CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DO PARANÁ
Curso de Pós Graduação em Engenharia Elétrica e Informática Industrial

---

DISSERTAÇÃO
apresentada ao CEFET-PR
para obtenção do título de

MESTRE EM CIÊNCIAS

por

LUIZ EDUARDO SOARES DE OLIVEIRA

ESTUDO SOBRE A EXTRAÇÃO DAS ESTRUTURAS LÓGICA E FÍSICA A PARTIR DE IMAGENS DE CHEQUES BANCÁRIOS BRASILEIROS

Banca Examinadora

Presidente e Orientador:

Prof. Dr. Flávio Bortolozzi	PUC-PR

Examinadores:

Prof. Dr. Robert Sabourin (co-orientador)	ÉTS-Canadá
Prof. Dr. Jacques Facon	PUC-PR
Prof. Dr. Celso Kaestner	CEFET-PR

Curitiba, 17 de março de 1.998.

LUIZ EDUARDO SOARES DE OLIVEIRA

ESTUDO SOBRE A EXTRAÇÃO DAS ESTRUTURAS LÓGICA E FÍSICA A PARTIR DE IMAGENS DE CHEQUES BANCÁRIOS BRASILEIROS

Dissertação apresentada ao Curso de Pós-Graduação em Engenharia Elétrica e Informática Industrial do Centro Federal de Educação Tecnológica do Paraná como requisito parcial para obtenção do título de "Mestre em Ciências" – Área de Concentração: Informática Industrial.

Orientador: Prof. Dr. Flávio Bortolozzi.

Curitiba
1.998

Dedico esse trabalho a
minha mãe, Mercêdes,
que de algum lugar
tem me apoiado e dado
força nos momentos
mais difíceis.

Agradecimentos

À Deus, que me permitiu tudo isso.

A minha esposa Marisa, pelo amor, dedicação, companheirismo e compreensão.

Ao professor Flávio Bortolozzi, pela orientação, apoio e compreensão.

Ao professor Jacques Facon, pela amizade e o apoio em Morfologia Matemática.

Ao professor Robert Sabourin, pela oportunidade e orientação.

Aos colegas do grupo, Cinthia, Alceu, José Eduardo e Justino, pelo aprendizado.

A todas pessoas que de alguma forma me apoiaram nessa fase da minha vida.

Sumário

Lista de Figuras..... IX
Lista de Tabelas..... XII
Lista de Abreviaturas..... XIII
Resumo..... XIV
Abstract..... XV

Capítulo I – Introdução
1.1 Desafio..... 16
1.2 Motivação..... 17
1.3 Proposta..... 18
1.4 Contribuição..... 19
1.5 Organização..... 21

Capítulo II – Estado da Arte
2.1 Introdução..... 23
2.2 Análise de Imagem de Documento 23
2.2.1 Técnicas *Top-Down* 23
2.2.2 Técnicas *Bottom-Up* 26
2.3 Análise de Imagem de Documento Baseado em Conhecimento..... 27
2.4 Análise de *Layout* de Documento..... 28
2.5 Processamento Digital de Imagens – PDI 30
2.5.1 Imagem Digital..... 31
2.5.2 Histograma..... 31
2.5.3 Hiperbolização Quadrática do Histograma..... 33
2.5.4 Limiarização (*Thresholding*) 34
2.5.4.1 Método de Otsu 35
2.5.5 Morfologia Matemática Binária..... 36
2.5.5.1 Erosão..... 37
2.5.5.2 Dilatação..... 38
2.5.5.3 Abertura..... 39

2.5.5.4 Fechamento..... 40
2.5.5.5 Condicionalidade e Reconstrução Binária..... 41
2.5.5.6 Reconstrução Híbrida Rápida..... 44
2.6 Padrões de Documentos 44
2.6.1 SGML - *Standard Generalised Markup Language* 46
2.6.2 HTML - *HyperText Markup Language*..... 46
2.6.3 ODA – *Open/Office Document Architecture* 47
2.6.4 TeX 47
2.6.5 RTF – *Rich Text Format* 48
2.6.6 PDF – *Portable Document Format*..... 49
2.7 Ferramentas 49
2.7.1 Conversores de Formato 49
2.7.2 Compiladores 50
2.7.3 *Viewers* 50
2.8 Neste Capítulo 50

Capítulo III – Estrutura do Cheque Bancário Brasileiro

3.1 Introdução 52
3.2 Dimensões do Cheque 53
3.3 Elementos Componentes do Cheque Bancário Brasileiro 54
3.4 Impressão e Confecção do Formulário dos Cheques 60
3.5 O Formulário Cheque 61
3.5.1 Fontes 61
3.5.2 Padrão de Fundo 61
3.5.3 Linhas 62
3.6 Neste Capítulo 63

Capítulo IV – Padrões

4.1 Introdução 64
4.2 SGML (*Standart Generalized Markup Language*) 64
4.2.1 Características do SGML 65
4.2.1.1 *Markup* Descritivo 65
4.2.1.2 Tipos de Documento 65

4.2.1.3 Independência de Dados 66
4.2.2 Estrutura SGML 66
4.2.2.1 Elementos 67
4.2.2.2 Um Exemplo 67
4.2.3 Definindo Estruturas de Documento: O *DTD* 70
4.2.3.1 Um Exemplo de *DTD* 70
4.2.3.2 Regras de Minimização 71
4.2.3.3 Conteúdo do Modelo 71
4.2.3.4 Indicadores de Ocorrência 72
4.2.3.5 Grupo de Conectores 72
4.2.3.6 Modelos de Grupos 73
4.2.4 Atributos 74
4.2.5 Entidades SGML 79
4.3 ODA (*Open/Office Document Architecture)* 80
4.3.1 Estrutura Genérica 81
4.3.2 Atributos das Classe de Objetos 84
4.3.3 Arquitetura do Conteúdo 87
4.3.4 Formato de Troca 88
4.3.4.1 FOD011 ODA - *Simple document structure* 90
4.3.4.2 FOD026 ODA - *Enhanced document structure* 90
4.3.4.3 FOD036 ODA - *Extended document structure* 90
4.4 Comparação entre os modelos ODA e SGML 91
4.5 Neste Capítulo 93

Capítulo V – Implementação

5.1 Introdução 95
5.2 Aquisição da Imagem 95
5.3 Segmentação 96
5.3.1 Limiarização da imagem 97
5.3.2 Exclusão dos padrões indesejados 97
5.3.2.1 Análise Global 98
5.3.2.2 Análise Local 103
5.3.3 Detecção dos separadores no cabeçalho do cheque 106

5.3.4 Marcação dos blocos 108
5.4 Reconhecimento 112
5.5 Codificação SGML..... 113
5.5.1 O DTD para o cheque 114
5.5.2 Elementos do Cheque Bancário Escritos em SGML 115
5.6 Codificação ODA 117
5.7 Neste Capítulo..... 120

Capítulo VI – Análise e Discussão dos Resultados 122
Capítulo VII – Conclusão 128

Referências 130

Anexo A – Resultados do processamento através da análise local 135

Lista de Figuras

Figura 1.1 Visualização do trabalho proposto 19
Figura 1.2 Estrutura global do Projeto ABCPR Group 20
Figura 2.1 (a) Imagem Original, (b) Fechamento morfológico vertical, (c) Fechamento morfológico horizontal e (d) Imagem resultante com os blocos marcados 24
Figura 2.2 Documentos e seus *layout´s* 29
Figura 2.3 Níveis do Processamento Digital de Imagens 30
Figura 2.4 Cheque e o seu respectivo histograma 32
Figura 2.5 Histograma de uma imagem escura..... 32
Figura 2.6 Cheque Bancário antes da hiperbolização quadrática do histograma 33
Figura 2.7 Cheque Bancário após a hiperbolização quadrática do histograma..... 33
Figura 2.8 Exemplos de malhas de elementos estruturantes. (a) quadrado, (b) linha vertical, (c) linha horizontal, (d) cruz e (e) rhombus..... 37
Figura 2.9 Imagem binária original 38
Figura 2.10 Imagem erodida com um elemento estruturante quadrado com três iterações 38
Figura 2.11 Imagem da figura 2.9 dilatada por um elemento estruturante quadrado com cinco iterações 39
Figura 2.12 Imagem original com ruído 40
Figura 2.13 Imagem da figura 2.12 aberta com elemento estruturante cruz 40
Figura 2.14 Imagem original antes do fechamento binário 40
Figura 2.15 Imagem da figura 2.14 após o processo de fechamento com elemento estruturante cruz 41
Figura 2.16 Imagem do texto original 42
Figura 2.17 Imagem dos marcadores 42
Figura 2.18 Dilatação normal com cinco iterações 42
Figura 2.19 Dilatação condicional com cinco iterações 42
Figura 2.20 Estágios da reconstrução binária 43

Figura 3.1 Modelo para a distribuição de informações em cheques bancários Brasileiros 52
Figura 3.2 Modelo padrão de preenchimento dos campos e áreas de um cheque 53
Figura 3.3 Dimensões mínimas e máximas para um cheque bancário 54
Figura 3.4 Um cheque bancário dividido em quatro faixas 54
Figura 3.5 Descrição dos elementos de uma linha MICR 60
Figura 3.6 Diferentes tipos de fontes usados nos formulários de cheques 61
Figura 4.1 Composição de um documento dentro do modelo ODA 80
Figura 4.2 Objetos genéricos do modelo ODA 83
Figura 4.3 Inter-relação entre as estruturas lógica e física especificadas no modelo ODA 84
Figura 4.4 Exemplo de uma estrutura lógica genérica no modelo ODA 87
Figura 5.1 Etapas do processamento do cheque 95
Figura 5.2 Etapas na segmentação do cheque bancário brasileiro 96
Figura 5.3 Imagem original do cheque 100
Figura 5.4 Figura 5.4: Resultado da equalização por hiperbolização quadrática do histograma..... 100
Figura 5.5 Figura 5.5: Resultado do processo de limiarização pelo método de Otsu..... 100
Figura 5.6 Imagem do cheque sem as linhas verticais..... 101
Figura 5.7 Imagem do cheque sem as linhas verticais após a reconstrução..... 101
Figura 5.8 Imagem do cheque sem as linhas horizontais..... 101
Figura 5.9 Imagem do cheque sem as linhas horizontais após a reconstrução 102
Figura 5.10 Imagem do cheque após o processo de erosão binária..... 102
Figura 5.11 Imagem do cheque após a reconstrução da Figura 5.10..... 102
Figura 5.12 Delimitadores do cheque na análise local..... 104
Figura 5.13 Imagem dos campos da segunda faixa..... 104
Figura 5.14 Imagem dos campos da terceira faixa..... 104
Figura 5.15 Resultado final na análise local (equivalente a Imagem 9 da análise global)..... 105
Figura 5.16 Resultado final na análise global..... 105

Figura 5.17 Cheque com delimitadores horizontais..... 106
Figura 5.18 Delimitadores verticais obtidos a partir de delimitadores horizontais..... 107
Figura 5.19 Dilatação binária com elemento estruturante linha horizontal..... 109
Figura 5.20 Fechamento binário da imagem da Figura 5.19..... 109
Figura 5.21 Dilatação binária com elemento estruturante linha vertical..... 109
Figura 5.22 Fechamento binário da imagem da Figura 5.21..... 110
Figura 5.23 Operação lógica AND entre as imagens das Figuras 5.22 e 5.20.... 110
Figura 5.24 Dilatação com elemento estruturante quadrado na imagem da Figura 5.23..... 110
Figura 5.25 Dilatação com elemento estruturante linha vertical..... 111
Figura 5.26 Blocos marcados para a fase de reconhecimento..... 111
Figura 5.27 Modelo para a distribuição de informações em cheques bancários brasileiros..... 118
Figura 5.28 Distribuição física dos campos no formulário do cheque..... 119
Figura 5.29 Inter-relacionamento entre as estruturas lógica e física..... 120
Figura 6.1 Imagem limiarizada após 1 iteração da equalização 125
Figura 6.2 Imagem limiarizada após 2 iterações da equalização 125

Lista de Tabelas

Tabela 3.1 Composição da Faixa número 1 do cheque bancário..... 55
Tabela 3.2 Códigos de Tipificação 59
Tabela 4.1 Palavras chaves para especificar conteúdo do atributo 76
Tabela 4.2 Palavras chaves para interpretar a ausência do atributo 77
Tabela 4.3 Palavras chaves para especificar conteúdo do atributo 85
Tabela 4.4 Exemplo do modelo físico genérico dentro do modelo ODA 86
Tabela 5.1 Arquivos necessários para compilar um código SGML 117
Tabela 5.2 Modelo lógico para ao cheque bancário 117
Tabela 5.3 Modelo físico para o cheque bancário 118
Tabela 6.1 Comparativo entre as análises global e local 127

Lista de Abreviaturas

SGML	Standard Generalized Markup Language
ODA	Open/Office Document Architecture
ABCPR Group	Automatic Bank Check Processing Research Group
PDI	Processamento Digital de Imagens
RLSA	Run-Length Smoothing Algorithm
HTML	Hyper Text Markup Language
WWW	World Wibe Web
RTF	Rich Text Format
ACM	Association for Computing Machinery
DTD	Document Type Definition
DMU	DeMontfort University
UCL	University College London
JEDI	Joint Eletronic Document Interchange
ISO	International Standards Organization
DSSSL	Document Style Semantics and Specification Language
SPDL	Standard Page Description Language
PDF	Portable Document Format
ECMA	European Computer Manufactures Association
GI	Generic Identifier
CGM	Computer Graphics Metafile
VDC	Virtual Device Coordinates
DAP	Document Application Profiles
ODIF	Open Document Interchange Format
CEFET-PR	Centro Federal de Educação Tecnológica do Paraná
CNPQ	Conselho Nacional de Desenvolvimento Científico e Tecnológico
dpi	*dots per inch*
ÉTS	École de Technologie Supérieure
PUC-PR	Pontifícia Universidade Católica do Paraná

Resumo

O trabalho apresenta um estudo sobre a extração das estruturas lógica e física a partir de imagens de cheques bancários brasileiros. Foi feito um levantamento dos padrões e normas existentes capazes de armazenar as estruturas física e lógica do cheque bancário brasileiro. Elaboramos um estudo teórico entre os dois principais padrões: SGML (*Standard Generalized Markup Language*) e o ODA (*Open/Office Document Architecture).*

Analisamos diferentes tipos de cheques bancários brasileiros e constatamos que os mesmos possuem uma estrutura de *layout* bem definida pelo Banco Central do Brasil, porém bastante complexa devido a grande variedade de padrões de fundos e estilos de fontes utilizadas.

Elaboramos um estudo que propõe-se a segmentar e armazenar o conteúdo do cheque bancário brasileiro sem preenchimento utilizando o Processamento Digital de Imagens e padrões de documentação. Os principais resultado desse trabalho estão relacionados ao uso de ferramentas de morfologia matemática binária para a segmentação do formulário do cheque bancário e ao uso padrões de documentação para a codificação das estruturas lógica e física de um documento.

Palavras Chave: padrões de documentação, ODA, SGML, morfologia matemática binária, estrutura física, estrutura lógica.

Abstract

The work presents a study about the extraction of logical and physical structures from images of Brasilian bank checks. We have investigated different standards and norms used to store logical and physical structure of documents. Then, we have carried out a theoretical study between the two main industrial standards: SGML (*Standard Generalized Markup Language*) e o ODA (*Open/Office Document Architecture).*

In this work, different types of Brazilian bank checks were employed. In spite of the fact that they have a specific structural layout defined by Central Bank of Brazil, we have to deal with a great variety of bank check background and font styles.

We intend in this work study to segment and store the contents of Brazilian bank checks using digital image processing and standards of documentation. The main results of the dissertation are related to the use of binary mathematical morphology to the segment the form of bank check and the use of standards of documentation to codify the logical and physical structures of the document.

Keywords: standards of documentation, ODA, SGML, binary mathematical morphology, physical structure, logical structure.

Capítulo I
Introdução

Hoje muitos documentos estão sendo criados, distribuídos e apresentados sob papel. Ou seja, o papel é o meio principal utilizados em livros, manuais, jornais, revistas, correspondências, documentos bancários entre outros. Entretanto, o baixo custo e ao mesmo tempo o aumento da *performace* do *hardware* fará com que o armazenamento e a distribuição de documento eletrônicos sejam o meio predominante [WON82], [HU94].

