http://www.foolabs.com/xpdf/>.

YANG, Y.; LIU, X. A re-examination of text categorization methods. In: *Proc. 22th Ann. Int'l ACM SIGIR Conf. Research and Development in Information Retrieval*. Califórnia, Canadá: ACM, 1999. p. 42–49.

ANEXO A

Palavras Desconsideradas Durante a Seleção das Palavras Utilizadas nos Testes

A, about, above, across, after, afterwards, again, against, all, almost, alone, along, already, also, although, always, am, among, amongst, amoungst, amount, an, and, another, any, anyhow, anyone, anything, anyway, anywhere, are, around, as, at, back, be, became, because, become, becomes, becoming, been, before, beforehand, behind, being, below, beside, besides, between, beyond, bill, both, bottom, but, by, call, can, cannot, cant, co, computer, con, could, couldnt, cry, de, describe, detail, do, done, down, due, during, each, eg, eight, either, eleven, else, elsewhere, empty, enough, etc, even, ever, every, everyone, everything, everywhere, except, few, fifteen, fify, fill, find, fire, first, five, for, former, formerly, forty, found, four, from, front, full, further, get, give, go, had, has, hasnt, have, he, hence, her, here, hereafter, hereby, herein, hereupon, hers, herself, him, himself, his, how, however, hundred, i, ie, if, in, inc, indeed, interest, into, is, it, its, itself, keep, last, latter, latterly, least, less, ltd, made, many, may, me, meanwhile, might, mill, mine, more, moreover, most, mostly, move, much, must, my, myself, name, namely, neither, never, nevertheless, next, nine, no, nobody, none, noone, nor, not, nothing, now, nowhere, of, off, often, on, once, one, only, onto, or, other, others, otherwise, our, ours, ourselves, out, over, own, part, per, perhaps, please, put, rather, re, same, see, seem, seemed, seeming, seems, serious, several, she, should, show, side, since, sincere, six, sixty, so, some, somehow, someone, something, sometime, sometimes, somewhere, still, such, system, take, ten, than, that, the, their, them, themselves, then, thence, there, thereafter, thereby, therefore, therein, thereupon, these, they, thick, thin, third, this, those, though, three, through, throughout, thru, thus, to, together, too, top, toward, towards, twelve, twenty, two, un, under, until, up, upon, us, very, via, was, we, well, were, what, whatever, when, whence, whenever, where, whereafter, whereas, whereby, wherein, whereupon, wherever,

whether, which, while, whither, who, whoever, whole, whom, whose, why, will, with, within, without, would, yet, you, your, yours, yourself, yourselves.