Neste trabalho elaboramos um estudo sobre a extração das estruturas lógica e física a partir de uma imagem de um cheque bancário brasileiro. Essas estruturas nos possibilitam codificar uma imagem digital todo conteúdo do cheque fazendo com que o armazenamento e pesquisa a esses documento se torne mais simples e mais rápido. Além disso, este trabalho faz parte de um sistema Multi-agente de processamento automático de cheque bancário, tendo a função de auxiliar no processo de segmentação.

Este trabalho faz parte do projeto integrado de pesquisa do CNPQ (n° 520324/96-0) sobre o estudo da viabilidade de um sistema de processamento e análise de imagens de documentos bancários, além de fazer parte do grupo de pesquisa ABCPR *Group* (*Automatic Bank Checks Processing Research Group*).

1.1 Desafio

O processamento de imagens de cheques bancários não é uma tarefa simples de ser executada. Atribuímos isso a alguns fatores, tais como a complexidade do *layout* do documento, e a grande quantidade de padrões de fundos de cheques, padrões esses, necessários para conservar os requisitos de segurança dos bancos [BCO83].

Alguns estudos na literatura, tais como [KOE97], usam o conhecimento prévio do padrão do fundo do cheque para removê-lo, e, por conseqüência recuperar as informações necessárias. Porém achamos que esta solução não é compatível com a realidade do processamento de cheques, pois teríamos que conhecer todos os padrões de fundos dos cheques bancários brasileiros, além de ter que adequar o sistema a possíveis mudanças nos padrões de fundos dos cheques.

Tendo isso em vista, nosso desafio constitui em gerar as estruturas física e lógica a partir de uma imagem de um cheque bancário brasileiro em 256 níveis de cinza e 300 dpi, sem conhecimento prévio do padrão de fundo do cheque, bem como codificá-las em algum padrão existente no mercado. Para isso estudaremos e compararemos os padrões e normas mais aceitos no mercado atualmente.

1.2 Motivação

Para integrar os documentos existentes a uma base de dados, um sistema de Análise de Imagem de Documento se faz necessário. Esse sistema deverá ser capaz de extrair a informação estrutural de documentos impressos para criar bancos de dados. Um sistema de Análise de Imagem de Documento é muitas vezes definido como um processo, o qual transforma páginas de documentos em informações estruturadas para o processamento através de computadores.

Quando falamos em documentos, estamos falando de documentos que podem ser armazenados em algum dispositivo eletrônico, ou transmitido através de uma rede. Furuta em [FUR89] mostra que duas representações são essenciais para a Análise de Imagem de Documento:

- A primeira descreve a forma na qual o documento está formatado, e é como ele será enviado para o dispositivo de saída; é normalmente chamada de estrutura física.

- A segunda é manipulada pelo sistema do usuário e especifica a estruturação e o conteúdo do documento; é normalmente chamada de estrutura lógica.

Recentemente trabalhos com documentos eletrônicos têm enfatizado a importância da manutenção das estruturas do documento em todos os estágios de edição, armazenamento e transmissão. A manutenção da estrutura lógica básica do documento pode trazer grandes vantagens para um sistema de processamento de documentos [BRO89].

1.3 Proposta

Propomos neste trabalho fazer um estudo dos padrões e normas existentes no mercado capazes de armazenar as estruturas física e lógica do cheque bancário brasileiro, bem como um estudo comparativo entre os dois principais padrões. É importante salientar que o *layout* da estrutura do cheque bancário deverá ser completamente codificada em um padrão.

Para que possamos extrair as estruturas acima mencionadas, será necessário a obtenção de um método que forneça bons resultados na extração dos dados impressos no formulário do cheque. Usaremos para isso o Processamento Digital de Imagens, mais especificamente, as operações da morfologia matemática binária.

O conhecimento da estrutura do documento tem um papel importante na análise do documento [TAN91][AKI95], por este motivo estudaremos as regras e a legislação vigente no país no que diz respeito aos cheques bancários brasileiros. Na Figura 1.1 podemos visualizar, a seqüência do trabalho proposto.

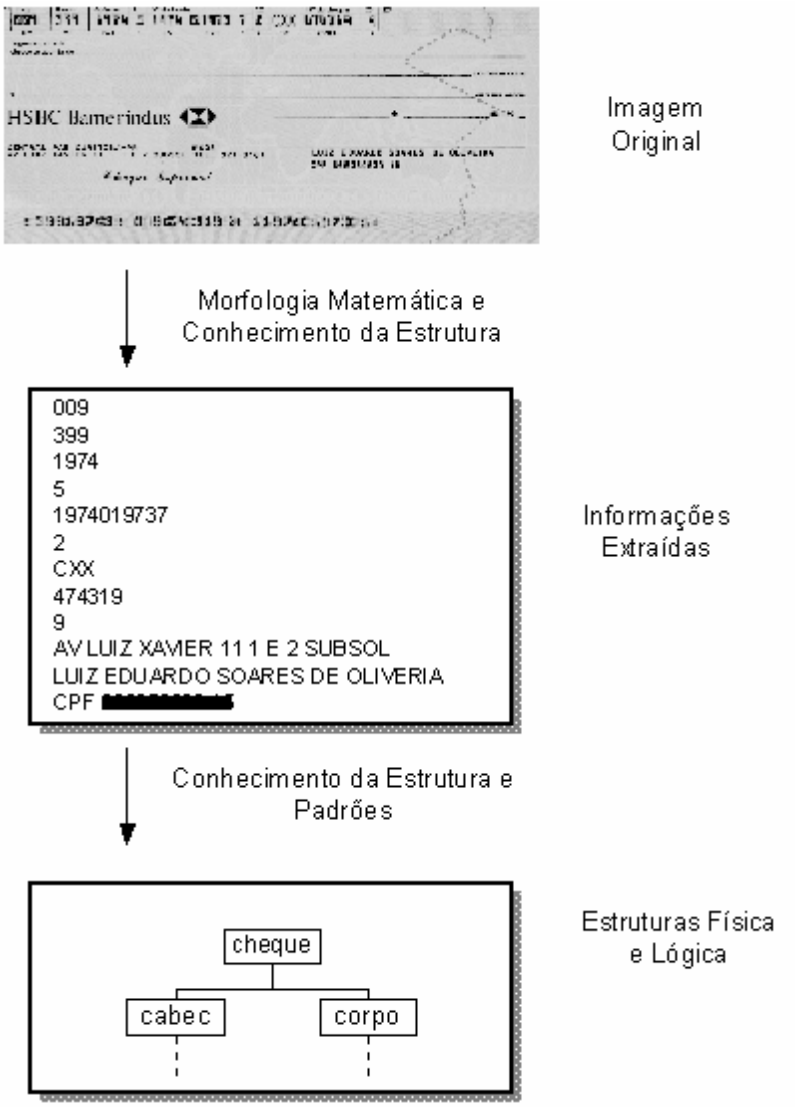

Figura 1.1: Visualização do trabalho proposto.

1.4 Contribuição

A principal contribuição deste trabalho está relacionada ao processamento de cheques bancários brasileiros, porém existe uma contribuição considerável na área de padrões para a construção de estruturas física e lógica de um documento.

Em se tratando do processamento de imagem propriamente dito, fizemos um estudo geral sobre o Processamento Digital de Imagens, e elaboramos um método baseado em morfologia matemática para executar a Análise da Imagem do

Documento, a fim de recuperar todas as informações pré-impressas no formulário do cheque. Buscamos na literatura o estado atual da arte na área de interpretação de documentos.

Em se tratando de padrões, fizemos um levantamento sobre os principais padrões existentes no mercado, e as principias ferramentas (conversores, compiladores, etc..) que envolvem esses padrões. Elaboramos um estudo teórico sobre os dois padrões mais importantes atualmente, que são SGML (*Standard Generalized Markup Language*) e o ODA (*Open/Office Document Architecture),* bem como um comparativo entre os mesmos.

Uma contribuição particular deste trabalho, é o fato de fazer parte de um grupo de pesquisa sobre processamento de cheque bancários, o ABCPR Group (*Automatic Bank Checks Processing Research Group*). O ABCPR *Group* é uma parceria entre o CEFET-Pr e a PUC-Pr com convênio com a ÉTS/Université du Quebec-Canadá, com o objetivo de desenvolver pesquisas relacionadas ao tratamento automatizado de cheques bancários, e tem como uma de suas diretrizes trabalhar em uma plataforma Multi-agentes. Este tipo de plataforma sistema possui agentes (processos) que trabalham em conjunto para resolver problemas que estão além de suas capacidades individuais, apresentando assim vantagens significantes em relação às plataformas centralizadas [O`HA96].

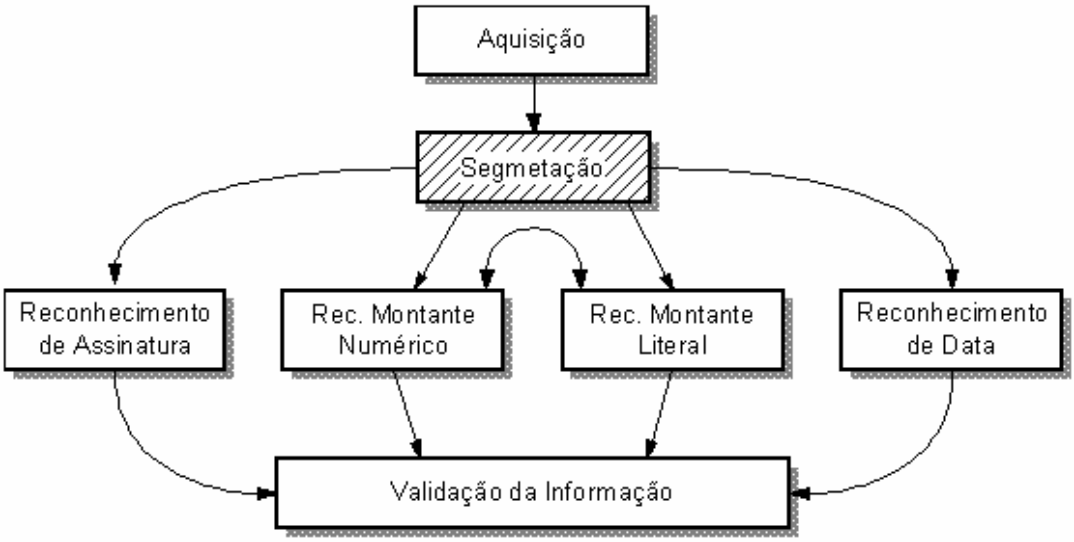

Figura 1.2: Estrutura global do Projeto ABCPR Group.

Dentro do sistema Multi-agentes, este trabalho está compreendido no módulo de segmentação. Na Figura 1.2 mostramos a estrutura global do projeto do ABCPR Group.

1.5 Organização

Nesta seção apresentaremos a estrutura desta dissertação. Este trabalho foi dividido em 7 capítulos, os quais apresentaremos a seguir:

Neste Capítulo apresentamos os principais objetivos e contribuições deste trabalho.

O Capítulo II, apresenta um estudo bibliográfico afim de mostrar o estado atual da arte na área de interpretação de documentos. Apresentamos várias técnicas utilizadas para a Análise de Imagem de Documento, bem como um estudo básico sobre Processamento Digital de Imagens. No final deste capítulo, são mostrados vários padrões de documento existentes no mercado, bem como algumas ferramentas, tais como conversores de formato, compiladores e *viewers.*

No Capítulo III estudaremos a estrutura do *layout* do cheque bancário brasileiro. Apresentaremos as normas e legislação do Banco Central do Brasil para a confecção de cheques bancários.

No Capítulo IV, elaboramos um estudo teórico sobre os dois principais padrões no mercado, o SGML (*Standart Generalized Markup Language*) e o ODA (*Open/Office Document Architecture).* Apresentamos o SGML com um exemplo capaz de exemplificar boa parte da sua sintaxe. Para o ODA, mostramos como são construídas as estruturas física e lógica, e também como as mesmas se relacionam. Para finalizar este capítulo, apresentaremos um comparativo entre eles, apresentando alguns pontos fortes e fracos de cada lado.

No Capitulo V, apresentaremos um processamento necessário para se obter as estruturas lógica e física do cheque bancário a partir de uma imagem digital em níveis de cinza. Ainda neste capítulo, apresentaremos como codificar o resultado da segmentação em SGML e ODA.

No Capítulo VI, abriremos um espaço para discussões sobre o trabalho em questão, tais como as dificuldades encontradas, apresentação de algumas considerações sobre as metodologias implementadas na busca da segmentação de imagens, e também algumas colocações que achamos necessárias para o aperfeiçoamento deste trabalho.

No Capítulo VII apresentaremos as conclusões do trabalho, as nossas contribuições e os trabalhos futuros para o aperfeiçoamento do que apresentamos.

Capítulo II
Estado da Arte

2.1 Introdução

Durante os últimos anos, a análise e reconhecimento de documentos eletrônicos vem ganhando importância rapidamente. Neste capítulo mostraremos o desenvolvimento deste campo de pesquisa, bem como as técnicas incorporadas de varias áreas de pesquisas relacionadas. As áreas discutidas são Análise de Imagem de Documento, Análise de Imagem de Documento Baseado em Conhecimento e Análise de *Layout* de Documento. Ainda neste capítulo, mostraremos alguns padrões de documentos capazes de armazenar as estruturas lógica e física do documento e também um breve estudo sobre Processamento Digital de Imagens.

2.2 Análise de Imagem de Documento

Em um problema de interpretação de documento, a primeira tarefa que deve ser realizada é a segmentação e a classificação de todos os blocos da imagem do documento. Isso constitui a Análise de Imagem de Documento, e é o aspecto mais estudado dentro do campo da interpretação de documentos.

É interessante notar que a Análise de Imagem de Documento está baseada no campo de Processamento Digital de Imagem (PDI) e especificamente em técnicas como segmentação e filtragem, as quais podem extrair objetos de interesse da imagem.

2.2.1 Técnicas *Top-Down*

As técnicas que analisam o documento do geral para o detalhe são conhecidas como *Top-Down*. Abaixo apresentaremos algumas dessas técnicas.

Wong em [WON82] propôs o uso de um *run-length smoothing algorithm* (RLSA) para segmentar e classificar documentos eletrônicos em regiões de texto.

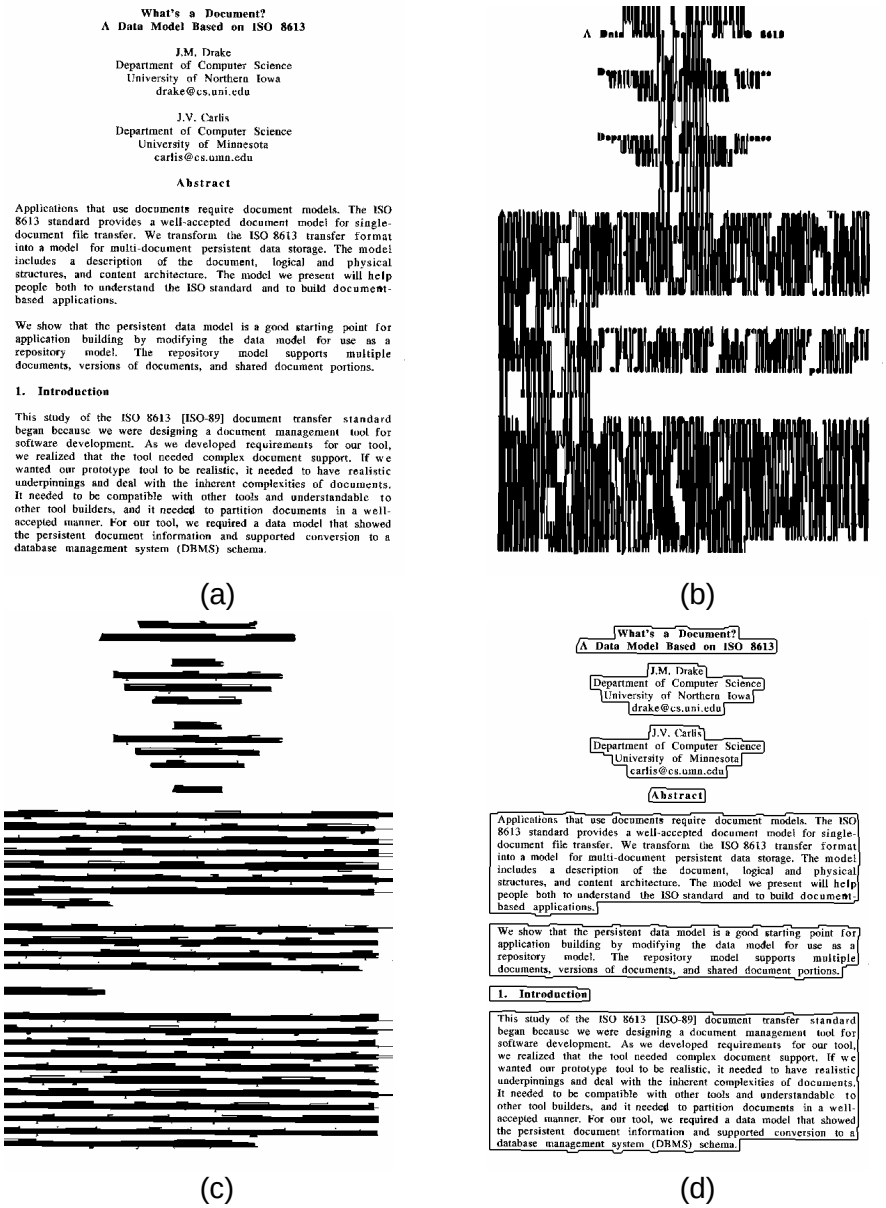

Figura 2.1: (a) Imagem Original, (b) Fechamento morfológico vertical, (c) Fechamento morfológico horizontal e (d) Imagem resultante com os blocos marcados.

Nesse método, o documento é "manchado" pela conversão de uma seqüência de *pixels* brancos de tamanho menor que um dado limiar para *pixels* pretos, ou seja, uma seqüência de entrada é modificada para se obter uma outra seqüência

de saída, onde os *zeros* são convertidos para *uns* se o número de *zeros* adjacentes é menor que um determinado limiar.

Esta operação de manchamento é feita separadamente nas direções horizontais e verticais, e o resultado da imagens manchadas são combinados usando a operação lógica E para resultar a imagem final manchada. A segmentação desta imagem manchada é então feita usando a projeção dos contornos, produzindo os blocos impressos na imagem original. É importante ressaltar que a escolha do limiar é um fator importante no processo, pois ele definirá a qualidade do resultado obtido, já que pode unir elementos indesejadamente, como no caso de colunas muito próximas.

Outra maneira de se implementar este método, é através de morfologia matemática. Os manchamentos horizontal e vertical, são executados por operações morfológicas de fechamento, com elementos estruturante “linha horizontal” e “linha vertical” respectivamente (Figura 2.1). O resultado obtido pela operação lógica E entre as imagens manchadas horizontal e vertical, deve ser novamente fechada por um elemento estruturante “linha horizontal”. Desse resultado devem-se extrair as bordas, que devem representar os blocos da imagem.

Nagy em [NAG91] propôs o uso de cortes horizontais e verticais recursivos para dividir um documento em seus blocos constituintes. Esse método é conhecido como “*X-Y cuts*”. Nesse método o documento é recursivamente dividido usando a técnica de projeção de contornos. Esta projeção é calculada (i.e., todos ao *pixels* pretos são contados ao longo de toda a linha/coluna) na direção horizontal bem como na vertical. Um pixel zero indica a presença de um “espaço branco”, i.e., um espaço ao longo de todo tamanho/altura da imagem do documento livre de impressão. Um corte horizontal ou vertical é então feito ao longo do espaço em branco, produzindo duas sub-imagens. Para cada uma dessas sub-imagens, a mesma operação é feita em outra direção (i.e., se o primeiro corte foi na vertical, então o próximo deverá ser na horizontal). Deste modo, o processo é recursivamente aplicado para cada sub-imagem do documento até não existir mais a possibilidade de executar cortes. O blocos de *pixels* pretos resultantes são considerados a representação dos blocos impressos indivisíveis no documento.

Já Baird em [BAI90] propõe um método diferente, baseado na análise do fundo da página do documento antes que dos objetos, para localizar os blocos de linhas. Esse método é conhecido pelo nome de *partição do fundo da imagem.* Ele consiste em delimitar os objetos pelo meio das regiões brancas modelizadas por retângulos. Todos os retângulos brancos máximos são identificados e classificados, dos maiores e menores. Os maiores retângulos são supostamente correspondentes às margens, espaços entre colunas ou entre parágrafos. Sua seleção permite realizar uma partição do fundo da imagem em blocos estruturados. A vantagem de considerar o fundo antes que os objetos resulta em maior independência frente à natureza do texto a se extrair (o fundo é o delimitador mais comum em documentos de qualquer tipografia ou língua). Um inconveniente desse estudo reside na escolha do tamanho mínimo necessário para um retângulo branco ser considerado como delimitador.

2.2.2 Técnicas *Bottom-Up*

De maneira inversa ao *Top-Down*, as técnicas *Bottom-Up* buscam primeiramente os menores componentes da imagem. Por exemplo, em um texto seriam analisados primeiramente os caracteres, em seguida as palavras e assim por diante.

O'Gorman em [O'GO92] apresentou um método chamado *Document Spectrum Analysis*. Nesse método, todos os componentes conectados da imagem são primeiramente detectados, e então os blocos são gerados na técnica *Bottom-Up* encontrando os vizinhos de cada componente e agrupando os diferentes componentes. Este método de segmentação produz resultados muito bons para vários tipos de documentos, particularmente para páginas de jornais.

Pratt em [PRA78] e Srihari em [SRI89] apresentam a Transformada de Hough com a função de segmentar e classificar blocos em imagem de documentos. A Transformada de Hough foi desenvolvida por Paul Hough em 1962 e patenteada pela IBM e tem sido reconhecida como uma ferramenta robusta para o reconhecimento de formas analiticamente definidas, ou ainda curvas em imagens ruidosas. Por outro lado,

o grau de complexidade limita fortemente o uso da Transformada de Hough. Outros métodos de Transformada de Hough Rápida foram propostos. Alguns deles seguem um dos dois estudos:

- Tentar usar mais informação do ponto de contorno, normalmente sua orientação, e combinar essa informação com as propriedades geométricas da forma a ser detectada;

- Usar um algoritmo probabilístico, no qual somente um sub-conjunto aleatório selecionado dos pontos de contorno são usados na decisão.

2.3 Análise de Imagem de Documento Baseado em Conhecimento

O uso de conhecimento sobre a estrutura básica de diferentes classes de documentos para analisar e identificar diferentes componentes do documento. É conhecido como Análise de Imagem de Documento Baseado em Conhecimento. Segundo Tang em [TAN91] o conhecimento profundo da estrutura do documento facilita a sua análise e seu reconhecimento.

A Análise de Imagem de Documento Baseado em Conhecimento tem sido usado em vários domínios de imagens (imagens aéreas, imagens médicas etc..). A importância do uso de técnicas baseadas em conhecimento na análise de imagens é mencionado em [SHA85]. A necessidade de usar várias fontes de conhecimento para a interpretação do documento, bem como a combinação de técnicas *Top-Down* e *Bottom-Up* para a segmentação do documento é descrita em [TAY92].

Várias outras estratégias de controle para análise de imagens tem sido desenvolvidas. Entre elas o uso de redes semânticas para representar o conhecimento [BUN84]. Sistemas baseados em regras de inferência tem se tornado popular para análise de imagens, particularmente em domínios quando as características da imagem e relações podem ser efetivamente representadas através de regras.

Sistemas baseados em regras tem a vantagem de poder ser representado de maneira declarativa. A desvantagem é que qualquer regra nova deverá ser incluída na base de regras do sistema. E pode levar à inconsistência com regras já existentes.

Uma das arquiteturas propostas para manutenção do conhecimento do domínio do sistema é o controle de arquitetura *blackboard*. Um sistema de controle *blackboard* provê um controle flexível do ambiente que facilita a iteração de resolução de problemas distribuídos e provê uma solução integrada [NIY94] .

2.4 Análise de *Layout* de Documento

Análise de *Layout* de Documento envolve a extração da estrutura física de um documento. Muitos trabalhos de Análise de Imagem de Documento utilizando técnicas *Top-Down* e conhecimento da estrutura do documento extraem completamente a estrutura física do documento.

Um conceito importante é encontrado em [FUR89]: O *layout* físico descreve o formato do documento, isto é, o posicionamento de cada campo dentro do documento, enquanto que o *layout* lógico ou estrutura lógica especifica o conteúdo do documento.

Watanabe em [WAT92] classificou os documentos em quatro classes baseado na dependência entre a *layout* físico e o lógico. A correlação aproximada entre as estruturas é mostrado na Figura 2.2.

- Classe 1: O documento tem sua própria estrutura de *layout* físico. Esse *layout* é definido como um formulário. Os separadores são especificados explicitamente. Dados individuais devem ser colocados em seus lugares pré-definidos. Um exemplo típico deste tipo é uma tabela ou um formulário. Neste caso as linhas verticais e horizontais tem o papel de delimitar os campos.

- Classe 2: O documento possui relacionamento geométrico entre os itens. A posição relativa de um item é determinado de acordo com seu estado alocado. A

estrutura do *layout* lógico coincide com o estrutura do *layout* físico. Um exemplo típico é um catálogo de biblioteca. Cada item esta colocado em espaços pré-determinados.

- Classe 3: Apesar dos documentos terem seu próprio *layout* de estrutura, o relacionamento entre os itens são determinados pelo *layout* lógico da estrutura. É claro que o *layout* físico é a base para compor documentos. Um exemplo dessa classe são os jornais.

- Classe 4: O documento não obedece nenhum tipo de *layout* físico. Exemplo claro disso temos em folhas de propagandas ou capas de revistas.

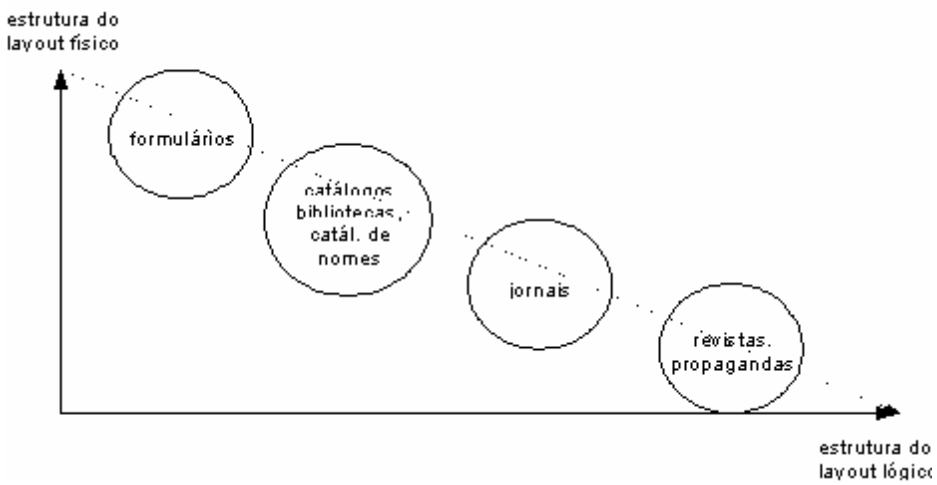

Figura 2.2 Documentos e seus *layout´*s.

Watanabe mostra em [WAT92] que os documentos da primeira classe podem ser especificados por uma árvore binária, onde esta árvore recebe o nome de estrutura de descrição. A árvore é dividida em duas classes: árvore de estrutura global e estrutura local.

O campo da automação de escritório tem feito uma contribuição substancial para a área de Análise de *Layout* de Documento. Doster em [DOS84] descreveu como a base de dados da descrição de um documento pode ser usada para representar conhecimento sobre as características da imagem de um documento. Peels descreveu em [PEE85] o processo oposto; i.e., com base na descrição lógica do

documento, (usando um padrão de codificação) transformar em uma representação física formatada.

Os padrões mais usados para representação da estrutura de um documento são o *Standard Generalized Markup Language (SGML)* e o *Open/Office Document Architecture (ODA)*. O problema da derivação da estrutura lógica do documento é importante para automação de escritório, visto que um documento logicamente analisado pode ser transmitido eletronicamente para outros lugares e reconstruídos a partir da estrutura lógica.

2.5 Processamento Digital de Imagens - PDI

As técnicas *Top-Down* mostradas acima, são eficientes para executar a tarefa de marcação dos blocos no cheque bancário. Entretanto, esses métodos trabalham com imagens binárias e com um fundo livre de ruído. O Processamento Digital de Imagens tem por objetivo a manipulação sistemática de imagens, com o intuito de gerar outra imagem que deve apresentar uma qualidade melhor para uma análise posterior. O processamento digital de imagens consiste de 3 níveis: baixo, médio e alto, como mostrado na Figura 2.3.

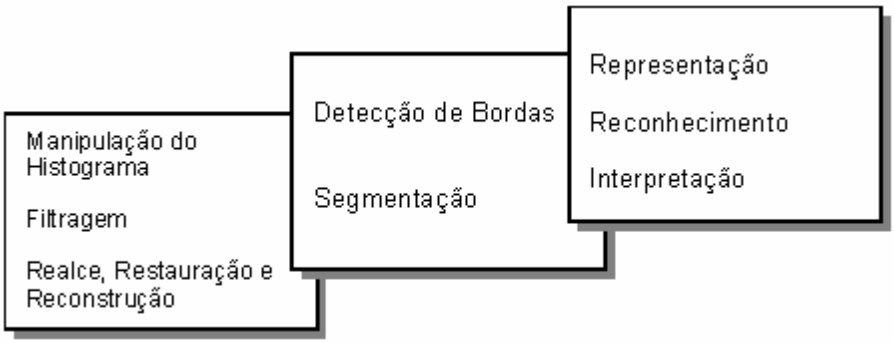

Figura 2.3: Níveis do Processamento Digital de Imagens.

Após a aquisição da imagem é necessário melhorar a sua qualidade afim de realçar as informações relevantes ao seu processamento usando processos de filtragem, manipulação de histograma, reconstrução entre outros. Feito isso, a segmentação e a detecção de bordas surgem num segundo estágio, particionando a

imagem e extraindo dela somente as primitivas de interesse de seu analisador. O próximo passo é, então, interpretar todos os valores fornecidos pelo processo.

Tendo em vista que as imagens definidas pelo ABCPR Group devem ser adquiridas em 300 dpi e 256 níveis de cinza, abaixo apresentaremos o ferramental necessário que nos possibilitam converter essas imagens para binárias e com um fundo livre de ruído.

2.5.1 Imagem Digital

Uma imagem é representada por uma função f(x,y) que apresenta na coordenada (x,y) sua intensidade de luz. Esta intensidade aparece num intervalo que vai de zero ao infinito, por ser a luz uma forma de energia. Por outro lado, as imagens no mundo real são vistas por nossos olhos graças a dois fatores fundamentais que são a quantidade de luz que incide em um objeto e a quantidade dessa luz refletida por esse mesmo objeto. É sabido que o computador trata a imagem de forma númerica, por esta razão tratamos a imagem como sendo uma matriz de N x M.

Essa matriz apresenta as coordenadas espaciais da imagem digitalizada (amostragem), e também sua quantificação. A amostragem relaciona-se à escala da imagem e suas coordenadas espaciais (256x256 pixels, 512x512 pixels entre outras), já a quantificação tem como medidas as resoluções em níveis de cinza, que geralmente são de 16, 32, 64, 128 ou 256.

Achamos importante essas considerações, pois a imagem digital é o objeto base no Processamento Digital de Imagem.

2.5.2 Histograma

O histograma de uma imagem digital é um gráfico que mostra a distribuição dos pixels da imagem entre os níveis de cinza que a mesma possui. Ele proporciona uma visão global da distribuição.

Pode-se observar a existência de um número maior de pixels claros tanto na imagem quanto no histograma. Se a imagem do cheque fosse mais escura, consequentemente teríamos um histograma parecido com o da Figura 2.5.

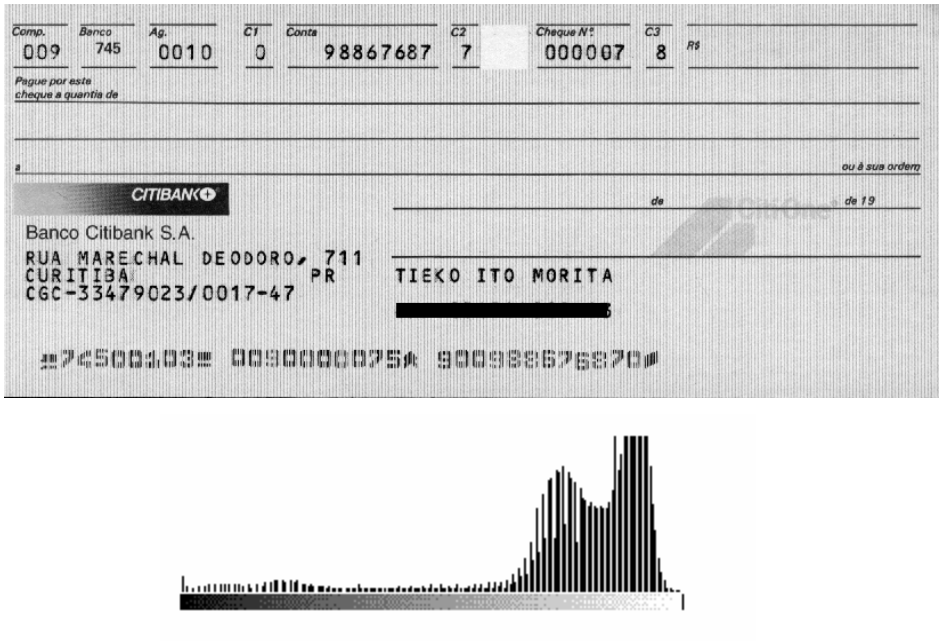

Figura 2.4: Cheque e o seu respectivo histograma.

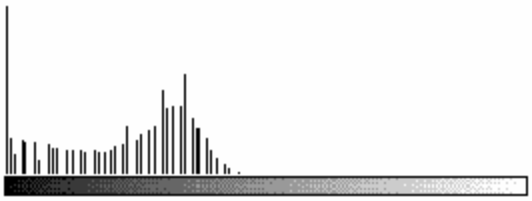

Figura 2.5: Histograma de uma imagem escura.

Rabbani em [RAB92] define o histograma da seguinte maneira: “Considerando-se uma imagem f(j,k), que tem Nx pixels por linha e Ny linhas, com Nb bits por pixel, o histograma hf(i), para cada valor i, representa o número de vezes que este valor i aparece na imagem.” Onde i representa um valor inteiro qualquer no intervalo de 0 a 255, o qual representa os níveis de cinza de uma imagem.

2.5.3 Hiperbolização Quadrática do Histograma

O primeiro passo para converter uma imagem de um cheque bancário com 256 níveis de cinza para uma imagem binária é a remoção do fundo. Segundo Santos em [SAN97] a solução encontrada para o problema do fundo dos cheques foi a de efetuar um pré-tratamento, fazendo com que a luminosidade fosse melhorada e seus detalhes realçados, originando, assim, uma imagem mais inteligível. O processo que mostrou poder alterar a panorâmica das imagens de cheques foi o método da hiperbolização quadrática do histograma, apresentado por Cobra em [COB96] e Santos em [SAN97] (Figuras 2.6 e 2.7).

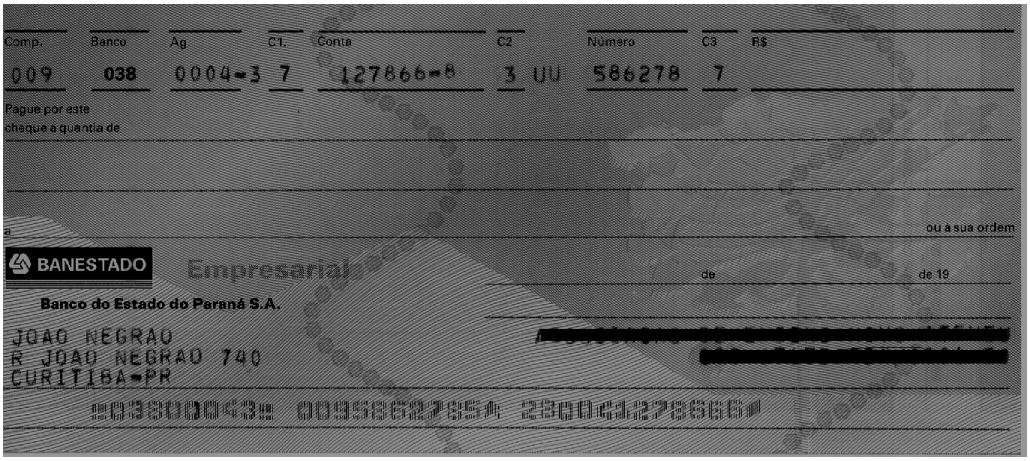

Figura 2.6: Cheque Bancário antes da hiperbolização quadrática do histograma.

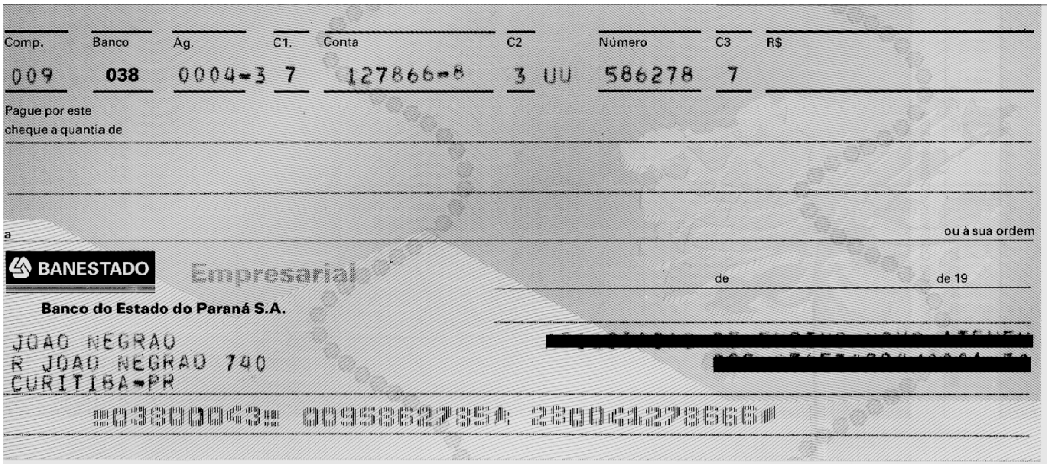

Figura 2.7: Cheque Bancário após a hiperbolização quadrática do histograma.

A hiperbolização quadrática do histograma, é baseada em um modelo do sistema visual periférico humano que leva em conta o fato de que este se acomoda à intensidade média da cena observada, e não à intensidade dos pixels individuais, como subentende o modelo logarítmico utilizado na hiperbolização de histograma.

O resultado é uma transformação que fornece uma distribuição mais espalhada dos níveis de cinza, evitando a concentração excessiva nos tons escuros verificada na técnica da hiperbolização.

2.5.4 Limiarização (*Thresholding*)

A limiarização é um método que se baseia no histograma da imagem, buscando encontrar ali "regiões bem definidas", afim de poder efetuar a divisão da imagem em objetos ou regiões. A continuidade dos níveis de cinza é a primitiva de maior valor na segmentação por região. Percebemos assim, que a limiarização efetua a subdivisão da imagem em função das "regiões significativas" contidas no histograma da imagem.

Seja uma imagem digital qualquer, onde f(x,y) representa o nível de cinza de um ponto qualquer, supondo a existência de apenas dois objetos, claramente definidos em função dos seus níveis de cinza: um fundo escuro sob um objeto claro. Se observarmos o histograma dessa imagem, fica claro que ao selecionarmos um limiar T que esteja entre as duas regiões do histograma, que representam os objetos da imagem, poderemos extrair qualquer um desses objetos. Assim:

$$g(x,y) = \begin{cases} 1 \; se \;\; f(x,y) > T \\ 0 \; se \;\; f(x,y) \leq T \end{cases}$$

onde g(x,y) representa o ponto limiarizado.

No entanto o estudo torna-se mais interessante se levarmos em consideração uma imagem com dois tipos de objetos num fundo escuro. Neste caso o histograma passaria ser de bimodal no caso anterior para multimodal.

Para que seja possível a separação desses objetos, devemos supor que:

$$g(x,y)=\begin{cases} objeto\ 1\ se\ T1 < f(x,y) \le T2 \\ objeto\ 2\ se\ f(x,y) > T2 \\ fundo\ se\ f(x,y) \le T1 \end{cases}$$

Pela multimodalidade do histograma, parece ser fácil perceber que esse tipo de limiarização apresenta um maior grau de complexidade, em função da dificuldade de se estabelecer múltiplos limiares que efetivamente isolem regiões de interesse.

Em [GON92] são apresentados três tipos básicos de limiarização de uma forma clara e concisa.

2.5.4.1 Método de Otsu

Segundo Santos em [SAN97] a aplicação da limiarização pelo método de Otsu em uma imagem que sofreu uma hiperbolização quadrática do histograma fornece resultados muito bons. Nesta seção descrevemos o método proposto por Otsu [SAH88].

Este método tem a vantagem de não utilizar nenhum conhecimento *a priori* da imagem, baseando-se apenas em seu histograma em níveis de cinza. A idéia é encontrar no histograma um limiar ótimo que separe os objetos da imagem através da construção de duas classes $C_0 = \{0,1,...,t)\ e\ C_1 = \{t+1, t+2,...,l-1\}$, a partir de um nível de cinza arbitrário, utilizando para isso a análise de discriminante.

Para encontrar o limiar ótimo pode-se minimizar-se a seguinte função (mais simples segundo Otsu) de critérios que se referem a t:

$$h = \frac{s_B^2}{s_T^2}$$

Como s_T^2, que é a variância total, é independente do nível de cinza, é necessário apenas maximizar a função s_b^2, que é a variância inter-classes. O limiar t ótimo será definido da seguinte forma:

$$t^* = ArgMax\ s_b^2$$

Para tanto temos que:

$$s_T^2 = \sum_{i=0}^{l-1} (i - m_T)^2 P_i \qquad m_T = \sum_{i=0}^{l-1} ip_i$$

$$s_b^2 = w_0 w_1 (m_1 m_0)^2 \qquad w_1 = 1 - w_0 \qquad w_0 = \sum_{i=0}^{t} P_i$$

$$m_1 = \frac{m_T - m_t}{1 - w_0} \qquad m_0 = \frac{m_t}{w_0} \qquad m_t = \sum_{i=0}^{t} ip_i$$

onde w_0 e w_1 correspondem às variâncias nas classes C_0 e C_1. m_0 e m_1 suas respectivas médias. m_T e m_t, as médias total e por classe. s_B^2, s_T^2 as variâncias inter-classes e total, respectivamente.

2.5.5 Morfologia Matemática Binária

O principio básico da morfologia binária consiste em extrair informações relativas à geometria e à topologia de um conjunto desconhecido de uma imagem. Na morfologia matemática binária, na vizinhança de cada pixel da imagem original, é procurada uma determinada situação de pontos pretos e brancos. Quando esta situação existe, ao pixel correspondente da imagem é dado um rótulo “verdadeiro”, caso contrário “falso”. Portanto uma operação morfológica binária é determinada a partir da vizinhança examinada ao redor do ponto central. A situação acima referida é procurada por um conjunto bem definido chamado elemento estruturante.

Facon em [FAC96] define elemento estruturante como: “Um conjunto completamente definido e conhecido (forma, tamanho), que é comparado, a partir de

uma transformação ao conjunto desconhecido da imagem.“ O resultado dessa transformação permite avaliar o conjunto desconhecido. Isso se dá pois o formato e o tamanho do elemento estruturante possibilitam testar e quantificar de que maneira o elemento estruturante “está ou não está contido” na imagem.

A morfologia matemática age sobre imagens digitais a partir de elementos estruturantes geralmente definidos em uma malha retangular, como podemos verificar na Figura 2.8.

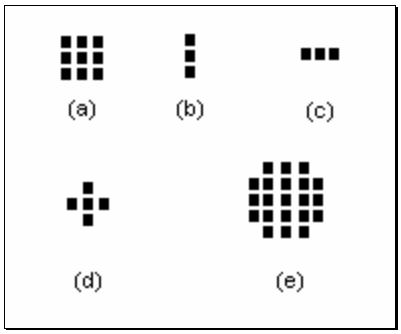

Figura 2.8: Exemplos de malhas de elementos estruturantes. (a) quadrado, (b) linha vertical, (c) linha horizontal, (d) cruz e (e) rhombus.

Abaixo apresentamos os operadores elementares da morfologia matemática que são a erosão e a dilatação.

2.5.5.1 Erosão

Serra em [SER82] define a operação de erosão binária da seguinte maneira: A erosão de um conjunto X pelo elemento estruturante B é:

$$ero^{B}(X) = X\ ero\ B = \{x \in e : B_x \subset X\}$$

A erosão causa os seguintes efeitos em uma imagem: As partículas são diminuídas, as partes da imagens que possuem tamanho menor que o elemento estruturante desaparecem, os buracos existentes aumentam e os objetos que compõem a imagem ficam mais distantes uns dos outros. Na Figuras 2.9 e 2.10

mostramos a imagem original e a imagem erodida por um elemento estruturante quadrado em três iterações.

Figura 2.9: Imagem binária original.

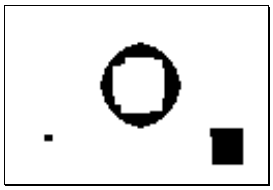

Figura 2.10: Imagem erodida com um elemento estruturante quadrado com três iterações.

Podemos observar neste exemplo que o menor quadrado quase desapareceu, e se houvesse mais uma iteração ele desapareceria. O buraco presente no meio do círculo aumentou e o quadrado maior ficou menor do que o original. Em todos os casos, o objeto ficou menor do que o original, e esse é o efeito principal da erosão.

2.5.5.2 Dilatação

Serra em [SER82] define a operação de dilatação binária da seguinte maneira: A dilatação de um conjunto X pelo elemento estruturante B e:

$$dil^{B}(X) = X\ dil\ B = \{x \in X : B_x \cap X \neq \varnothing\}$$

A dilatação causa os seguintes efeitos na imagem: aumento dos objetos que compõem a imagem, preenchimento de pequenos buracos existentes na imagem e conexão dos os objetos mais próximos. Mostramos isso na Figura 2.11.

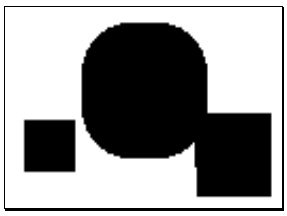

Figura 2.11: Imagem da Figura 2.9 dilatada por um elemento estruturante quadrado com cinco iterações.

Podemos verificar que os efeitos acima citados se manifestam. O buraco no meio do círculo desapareceu, o quadrado maior conectou-se ao círculo e o quadrado menor aumentou seu tamanho. Ou seja, todos os objetos tiveram seus tamanhos acrescidos, o que é o efeito principal da dilatação.

Vimos até agora a capacidade que a erosão e a dilatação têm em corrigir defeitos na imagem e eliminar partículas ou buracos indesejáveis. Porém ambas alteram o tamanho dos objetos da imagem, a erosão diminui e a dilatação aumenta. Veremos agora duas operações que eliminam partículas indesejáveis sem alterar o tamanho dos objetos da imagem, a abertura e o fechamento.

2.5.5.3 Abertura

Facon em [FAC96] define a operação de abertura binária utilizando as definições de erosão e dilatação acima apresentadas da seguinte maneira: A abertura de um conjunto X pelo elemento estruturante B:

$$abe^{B}(X) = X\ abe\ B = (X\ ero\ B)\ dil\ \tilde{B} = dil^{\tilde{B}}\left(ero^{B}(X)\right)$$

A abertura nivela os contorno pelo interior, separa os objetos, elimina partículas de tamanho inferior ao elemento estruturante com a vantagem de que os elementos restantes na imagem ficam quase idênticas às originais, mas o conjunto aberto é menos rico em detalhes que o conjunto original. Nas Figuras 2.12 e 2.13 mostramos o efeito da abertura binária.

Figura 2.12: Imagem original com ruído.

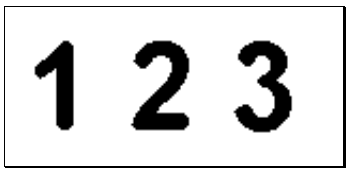

Figura 2.13: Imagem da Figura 2.12 aberta com elemento estruturante cruz.

Observamos que todo o ruído existente na Figura 2.13 foi eliminado com o processo de abertura, e o imagem resultante é muito parecida com a imagem original.

2.5.5.4 Fechamento

Facon em [FAC96] define a operação de fechamento binário utilizando as definições de erosão e dilatação acima apresentadas da seguinte maneira: O fechamento de um conjunto X pelo elemento estruturante B:

$$fec^{B}(X) = X \;\; fec\, B = (X \;\; dil\; B)\, ero\, \tilde{B} = ero^{\tilde{B}}\left(dil^{B}(X)\right)$$

O fechamento suaviza as fronteiras pelo exterior, preenche os buracos no interior das partículas inferior em tamanho em relação ao elemento estruturante, conectas as partículas próximas, os objetos restantes após o fechamento ficam quase idênticos aos originais, o conjunto fechado é mais regular do que o conjunto inicial e como na abertura, o conjunto fechado é menos rico em detalhes que o conjunto inicial.

Figura 2.14: Imagem original antes do fechamento binário.

Figura 2.15: Imagem da Figura 2.14 após o processo de fechamento com elemento estruturante cruz.

Como podemos comprovar, os efeitos do fechamento são visíveis na Figura 2.15, onde o buraco no quadrado desapareceu e os dois objetos foram conectados sem alteração no tamanho dos objetos, a não ser pelo local da conecção.

A abertura e o fechamento diferem da erosão e da dilatação pela propriedade da idempotência. Essa propriedade é crucial no processos de filtragem [FAC96].

2.5.5.5 Condicionalidade e Reconstrução Binária

Devido a necessidade de restringir o processamento da imagem em uma região específica da imagem surgem os conceitos de dilatação e erosão condicional. Esses operadores realizam processamento em um subconjunto Z do conjunto inicial X.

Com base nas definições vistas até então Facon em [FAC96] define erosão condicional da seguinte maneira: “A erosão condicional do subconjunto Z de X pelo elemento estruturante B em relação ao conjunto X, sendo que Z está incluído em X”.

$$ero_{cX}^{B}(Z) = \left(ero^{B}\left(Z \cup X^{c}\right)\right) \cap X$$

E de maneira análoga a definição de dilatação condicional.

$$dil_{cX}^{B}(Z) = \left(dil^{B}(Z)\right) \cap X$$

Quando a uma operação condicional é executada, o rumo da operação é modificado. Por exemplo: No caso de uma dilatação condicional, a forma de expansão não é isotrópica, ou seja, a dilatação ocorrerá dentro do conjunto Z.

Figura 2.16: Imagem do texto original.

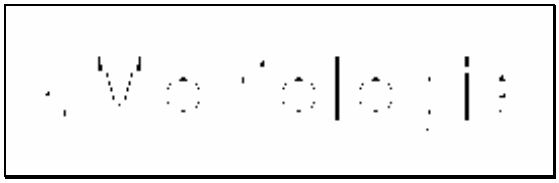

Figura 2.17: Imagem dos marcadores.

Na Figura 2.16 temos a imagem original e na Figura 2.17 temos a imagem dos marcadores. Os marcadores são resultantes de um processo anterior onde os objetos da imagem original sofreram desgastes. O objetivo é obter a imagem da Figura 2.16 dilatando a imagem da Figura 2.17. Se realizarmos uma operação de dilatação simples, jamais conseguiremos atingir a nossa meta, pois os conjuntos serão dilatados de forma isotrópica. A solução é uma dilatação condicional. Nas Figuras 2.18 e 2.19 vemos claramente o conceito da condicionalidade.

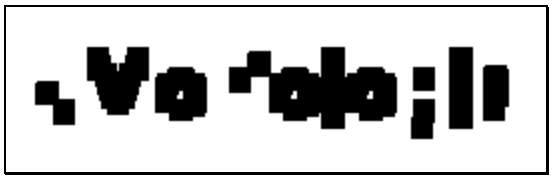

Figura 2.18: Dilatação normal com cinco iterações.

Figura 2.19: Dilatação condicional com cinco iterações.

Podemos notar que na Figura 2.18 a dilatação agiu de uma maneira convencional, ou seja, todos os objetos contidos na imagem foram dilatados incondicionalmente. Já na Figura 2.19 podemos notar que os objetos foram dilatados mas somente dentro do conjunto Z (imagem original da Figura 2.16).

Tendo o conceito de condicionalidade definido, podemos apresentar a idéia de reconstrução binária. A reconstrução está baseada no conceito de “marcador”, ou seja, a partir do marcador da imagem, que é um subconjunto da mesma, podemos reconstruir todos os objetos conexos dessa imagem [VIN93]. Através da reconstrução podemos eliminar os padrões indesejados sem deformar os padrões que buscamos.

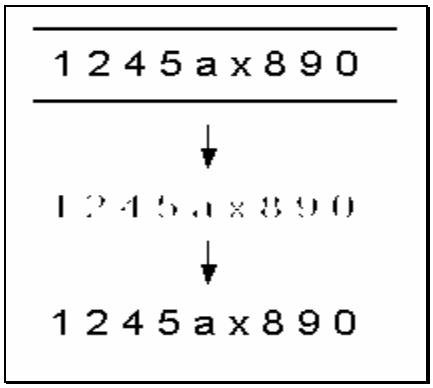

Figura 2.20: Estágios da reconstrução binária.

Na Figura 2.20 mostramos como é possível utilizar a reconstrução sem perder os padrões desejados. Neste exemplo o objetivo era excluir as duas bordas que aparecem na imagem inicial. Para resolver esse problema foi aplicado uma erosão com um elemento estruturante linha vertical com o intuito de eliminar as duas bordas. Feito isso nos sobrou a imagem intermediária, que possui marcadores de todos os caracteres existentes na imagem inicial menos das bordas. Isso é primordial para que o processo da reconstrução funcione adequadamente, pois a simples existência de um pixel da borda iria reconstruí-la juntamente com os caracteres. Para obter a última imagem fizemos uma dilatação condicional usando como máscara a imagem inicial.

2.5.5.6 Reconstrução Híbrida Rápida

Vincent em [VIN93] propôs várias implementações eficientes do processo de reconstrução binária a partir de um elemento estruturante planar. Os elementos utilizados manipulam o elemento estruturante decomposto conforme o sentido de rastreamento seguido. Se for planar, a sua decomposição também deve fornecer elementos estruturantes planares. Um dos algoritmos que foi implementado é chamado de reconstrução híbrida rápida e pode ser utilizado para imagens em níveis de cinza e binárias, sendo eficiente em ambas as situações. Nesse caso a reconstrução é total. Esse algoritmo em termos de reconstrução é o mais rápido.

2.6 Padrões de Documentos

Os meios e publicações e editorações modernos são altamente influenciados pelo avanço da tecnologia, e em conseqüência disso existe uma necessidade crescente para a padronização da troca de documentação eletrônica. O número de ferramentas de processamento de texto é grande e variado, e os fabricantes das mesmas aplicam quaisquer princípios que as tornem líderes de mercado e doravante um padrão dominante. Tem existido muitos esforços para introduzir tais padrões no processamento eletrônico de texto e permuta de informação; entre eles estão o *Office/Open Document Architecture (ODA)*, o *Standard Generalized Markup Language (SGML)* e o *HyperText Markup Language (HTML),* usado na *World Wide Web (WWW)*. Os fabricantes dominantes no mercado de processadores de texto (Microsoft WORD e Novell WordPerfect) tem provido ferramentas para importar documentos uma das outras, mas tem feito um esforço muito pequeno para promover a inter-operabilidade dos seus documentos com os padrões acima. Outros padrões comuns usados são o LaTeX e o Rich Text Format (RTF).

A ACM (*Association for Computing Machinery)* que desenvolve produtos eletrônicos tais como dicionários *on-line*, hipertextos entre outros, usa SGML para algumas aplicações. Segundo Fox em [FOX92], isso simplifica a análise subsequente,

eliminando a necessidade de uma gramática sofisticada para deduzir a estrutura e organização dos documentos e seus elementos.

O *layout* semântico do SGML é definido no SGML *Document Type Definition* (DTD). Gartland mostra em [GAR96] que *DeMontfort University (DMU)* está pesquisando os formatos de documentos existentes para tentar prover um método para gerar SGML DTD´s para um conjunto de tipos de documentos diferentes. A DMU está concentrada no estudo de como documentos SGML podem ser gerados facilmente a partir de ambientes de processamento de texto populares.

Kohsari em [KOH96] mostra que a *University College London (UCL)* está pesquisando o estado atual do padrão ODA, HTML e implementações WWW bem como os conversores entre eles. A UCL também está desenvolvendo sistemas que permitem que um documento seja preparado em um formato para transferência, armazenado e recuperado em outro. Esse trabalho da UCL faz parte de um projeto maior chamado JEDI (*Joint Eletronic Document Interchange),* no qual estão envolvidas UCL e DMU.

Kirstein em [KIR96] avaliou todas as metas propostas no início para o projeto JEDI e concluiu que todos os objetivos foram alcançados, exceto os trabalhos envolvendo ODA. Os trabalhos mostraram que os fabricantes não estão investindo em ODA de uma maneira enérgica, os produtos ainda não tem uma funcionalidade adequada para satisfazer os usuários, e existe uma probabilidade que isso não mude.

Talvez a grande diferença do estado atual do ODA e do SGML está no fato de que para SGML são encontrados uma grande variedade de ferramentas de domínio público, enquanto que as ferramentas para ODA são muito difíceis de serem encontradas. Por exemplo: A Bull desenvolveu uma ferramenta chamada Bull Office Team View que possibilitava o acesso, compartilhamento e gerenciamento de documentos sobre uma rede de computadores, além da manipulação de arquivos ODA, mas a sua comercialização foi suspensa e é indicado para o seu lugar o Lotus Notes. Porém o Lotus Notes não tem a capacidade de manipular arquivos ODA.

Na seqüência descrevemos alguns padrões existentes no mercado, e entre eles SGML e ODA, os quais veremos com mais detalhes futuramente.

2.6.1 SGML - *Standard Generalised Markup Language*

SGML é um padrão internacional para especificar linguagens *markup*. Ele é descrito pela *International Standards Organization (ISO)* como ISO 8879:1986 documento = "*Information Processing – Text and Office System – Standard Generalised Markup Language (SGML)"*.

SGML permite que a informação seja definida em uma estrutura, independentemente de como essa informação será processada. Um dos arquivos SGML é o *Document Type Definition (DTD)*. Esse arquivo contém informação relativa de como os elementos são estruturados no arquivo SGML e como a informação é estruturada. Um documento escrito em SGML não contém informação de como formatar ou imprimir um documento. Tais coisas são deixadas para outros padrões tais como DSSSL (ISO 10179.2) and SPDL (ISO 10190).

2.6.2 HTML - *HyperText Markup Language*

HTML é uma linguagem *markup* usada na Internet para a construção de documentos WWW. Esforços por parte da *Internet Engineering Task Force (IETF)* tem sido feito para tornar HTML um padrão definitivo para a Internet. Atualmente IETF definiu HTML 2.0 como a versão base, e onde todo o trabalho de desenvolvimento deve ser baseado. HTML é um tipo de documento descrito em SGML e como tal, tem um DTD associado que pode ser compilado por um compilador SGML.

Abaixo mostramos um texto codificado em HTML.

```
<HTML>
<HEAD>
<TITLE></TITLE>
</HEAD>
<BODY BACKGROUND="fundon.jpg" BGCOLOR="#000000">
```

```
<FONT FACE="Arial Black">
<P ALIGN="center"><FONT SIZE="+7">  </FONT><FONT SIZE="-7"><IMG
SRC="ani_bemvindo.gif"></FONT></P>

<H1 ALIGN="LEFT"><FONT SIZE="-7"><I><FONT COLOR="Blue">
</FONT></I><I><FONT COLOR="Blue"></FONT></I><I><FONT
COLOR="Blue"></FONT></I><IMG SRC="new6.gif"></FONT></H1>

<H1 ALIGN="center"><FONT SIZE="7"><I><FONT
COLOR="Blue">HP</FONT></I></FONT></H1>
<FONT COLOR="Blue"><FONT COLOR="Blue"><FONT COLOR="Blue"><FONT
```

Podemos notar que o esquema de codificação no HTML e no SGML são iguais, com a característica básica dos *start-tags* e *end-tag.* Veremos mais adiante as características desse tipo de linguagem, também conhecida como linguagem *markup*.

2.6.3 ODA – *Open/Office Document Architecture*

ODA é um padrão internacional (ISO 8613) definido pela *International Standards Organisation (ISO)* para documentos de escritório [ISO91]. ODA pretende incluir as características convencionais e avançadas dos processadores de texto. Ele almeja definir a estrutura lógica dos documentos, incluindo apresentação de *layout* e *links.* ODA supri as necessidades dos ambientes de escritório (por este motivo o nome original é *Office Document Architecture*). Na literatura, ODA é geralmente referenciado como sendo um padrão superior ao SGML, porém os produtos que usam o padrão ODA ainda são muito limitados em uma escala global.

2.6.4 TeX

Este padrão é altamente aplicado na comunidade científica. TeX foi desenvolvido como uma linguagem de formatação para imprimir tabelas e fórmulas complexas. Documentos escrito em TeX podem ser transferidos com sucesso entre várias plataformas. Um produto baseado em TeX é o LaTeX, o qual também oferece características para estruturação lógica. TeX pode atuar como uma solução “back-end” para sistemas de formatação de textos baseados em SGML.

2.6.5 RTF – *Rich Text Format*

Desenvolvido pela Microsoft, RTF usa as funções de mais baixo nível que são usadas pelo Microsoft Word. RTF é um formato apropriado para transferência sobre a Internet de arquivos do tipo ASCII.

Informações sobre o *layout* da página e as fontes usadas são armazenadas nas informações de cabeçalho de cada documento. RTF é um dos formatos mais suportados para permuta de documentos entre pacotes de processamento de texto, apesar de ter sido originado para permitir que arquivos fossem movidos de uma plataforma para outra.

Abaixo mostramos um texto codificado em RTF.

```
{\rtf1\ansi\ansicpg1252\uc1\deff1\deflang1033\deflangfe1046{\fonttbl{\
f0\froman\fcharset0\fprq2{\*\panose 020206030504050203 04}Times New
Roman;}{\f1\fswiss\fcharset0\fprq2{\*\panose
020b0604020202020204}Arial;}}
{\stylesheet{\nowidctlpar\widctlpar\adjustright
\f1\fs22\lang1046\cgrid \snext0 Normal;}{\*\cs10 \additive Default
Paragraph Font;}}{\info{\title 009}
{\author Luiz Eduardo}{\operator Luiz Eduardo}
{\creatim\yr1997\mo7\dy24\hr9\min24}{\revtim\yr1997\mo7\dy24\hr9\min27
}{\version2}{\edmins0}{\nofpages1}{\nofwords0}{\nofchars0}{\*\company
Luiz E S Oliveira}{\nofcharsws0}{\vern73}}
{\*\pnseclvl1\pnucrm\pnstart1\pnindent720\pnhang{\pntxta
.}}{\*\pnseclvl2\pnucltr\pnstart1\pnindent720\pnhang{\pntxta
.}}{\*\pnseclvl3\pndec\pnstart1\pnindent720\pnhang{\pntxta .}}
{\*\pnseclvl4\pnlcltr\pnstart1\pnindent720\pnhang{\pntxta
)}}{\*\pnseclvl5\pndec\pnstart1\pnindent720\pnhang{\pntxtb (}{\pntxta
)}}{\*\pnseclvl6\pnlcltr\pnstart1\pnindent720\pnhang{\pntxtb (}{\pntxta
)}}{\*\pnseclvl7\pnlcrm\pnstart1\pnindent720\pnhang
{\pntxtb (}{\pntxta
)}}{\*\pnseclvl8\pnlcltr\pnstart1\pnindent720\pnhang{\pntxtb
(}{\pntxta )}}{\*\pnseclvl9\pnlcrm\pnstart1\pnindent720\pnhang{\pntxtb
(}{\pntxta )}}\pard\plain \nowidctlpar\widctlpar\adjustright
\f1\fs22\lang1046\cgrid {
Teste de formato RTF
\par 009
\par 001
\par 3
\par 2
\par }}
```

2.6.6 PDF – *Portable Document Format*

PDF é um formato desenvolvido pela Adobe. Como o HTML, PDF permite que documentos sejam transmitidos através da rede. Existem planos para que este formato permita a inclusão de áudio e vídeo no documentos. Poucas plataformas possuem PDF *viewers* disponíveis. Atualmente, a SUN, Windows e Macintosh possuem.

2.7 Ferramentas

Nas seguintes páginas apresentamos resumidamente várias ferramentas disponíveis capazes de manipular formatos diferentes de documentos. Mais detalhes sobre essas ferramentas e análise crítica das mesmas podem ser encontradas em [GAR96].

2.7.1 Conversores de Formato

Ferramenta	Versão	Descrição
RTFtoHTML	2.7.5	Converte documentos RTF em formato HTML. Essa ferramenta usa regras de transação definidas em um arquivo chamado htmltrans. Isso significa que essas regras podem ser alteradas para produzir uma tradução mais correta para cada tipo de documento.
RTFtoWEB	1.5	É uma extensão do RTFtoHTML.
RTF2LaTex	1.5	Converte arquivos RTF para LaTeX.
wp2x	2.3	Pode converter arquivo WordPerfect (versão 5.1) para diferentes formatos entre eles: LaTeX, SGML e HTML.
wp2LaTeX		Converte arquivos WordPerfect para macros LaTeX desde que certas regras sejam seguida na escrita do arquivo WordPerfect.
LaTex2RTF	1.1	Converte arquivos LaTeX para formato RTF.
DTD2HTML	1.3.0	Permite ao usuário visualizar um DTD (*Document Type Definition)* de uma maneira gráfica usando um *browser* qualquer.
SGML2LaTeX	1.3	Converte arquivos SGML para LaTeX. Esse pacote vem com DTD´s (*Document Type Definitions*) pré-definidos para documentos LaTeX.
HTML2LaTeX		Converte documentos HTML para LaTeX. Possui

		documentação nos formatos HTML e Postscript.
html2ps	beta	Converte arquivos HTML em formato Postscript.
ps2html	2.0	Converte arquivos Postscript para HTML.

2.7.2 Compiladores

Ferramenta	Versão	Descrição
sgmls	1.1	sgmls é um compilador que valida documentos SGML. Após a execução, a representação dos elementos da estrutura é enviado para uma saída padrão.
vm2		vm2 também valida documentos SGML. Ele fornece a descrição de todos os componentes da estrutura como saída. Somente é executado em plataforma DOS.
arcSGML	1.0	arcSGML é uma conjunto de ferramentas usado para desenvolvimento de aplicações em SGML. Originalmente escrito para rodar em DOS.

2.7.3 *Viewers*

Ferramenta	Versão	Descrição
xdvi	18f	xdvi é um utilitário para visualização de arquivos em DVI (*DeVice Independent).*
GhostScript	2.6.2	Usado para visualizar e imprimir arquivos PostScript.
GhostView	1.5	GhostView é um *front end* para o GhostScript. Disponível para Windows e Unix.
Netscape		Disponíveis para várias plataformas. Seu propósito primário é visualizar documentos HTML sobre a Internet.
Lynx	2.4	Lynx é um *browser* WWW baseado em caracter para plataformas Unix e VMS.
Panorama	Pro	Panorama é um pacote comercial para visualizar documentos SGML. É fornecido pela SoftQuad.

2.8 Neste Capítulo

Neste capítulo apresentamos o estado atual das técnicas de análise de imagem de documento, bem como as principais abordagens dentro dessa área, que são *Bottom-Up* e *Top-Down.*

Vimos um breve estudo sobre Processamento Digital de Imagens, onde mostramos as operações básicas que nos darão sustentação para as implementações necessárias futuramente.

Por último, apresentamos os principais padrões de documentação e ferramentas que nos permitem armazenar as estruturas lógica e física de um documento. Como foi descrito acima, existe uma forte tendência para o lado do SGML.

No capítulo seguinte apresentaremos a estrutura do cheque bancário brasileiro, bem como a legislação vigente do Banco Central do Brasil para cheques bancários.

Capítulo III
Estrutura do Cheque Bancário Brasileiro

3.1 Introdução

Neste capítulo veremos a estrutura dos cheques bancários, ou seja, as informações contidas nos cheques, suas funções e posições, e também a legislação vigente do Banco Central do Brasil para cheques bancários.

O cheque bancário é um instrumento que os bancos colocam a disposição dos clientes para movimentar os fundos em seu poder e efetuar pagamentos sem a necessidade do manuseio do papel moeda. Um cheque representa uma ordem incondicional em direito que: é emitida por uma pessoa em favor de outra; é assinado pela pessoa que emite; e exige que o banco pague na apresentação uma determinada quantia em dinheiro.

Veremos agora a estrutura dos cheques bancários brasileiros, suas características em relação ao conteúdo e a disposição dos seus elementos componentes. O Conselho Monetário Nacional instituiu um modelo único de cheque visando uma maior segurança e eficiência nos serviços que os bancos prestam a coletividade.

<table>
<tr><td colspan="2">Número da Compensação, Número do Banco, Número da Agência, Número da Conta, Número do Cheque, e Digitos Verificadores</td><td>Valor Númerico</td></tr>
<tr><td colspan="3">Valor por Extenso</td></tr>
<tr><td colspan="3">Nome do Favorecido</td></tr>
<tr><td colspan="2">Logotipo da Banco</td><td rowspan="2">Local e Data de Emissão</td></tr>
<tr><td>Nome do Banco (Nome, Número, Endereço e Telefone da Agência)</td><td colspan="2">Assinatura, Nome do Cliente, Número do CPF/CGC e outras Informações</td></tr>
<tr><td colspan="3">Linha MICR CMC-7</td></tr>
</table>

Figura 3.1: Modelo para a distribuição de informações em cheques bancários brasileiros.

Na Figura 3.1 é mostrada a localização dos elementos dentro da área de um cheque bancário. Na Figura 3.2 é apresentado um modelo padrão de preenchimento dos campos e áreas do cheque segundo o padrão estabelecido pelas normas do Banco Central.

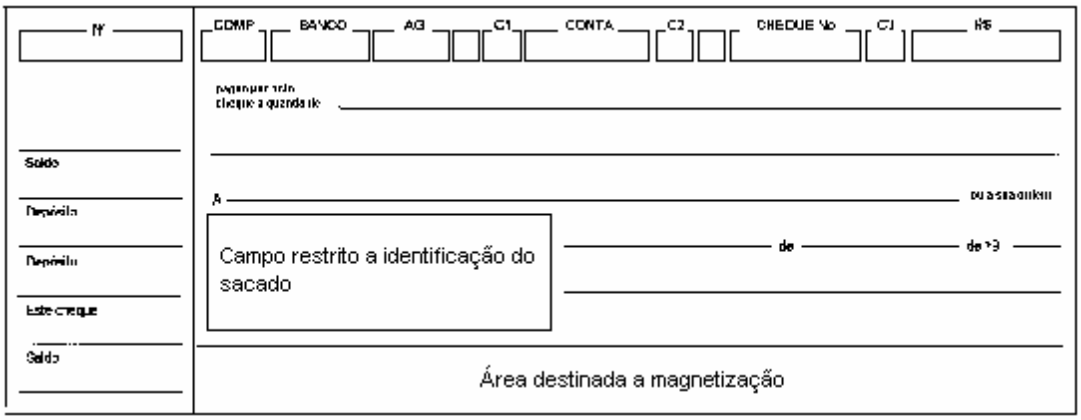

Figura 3.2: Modelo padrão de preenchimento dos campos e áreas de um cheque.

A seguir descreveremos os elementos que compõe a estrutura dos cheques bancários brasileiros, e que são relevantes para este trabalho. Esta descrição é baseada na resolução número 885 editada pelo Banco Central [BCO83].

3.2 Dimensões do Cheque

Quanto as dimensões dos cheques bancários brasileiros, o Banco Central estabelece que os cheques devem ter o comprimento de 175 mm com tolerância de mais ou menos 1 mm, e largura de 80 mm, com tolerância de 1 mm para mais e de até 4 mm para menos. Assim, um cheque bancário digitalizado em 300 dpi podem assumir uma dimensão de no mínimo 2021 *pixels* de comprimento 886 *pixels* de largura e no máximo de 2032 *pixels* de comprimento e 910 *pixels* de largura, como é mostrado na Figura 3.3.

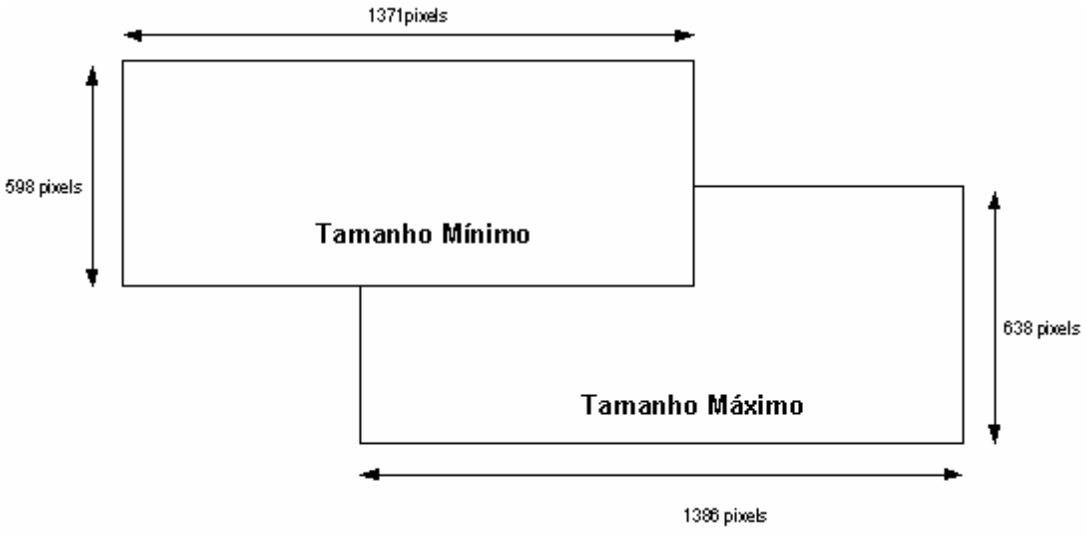

Figura 3.3: Dimensões mínimas e máximas para um cheque bancário.

3.3 Elementos Componentes do Cheque Bancário Brasileiro

Os cheques podem ser divididos em quatro faixas, como é mostrado na Figura 3.4.

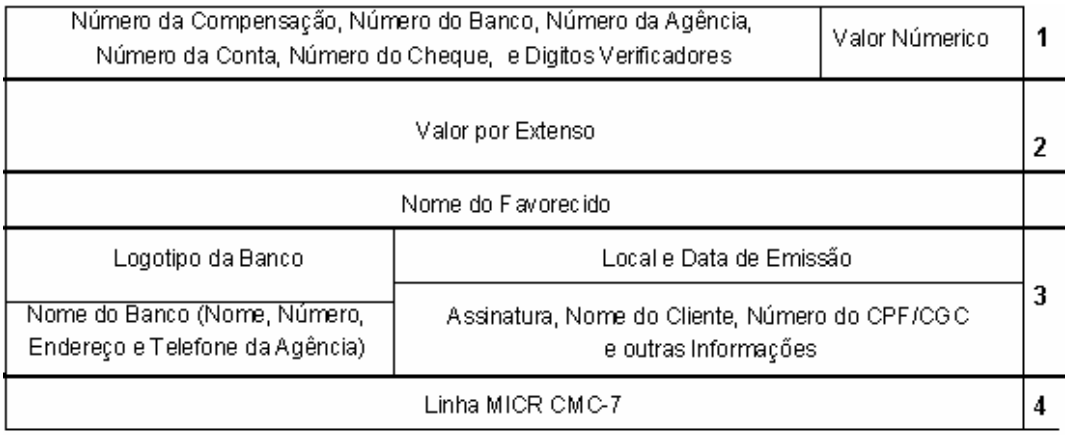

Figura 3.4: Um cheque bancário dividido em quatro faixas.

Na Tabela 3.1, descrevemos os elementos que são encontrados na primeira faixa do cheque.

Tabela 3.1: Composição da Faixa número 1 do cheque bancário.

Elemento	Descrição
COMP	Número da Câmara de Compensação a que está jurisdicionada a agência sacada, composto de 3 caracteres numéricos.
BANCO	Número de inscrição do Banco no Serviço de Compensação atribuído pelo Banco Central do Brasil, composto de 3 caracteres numéricos.
AG	Código da agência sacada, representado pelas 4 posições do número de ordem de inscrição no Cadastro Geral de Contribuintes (CGC).
( )	Posição de utilização facultativa para indicação de dígito verificador correspondente ao código da agência sacada.
C1	Dígito verificador correspondente aos campos BANCO, COMP e AG, calculado com peso de 2 a 9, módulo 11 a 0 (zero) no resto 10.
CONTA	Número da conta do emitente, podendo constar do mesmo a razão contábil ou ainda dígitos de auto-conferência. Compõe-se de, no máximo, 10 caracteres numéricos.
C2	Dígito verificador relativo ao número da conta calculado de forma idêntica ao C1.
( )	Posições de utilização optativa para registro da série de numeração do cheque.
CHEQUE N.	Número do cheque composto por 6 caracteres numéricos.
C3	Dígito verificador do número do cheque calculado de forma idêntica ao C1.
R$	Espaço destinado à especificação do valor do cheque em algarismos.

A segunda faixa, destinada à indicação do valor do cheque por extenso e nome do beneficiário, compreende: a expressão “Pague por este cheque a quantia de”, seguida de duas linha reservadas para a especificação do valor do cheque por

extenso, permitindo somente a expressão “e centavos acima” no final da segunda linha; uma terceira linha, iniciada com a preposição “a” e terminada com a expressão “ou à sua ordem” ou se for o caso, “não à ordem”, reservada para indicação do nome do beneficiário do cheque.

A terceira faixa destinada à identificação do banco e da agência sacados, local e data de emissão do cheque, assinatura e identificação do emitente, deve conter, à esquerda, na área assinalada na Figura 3.2, sendo impressos em primeiro plano, o nome do banco sacado, facultando-se que seja o mesmo precedido do logotipo da instituição; em segundo plano, deve ser identificada a agência sacada e seu endereço completo (logradouro, número, localidade e Unidade Federativa). À direita devem ser impressas as linhas reservadas à indicação do local e data de emissão do cheque e a assinatura do emitente.

Abaixo da linha de assinatura deverão constar o nome do correntista e o respectivo número do Cadastro de Pessoas Físicas (CPF) ou do Cadastro Geral de Contribuintes CGC. A exigência de registro, no cheque, do número do CPF ou do CGC do correntista somente prevalece para os documentos da espécie emitidos por pessoas físicas ou jurídicas que, na forma da lei, sejam ou venham a ser alcançadas pela obrigatoriedade de inscrição no CPF ou no CGC. Não são, portanto atingidos pela mencionada exigência os seguintes casos, em cujos cheques devem ser inscrita a palavra “ISENTO” no espaço reservado para o CPF ou CGC [BCO83] :

- Contas correntes tituladas por aqueles que por força de convênios ou acordos formalmente reconhecidos pelas autoridades brasileiras, estão isentos de tributação sobre os respectivos rendimentos ou recursos;

- Contas bancárias de uso exclusivo de embaixadas, consulados estrangeiros e representações de organismos internacionais, bem como as de uso pessoal de diplomatas, cônsules e funcionários administrativos estrangeiros, membros de missões diplomáticas e repartições consulares;

- Contas bancárias de funcionários não brasileiros de escritório de entidades internacionais, ou de peritos e técnicos que se encontrem no País, no âmbito de programas específicos de cooperação;

Também são atingidos pela exigência de registro de CPF ou CGC, devendo, igualmente, ser inscrita a palavra "ISENTO" no espaço reservado para aquele fim, os cheques em pagamento total ou parcial do passivo de instituição ou sociedade sob intervenção ou em liquidação extrajudicial, quando emitidos por prepostos do Banco Central do Brasil.

A quarta faixa é destinada a impressão de caracteres magnéticos, que também é conhecida como linha MICR. A linha MICR está posicionada em uma área denominada de "faixa limpa" localizada na parte inferior do cheque, bem como deve atender uma série de requisitos em relação ao seu posicionamento. A seguir apresentamos as principais regras que devem ser efetivamente seguidas:

- A "faixa de magnetização" ocupa, em toda a extensão horizontal do cheque, o espaço com 16 mm de altura a partir da base do formulário reservado exclusivamente para a impressão dos caracteres magnéticos;

- Centrada no interior da "faixa de magnetização", a 4,80 mm da base do formulário e a 6 mm da margem direita, deve ser delimitada a "banda de magnetização" com 161,95 mm de comprimento e 6,40 mm de altura sobre a qual atua o cabeçote de leitura de caracteres magnéticos;

- O eixo horizontal de simetria dos caracteres deve coincidir com o da faixa de magnetização, posicionando-se a 8,00 mm da base do formulário com tolerância de deslocamento vertical de no máximo 1,60 mm;

- O caracter padrão para magnetização de cheques bancários e outros documentos bancários é o Sistema de Caracteres Magnéticos Codificados em Sete Barras (CMC-7), podendo ser adotado qualquer tipo padronizado pela *European Computer Manufacturers Association* (ECMA), desde que observada a amplitude de 8

caracteres no espaçamento de 25,4 mm no sentido horizontal de modo que cada caractere com respectivo intervalo ocupe uma largura de 3,175 mm;

- A “banda de magnetização” comporta os quatro campos seguintes:

  Campo 1: Distante 1,6 mm da margem esquerda da banda de magnetização, ocupa horizontalmente uma extensão de 31,75 mm correspondendo a 10 posições de caracteres magnéticos;

  Campo 2: Ocupa, imediatamente à direita do campo 1, uma extensão horizontal de 38,10 mm correspondente a um espaço em branco de 3,175 mm e 11 posições de caracteres magnéticos;

  Campo 3: Ocupa imediatamente à direita do campo 2, uma extensão horizontal de 44,45 mm, correspondente a um espaço em branco, de 3,175 mm, e 13 posições de caracteres magnéticos;

  Campo 4: Ocupa, à direita do campo 3, uma extensão horizontal de 44,45 mm, correspondente a um espaço em branco de 3,175 mm e 13 posições de caracteres magnéticos distanciado 1,6 mm da margem direita da “banda de magnetização”;

Sendo assim, a linha MICR contém 50 posições numeradas da direita para a esquerda e agrupadas em quatro campos: valor, informações internas, compensação e informações auxiliares. A seguir descrevemos o conteúdo de cada um dos quatro campos da banda de magnetização.

- Campo 1: um símbolo S3, três caracteres correspondentes ao código de inscrição do banco no Serviço de Compensação, quatro caracteres correspondentes ao código da agência; um caractere correspondente ao dígito verificador (C2) dos elementos componentes do campo 2 (código da câmara de compensação, número do cheque e código de tipificação do documento), calculado através do módulo 10 e por último, novamente um símbolo S3.

- Campo 2: um espaço em branco, três caracteres correspondentes ao código da câmara de compensação a que esteja jurisdicionada a agência sacada, seis caracteres correspondentes ao números do cheque, um caractere correspondente ao código de tipificação do documento para cujo propósito está definida a codificação dada na Tabela 3.2, e por último temos um símbolo S5.

Tabela 3.2: Códigos de Tipificação.

Documento	Código
Cheque Comum	5
Recibo ou Ordem de Pagamento	6
Cheque de Viagem	7
Cheque Bancário	8

- Campo 3: um espaço em branco, um caractere correspondente ao dígito verificador (C1) do código do banco e do código da agência inscritos no campo 1, calculado através do módulo 10, dez caracteres correspondentes ao número da conta do correntista, podendo constar do mesmo a razão contábil e o dígito de auto-conferência da conta, um caracter correspondente ao dígito verificador (C3) das dez posições referentes ao número da conta do correntista calculado através do módulo 10 e por último um símbolo S1.

- Campo 4: este campo é reservado para a pós marcação do valor do cheque e é composto por um espaço em branco, doze caracteres representativos do valor do cheque, preenchendo-se com zeros ou deixando-se em branco as posições à esquerda não demandadas para exprimir a quantia e por último um símbolo S2.

Na Figura 3.5 apresentamos a descrição dos elementos de uma linha MICR.

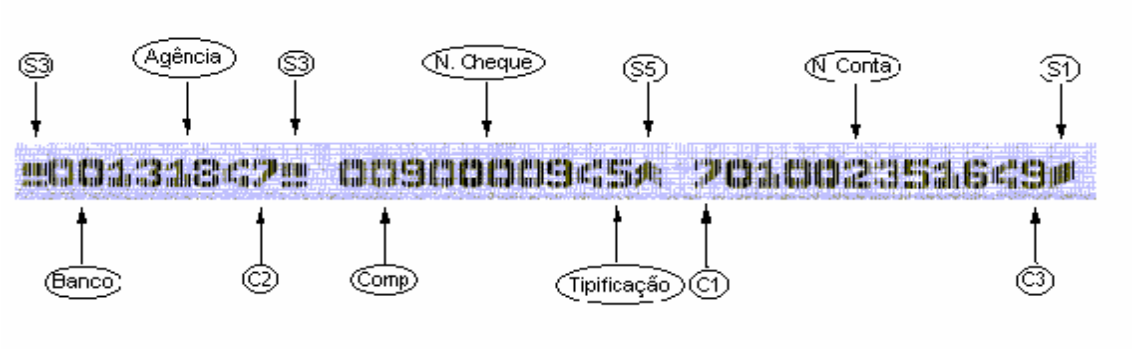

Figura 3.5: Descrição dos elementos de uma linha MICR.

3.4 Impressão e Confecção do Formulário dos Cheques

Os formulários dos cheques devem ser impressos em qualquer cor, com tinta indelével e de tonalidade firme, sendo permitido o uso de impressões por perfuração ou filigrama, desde que não apostas na faixa de magnetização e não produzam mossas ou rebarbas que prejudiquem a leitura mecânica do cheque. Os cheques podem ser confeccionados em folhas planas ou em formulários contínuos, devendo, nesse último caso, serem destacados por guilhotina de modo que obtenham bordas retas e lisas nas extremidades inferior e direita, utilizadas como referências no processo de leitura mecânica.

Os cheques bancários brasileiros possuem uma diversidade de cores, desenhos e estilos. Eles apresentam uma estrutura de *layout* complexa caracterizada pela presença de desenhos e Figuras ilustrativas que fazem parte do padrão de fundo. O padrão de fundo pode assumir várias cores e padrões, porém, os elementos que são impressos sobre o fundo, como linhas e seqüência de caracteres, apresentam um posicionamento ordenado, sendo impressos em uma só cor, geralmente, em preto [KOE97].

3.5 O Formulário Cheque

Podemos considerar o cheque como sendo um formulário pré-impresso, isto é, ele possui algumas informações previamente impressas e campos em branco onde devemos introduzir outras informações. De uma maneira geral podemos considerar que o formulário de um cheque é constituído por três elementos básicos: padrão de fundo, fontes e linhas. O padrão de fundo representa o próprio papel sobre o qual as demais informações serão impressas. As fontes representam a forma pela qual as seqüências de caracteres serão impressas. Por último, as linhas representam a forma e as dimensões das linhas presentes nos cheques.

3.5.1 Fontes

Este nome é dado a um conjunto de caracteres que possui as mesmas características ou propriedades. A letra A, por exemplo, pode ser representada de inúmeras maneiras, além disso, um caractere pode assumir diversos tamanhos, cores e estilos, etc. As normas do Banco Central do Brasil não estabelecem o tipo de fonte a ser usado na confecção dos formulários dos cheques bancários, assim, as instituições financeiras usam livremente vários tipos de fontes. Na Figura 3.6 mostramos algumas das fontes usadas para a confecção dos formulários de cheques.

Figura 3.6: Diferentes tipos de fontes usados nos formulários de cheques.

3.5.2 Padrão de Fundo

O padrão de fundo de um cheque está relacionado com as características do papel sobre o qual as informações são impressas.

O Banco Central em sua norma [BCO83], estabelece as seguintes características para o papel:

a) Peso por m² deve ser igual a 90g, com tolerância para mais ou menos de até 5%;
b) Espessura de 0,1 a 0,127 mm;
c) Rigidez de 3 a 4,5 em direção da máquina;
d) Superfície do tipo “sheffield” 72 – 125 – 10 cm³/mm; “bekk” 50 – 120 segundos;
e) Rasgado ou rutura (Elmendorff); mínimo de 40 gramas em ambas as direções;
f) Porosidade (Gurley): Mínimo 25 segundos para 100 cm³ de ar; máximo 200 segundos para 100 cm³ de ar;
g) Todas as análises devem ser efetuadas com umidade relativa de 50% e a uma temperatura de 20°;
h) O documento deve resistir a uma prova de resistência em cera igual a Dennilson 16 A;
i) Os fabricantes de papel e as gráficas devem cuidar para que haja quantidade;
j) Mínima de partículas magnetizáveis na composição do papel (ferro, etc.);
k) O cheque admite, no máximo, 30% de corante diluído em branco;
l) O cheque deve conservar os necessários requisitos de segurança, tais como papel indelével e fundo artístico.

3.5.3 Linhas

As linhas horizontais e verticais podem assumir diferentes intensidades e tamanhos, porém, impressas na cor preta. Os cheques devem possuir obrigatoriamente linhas de base que servem de guia para o preenchimento das informações manuscritas. Além disso, os cheques podem conter algumas caixas retangulares, geradas a partir da interseção de linhas horizontais e verticais, que limitam principalmente a região dedicada ao valor numérico.

3.6 Neste Capítulo

Vimos neste capítulo a estrutura dos cheques bancários, e a legislação vigente do Banco Central do Brasil para cheques bancários O conhecimento dessa estrutura será muito útil no momento da análise da imagem do documento, como veremos mais adiante nesse trabalho.

No capítulo seguinte será apresentado um estudo sobre os dois principais padrões de documentos no mercado, o SGML e o ODA.

Capítulo IV
Padrões de Documentos

4.1 Introdução

Este capítulo apresenta um estudo teórico sobre os dois principias padrões no mercado, o SGML (*Standart Generalized Markup Language*) e o ODA (*Open/Office Document Architecture).* Esses padrões são capazes de armazenar a estruturas lógica e física de um documento. No final do capítulo apresentamos um comparativo entre esses dois padrões.

4.2 SGML (*Standart Generalized Markup Language*)

SGML é um padrão internacional (ISO 8879) para representação de textos no formato eletrônico independentemente de dispositivos de *hardware* e de *software*. SGML é uma metalinguagem, ou seja, um meio de descrever uma linguagem formalmente, e neste caso uma linguagem *markup* [McQ97].

Historicamente, a palavra *markup* tem sido usada para descrever anotações ou outras marcas em um texto principalmente para instruir o digitador a maneira como um determinado texto deve ser impresso.

Generalizando esta idéia, nós definimos *markup* o meio de realizar uma interpretação explícita do texto. De um modo geral, todos os textos impressos são codificados com o objetivo de: marcas de pontuação, disposição das letras em torno da página, mesmo os espaços entre as palavras, podem ser considerados uma classe de *markup,* pois ajudam o leitor humano a identificar onde começa e acaba uma palavra.

Por linguagem *markup* entendemos como um conjunto de marcadores usados para codificar um texto. Uma linguagem *markup* deve especificar quais os marcadores permitidos, quando eles são requeridos, como esses marcadores devem

ser distinguidos do texto, e o que esses marcadores significam. O SGML provê os três itens acima relacionados.

4.2.1 Características do SGML

Segundo [McQ97] existem três características no SGML que a distinguem de outras linguagens *markup*: sua ênfase mais descritiva do que procedural; o conceito de tipo de documento; e a sua independência de qualquer sistema para representação do *script* no qual o texto é escrito.

4.2.1.1 *Markup* Descritivo

Um sistema de *markup* descritivo utiliza marcas que fornecem nomes para categorizar partes do documento. Códigos *markup* tais como <para> ou \end{list} identificam porções de um documento e afirmam que “o item seguinte é um parágrafo”, ou “é o fim de uma lista recentemente começada”. Por outro lado, um sistema de *markup* procedural define que em determinados pontos do documento, o procedimento PARÁGRAFO com alguns parâmetros deve ser chamado.

Com *markup* descritivo em vez de procedural o mesmo pedaço do texto pode ser processado por vários *softwares* diferentes, cada um aplicando diferentes instruções nas partes que acham relevantes. Por exemplo, um processo pode desprezar totalmente os rodapés existentes no texto, enquanto que outro processo que está sendo executado paralelamente pode buscar somente os rodapés para imprimí-los no final do texto. Diferentes processos de ordenação podem ser aplicados sobre a mesma parte do texto, cada um obtendo um resultado diferente.

4.2.1.2 Tipos de Documento

O SGML introduz a noção de tipo de documento, e consequentemente a definição de tipo de documento (*Document Type Definition - DTD*). Documentos são considerados como tipos. O tipo do documento é formalmente definido pelas suas partes constituintes e por sua estrutura [McQ97]. A definição de um artigo, por

exemplo, pode ser que seja constituída de um título e possivelmente de um autor, seguido de um resumo (*abstract*) e uma seqüência de um ou mais parágrafos. Qualquer documento desprovido de título, de acordo com sua definição formal, não deve ser um artigo.

Se documentos são tipos conhecidos, um programa com propósito especial (chamado analisador) pode ser usado para processar documentos alegando ser um tipo particular e verificando que todos os elementos requeridos para o tipo de documento estão de fato presentes e corretamente ordenados. Mais especificamente, documentos diferentes de um mesmo tipo podem ser processados de uma maneira uniforme. Programas podem ser escritos com a vantagem de ter conhecimento encapsulado sobre a informação da estrutura do documento, e deste modo podendo comportar-se de um modo mais inteligente.

4.2.1.3 Independência de Dados

A meta básica do SGML é garantir que os documentos codificados possam ser transportados de um ambiente de *hardware* e *software* para outro sem perder informações. As duas características discutidas até então, tratam de requerimentos em um nível abstrato. Esta característica trata de um nível de *string* de *bytes* (caracteres) dos quais dos documentos são compostos. SGML prove um mecanismo para substituição de *strings* que é um caminho simples e independente de plataforma para a troca de *strings* enquanto o texto é processado. Uma aplicação óbvia para esse mecanismo é garantir a consistência de nomenclatura. Esse mecanismo leva o nome de *entidade* e será discutido mais adiante.

4.2.2 Estrutura SGML

Nesta seção descrevemos o simples e consistente mecanismo de identificação da estrutura de unidades de texto que são providas pelo SGML.

4.2.2.1 Elementos

O termo técnico usado no padrão SGML para uma unidade de texto, visto como um componente estrutural, é *elemento.*

Em um texto *markup* todos os elementos devem ser rotulados de alguma maneira. O padrão SGML provê várias maneiras para realizar isso, mas a mais utilizada é inserir uma marca no início do elemento (*start-tag*) e outra no fim (*end-tag*).

O par *start-tag* e *end-tag* são usados para fechar a ocorrência de um determinado elemento dentro de um texto. Por exemplo: <título>Dissertação de Mestrado</título>. Como pode-se observar neste exemplo o *start-tag* tem a forma **<título>**, onde seu início se dá com o sinal de menor e o fim com o sinal de maior, e nesse espaço está o identificador do elemento, também conhecido como identificador genérico (*generic identifier - GI*). De maneira análoga, acontece com o *end-tag*, exceto que o sinal de menor é seguido de uma barra (/).

4.2.2.2 Um Exemplo

Um elemento pode ser vazio, isto é, ele pode estar sem conteúdo, ou pode conter um texto simples. Porém o mais comum são os elementos embutidos dentro de elementos de tipos diferentes.

Para ilustrar isso, vamos considerar um modelo estrutural simples. Optamos por mostrar um exemplo, para melhor colocar a sintaxe e as características do SGML. Para construção deste exemplo, baseamo-nos em [McQ97].

Vamos assumir que queremos identificar em uma antologia somente os poemas, seus títulos e suas estrofes e as linhas que compõe as estrofes. Em SGML, nosso tipo de documento é *antologia,* e é composta de uma série de *poemas*. Cada poema possui um *título* e varias ocorrências de *estrofes*, que por sua vez possui várias *linhas*. Mostramos um texto com todos os *tags* incluídos.

```
<antologia>
```

```
   <poema>
      <titulo>NONO</titulo>
         <estrofe>
          <linha>nononononononononon</line>
          <linha>nononononononononon</line>
          <linha>nononononononononon</line>
          <linha>nononononononononon</line>
         </estrofe>
         <estrofe>
          <linha>nononononononononon</line>
          <linha>nononononononononon</line>
          <linha>nononononononononon</line>
          <linha>nononononononononon</line>
         </estrofe>
   </poema>

         <!-- mais poemas -->

</antologia>
```

Em SGML uma linha de comentário começa com <!-- e acaba com -->.

Esse exemplo não estabelece nenhum tipo de regra, como por exemplo, um título não pode aparecer em outro lugar a não ser antes de uma estrofe, as linhas podem aparecer fora das estrofes. Nesses casos, o início e o fim de cada elemento devem ser explicitamente marcados, pois não existem regras definindo onde cada elemento pode ou não aparecer. Na prática, as regras são formuladas para diminuir a necessidade de muitos *tags*. Por exemplo, considerando nosso modelo, podemos definir as seguintes regras:

- Uma antologia contém um número de poemas e nada mais.
- Um poema sempre tem um título antes de uma estrofe.
- Separadamente do título, um poema é composto somente de estrofes.
- Estrofes são compostas somente de linhas e as linhas sempre estão dentro das estrofes.

- Nada pode seguir uma estrofe a não ser outra estrofe.
- Nada pode seguir uma linha a não ser outra linha ou o começo de uma nova estrofe.

Com base nessas regras, podemos inferir que não precisamos mais dos marcadores de fim de linha e de fim de estrofes, bem como o título não precisa de um marcador final, pois está implícito no começo da estrofe. Também podemos deduzir que o poema não precisa de um marcador final, desde que definimos que o fim de um poema está implícito no começo de um novo poema ou no fim da antologia. Aplicando essas simplificações temos um novo texto como segue:

```
<antologia>
   <poema>
      <titulo>NONONONONONO
         <estrofe>
          <linha>nononononononononon
          <linha>nononononononononon
          <linha>nononononononononon
          <linha>nononononononononon
         <estrofe>
          <linha>nononononononononon
          <linha>nononononononononon
          <linha>nononononononononon
          <linha>nononononononononon

   <poema>
         <!-- mais poemas -->

</antologia>
```

A possibilidade do uso de regras é uma característica muito importante em SGML. Mesmo não levando em consideração essas regras, podemos processar o texto *markup* para os mais diferentes propósitos. Um simples programa pode extrair somente os títulos dos poemas, ou somente as palavras usadas no poema para a construção de uma lista. Um outro programa poderia inserir uma linha em branco entre

o título e a primeira linha da estrofe. E é claro, o texto apresentado acima pode ser transportado de um computador para outro e do mesmo modo ser processado.

4.2.3 Definindo Estruturas de Documento: O *DTD (*D*ocument Type Definition)*

Regras tais como as que definimos anteriormente são o primeiro estágio na criação de uma especificação formal da estrutura num documento SGML, ou D*ocument Type Definition,* usualmente abreviado como *DTD*.

4.2.3.1 Um Exemplo de *DTD*

Um *DTD* é expressado em SGML como um conjunto de declarações, usando uma sintaxe simples definida no padrão [TRA95]. Para o modelo do poema, as declarações apropriadas seriam as seguintes:

```
<!ELEMENT antologia        - -   (poema+)>
<!ELEMENT poema            - -   (titulo?, estrofe+)>
<!ELEMENT titulo           - O   (#PCDATA)>
<!ELEMENT estrofe          - O   (linha+)>
<!ELEMENT linha            O O   (#PCDATA)>
```

Essas cinco linhas são exemplo da declaração formal dos elementos em SGML. A declaração é delimitada pelos sinais de menor e maior; o primeiro caracter após o sinal de menor é o sinal de exclamação seguido imediatamente por uma das palavras definidas em um pequeno conjunto do SGML, a qual especifica a classe do elemento declarado. As cinco declarações acima são todas do mesmo tipo: começam com a palavra ELEMENT, indicando que a declaração é um elemento, no sentido técnico definido acima. Cada linha consiste de três partes: um nome ou um grupo de nomes, dois caracteres que especificam as regras de minimização, e o conteúdo do modelo. Cada uma dessas partes serão discutidas futuramente. Os componentes da declaração são separados por espaços brancos, que podem ser um ou mais brancos, tabulações ou novas linhas.

A primeira parte de cada declaração acima é o identificador genérico do elemento que está sendo declarado, por exemplo “poema”, “título”, etc. É possível declarar vários elementos em uma mesma declaração, como discutiremos mais adiante.

4.2.3.2 Regras de Minimização

A segunda parte da declaração especifica o que chamamos de regras de minimização para o elemento em questão. Essas regras determinam tanto se o *start-tag* quanto o *end-tag* devem estar presente no elemento em questão. A regra se encontra na forma de um par de caracteres, separados por um espaço em branco, onde o primeiro caracter indica a presença do *start-tag* e o segundo caracter indica a presença do *end-tag*. Nesses casos tanto o hífen quanto a letra “O” (para “omisso” ou opcional) podem aparecer. O hífen indica que o *start-tag* deve estar presente, e a letra O que o mesmo pode não estar presente. Deste modo, no nosso exemplo, todos os elemento exceto **<linha>** devem ter o *start-tag*. Somente os elementos **<poema>** e **<antologia>** devem ter o *end-tag.*

4.2.3.3 Conteúdo do Modelo

A terceira parte de cada declaração, entre parênteses, é chamada de conteúdo do modelo do elemento, porque ela especifica que ocorrências o elemento pode realmente conter. Os conteúdos são especificados em termos de outros elementos ou usando palavras reservadas. Existem várias palavras reservadas, e entre elas a mais freqüentemente encontrada é PCDATA, como no nosso exemplo. PCDATA é a abreviação de *parsed character data,* e significa que o elemento que está sendo definido pode conter qualquer caracter válido para o SGML.

4.2.3.4 Indicadores de Ocorrência

A declaração para **<estrofe>** no exemplo acima indica que uma estrofe consiste de uma ou mais linhas. O uso do indicador de ocorrência (o sinal de mais) indica quantas vezes o elemento em seu conteúdo do modelo pode ocorrer. Existem três indicadores de ocorrências na sintaxe do SGML, convencionalmente representadas pelo sinal de mais (+), pelo sinal de interrogação (?) e pelo asterisco (*). O sinal de mais significa que existe uma ou mais que uma ocorrência do elemento em questão; o sinal de interrogação significa que pode ou não existir a ocorrência do elemento, e se existir ela ocorrerá somente uma vez; e o asterisco significa que o elemento pode não ocorrer, bem como pode ocorrer uma ou mais vezes. Deste modo, se o conteúdo de **<estrofe>** for (LINHA*), a estrofe poderá não ter linhas como também poderá ter uma ou mais linhas. Se o conteúdo de **<estrofe>** for (LINHA?), novamente a estrofe poderá não ter linhas, mas se houver linhas, não haverá mais que uma. A declaração de **<poema>** no exemplo acima mostra que pode existir um título ou nenhum, e também que deve existir pelo menos uma estrofe, porém várias são permitidas.

4.2.3.5 Grupo de Conectores

O conteúdo do modelo (TITULO?, ESTROFE+) contém mais que um componente, e deste modo é necessário especificar a ordem na qual esses componentes podem aparecer. Este ordenamento é determinado pelo grupo de conectores (a vírgula) usada entre os componentes. Existem três possíveis conectores, convencionalmente representados pela vírgula (,), a barra vertical (/) e o E comercial (&). A vírgula indica que os componentes conectados devem aparecer na ordem especificada pelo conteúdo do modelo. O E comercial indica que os componentes conectados devem aparecer e que podem aparecer em qualquer ordem. A barra vertical indica que somente um dos componentes conectados pode aparecer. Se a vírgula nesse exemplo for trocada pelo E comercial, o título pode aparecer tanto depois da estrofe do poema ou no fim da estrofe (mas nunca entre as estrofes). Se for substituído por uma barra vertical, então o poema deverá consistir se existe título ou estrofes, mas nunca ambos.

4.2.3.6 Modelos de Grupos

No nosso exemplo até agora, os elementos dos conteúdos de modelos tem sido elementos simples ou PCDATA. Isto é perfeitamente possível, porém pode-se definir conteúdo de modelo onde os componentes são listas de elementos, combinadas por grupo de conectores. Essas listas são conhecidas como grupos de modelos.

Para demonstrar essas facilidades, vamos expandir nosso exemplo para incluir tipos de versos variados. Por causa desta demonstração classificaremos os poemas como *couplets* ou *blank* . Um poema de versos *blank* consiste de linhas simples, consequentemente sem elementos adicionais para definí-lo. Um *couplet* é definido como uma **<linha>** seguido por outra **<linha>.**

```
<!ELEMENT couplet     O O   (linha1, linha2) >
```

Os elementos **<linha1>** e **<linha2>** tem exatamente o mesmo conteúdo de modelo do elemento **<linha>**. Por esta razão eles podem compartilhar a mesma declaração.

A declaração para o elemento **<poema>** pode agora ser alterada para incluir as três possibilidades:

```
<!ELEMENT poema - O (titulo?, (estrofe+ | couplet+ | linha+) ) >
```

Isto é, um poema consiste de um título opcional, seguido de uma ou mais estrofes, ou mais *couplets*, ou de várias linhas. Note a diferença entre esta definição e a seguinte.

```
<!ELEMENT poema - O (titulo?, (estrofe | couplet | linha)+  ) >
```

A segunda versão, pela aplicação do sinal para o grupo e não para cada elemento, permite então que um simples poema seja composto de uma mistura de estrofes, *couplets* ou versos brancos.

Modelos altamente complexos podem ser construídos desta maneira, unindo a complexidade estrutural de vários tipos de texto. Como no exemplo que segue, consideramos o caso de um verso misto no qual um refrão e/ou coro aparecem. O refrão pode ser composto da repetição de um elemento linha, ou ele pode ser um texto simples. Um refrão pode aparecer no começo do poema somente, ou como uma opção adicional de cada estrofe. Isso deveria ser expressado no conteúdo do modelo da seguinte maneira:

```
<!ELEMENT refrão     - -  (#PCDATA | linha+)>
<ELEMENT  poema      - O
          (titulo?, (linha+) | ( refrão?, (estrofe, refrão?)+) ))>
```

Ou seja, um poema consiste de um título opcional, seguido por uma seqüência de linhas, ou um poema sem nome, com um refrão opcional no início, seguido de uma ou mais ocorrências de cada grupo, sendo que cada membro é composto de uma estrofe seguida de um refrão opcional. A seqüência como “refrão – estrofe – estrofe – refrão” segue este padrão, como a seqüência “estrofe – refrão – estrofe – refrão”. A seqüência “refrão – refrão – estrofe – estrofe“ não segue o padrão, porem, nenhum faz a seqüência “estrofe – refrão – refrão – estrofe”. Entre outras condições explicitadas neste conteúdo do modelo é requerido que pelo menos a estrofe deve aparecer em um poema, e se ela não for composta somente de linhas, deverá existir um título e uma estrofe nesta mesma ordem.

4.2.4 Atributos

No contexto SGML, a palavra “atributo”, assim como outras palavras, tem um sentido técnico específico. Ela é usada para descrever informação que é em alguns casos a descrição da ocorrência de algum elemento específico mas diz respeito como parte do seu conteúdo [TRA95]. Por exemplo, você pode querer adicionar um atributo *status* para registrar alguns elementos em um documento e indicar seu grau de confiança, ou adicionar um atributo *identificador* nas ocorrências de um elemento em particular para referenciá-lo em outra parte do documento. Atributos são usado especificamente em tais circunstâncias.

Embora elementos diferentes possam ter atributos com o mesmo nome, (por exemplo, todos os elemento podem possuir um atributo chamado *id*), eles são sempre considerados diferentes, e podem ter diferentes valores atribuídos a eles. Se o elemento foi definido como tendo atributos, o valor dos atributos são fornecidos na instância do documento dentro do *start-tag* da ocorrência do elemento.

Por exemplo:

```
<poema id=P1 status”draft”> .... </poema>
```

O elemento **<poema>** foi definido com dois atributos: *id* e *status*. Neste exemplo o atributo *id* tem o valor P1 e o atributos *status* tem o valor *draft* . Um compilador SGML pode usar esses atributos de várias maneiras diferentes; por exemplo: um processo pode imprimir os poemas que tem o atributo *status* com o valor *draft* do mesmo modo que outro processo pode imprimir os poemas com o atributo *status* com o valor *revised*. O identificador *id* é um caso especial, por convenção, ele sempre fornece um valor único para identificar a ocorrência de um elemento, e é muito usado para resolver referências cruzadas.

Como os elementos, os atributos são declarados no *DTD* SGML, usando uma sintaxe similar. Bem como especificar o nome do elemento que está sendo criado, e possível especificar que classe de valores é aceita para o atributo e também o seu valor *default*. A seguinte declaração é usada para definir dois atributos, os quais especificamos acima para o elemento **<poema>*.***

```
<!ATTLIST      poema
         id          ID                            #IMPLIED
         status      (draft | revised | published) draft >
```

A declaração começa com o símbolo ATTLIST, o qual introduz uma lista de especificação de atributos. A primeira parte especifica o elemento (ou elementos) em questão. No nosso exemplo, os atributos estão sendo declarados somente para o elemento **<poema>**. Se vários elementos compartilham o mesmo atributo, eles podem ser definidos um uma única declaração. Seguindo o nome do elemento, uma série de

linhas, uma para cada atributo que está sendo declarado, e cada linha contendo três partes. Essa linha especifica o nome do atributo, o tipo do valor que ele pode assumir , e o seu valor *default*.

Os nomes de atributos (*id* e *status* no nosso exemplo) estão sujeitos as mesmas restrições como outros nomes em SGML; eles não precisam ser únicos em todo o *DTD*, somente dentro da mesma lista de atributos.

A segunda parte da especificação do atributo pode ser feita de duas formas, ambas ilustradas acima. O primeiro caso usa um número especial de palavras chaves para declarar que classe de valor o atributo pode conter. No exemplo acima, a palavra chave *ID* é usada para indicar que o atributo *id* é usado para fornecer um único valor identificador para cada instância de poema. A Tabela 4.1 apresenta as possíveis palavras chaves SGML.

Tabela 4.1: Palavras chaves para especificar conteúdo do atributo.

Palavra Chave	Descrição
CDATA	O valor do atributo pode conter qualquer caracter válido; *tags* podem estar incluídos no valor, mas não serão reconhecidos pelo analisador SGML, e não serão processados normalmente como os demais *tags*.
IDREF	O valor do atributo deve conter um ponteiro para algum outro elemento.
NMTOKEN	O valor do atributo é um *name token,* ou seja, (mais ou menos) qualquer string de caracteres alfanuméricos.
NUMBER	O valor do atributo é composto somente de números.

No exemplo acima, a lista de possíveis valores para o atributo *status* foi fornecida. Isto significa que o compilador pode verificar que no **<poema>** está definido que o atributo *status* pode assumir *draft*, *revised* ou *published* como valores, os quais são apresentados na Tabela 4.2.

Tabela 4.2: Palavras chaves para interpretar a ausência do atributo.

Palavra Chave	Descrição
#REQUIRED	O valor deve ser especificado.
#IMPLIED	O valor não necessita ser fornecido (como no caso do ID acima).
#CURRENT	Se o valor não for fornecido para a ocorrência do elemento, o último declarado deve ser usado.

A última parte da informação de cada definição de atributo especifica como o analisador deverá interpretar a ausência do atributo em questão. Isso pode ser feito pelo fornecimento de uma das palavras chaves listadas abaixo, ou (como neste caso) pelo fornecimento de um valor especifico que é então considerado como o valor para cada elemento que não possui um valor declarado. Usando o exemplo acima, se um poema é simplesmente marcado como **<poema>**, o analisador agirá exatamente como se o poema fosse marcado como **<poema status=draft>**. Alternativamente, uma das seguintes palavras chaves podem ser usadas para especificar o valor *default* de um atributo.

Por exemplo, se a definição do atributo acima for rescrita como:

```
<!ATTLIST      poema
         id          ID                            #IMPLIED
         status      (draft | revised | published) #CURRENT >
```

então os poemas que aparecerem na antologia deverão ser tratados exatamente como o último poema processado pelo analisador.

O que é muitas vezes necessário é referenciar a ocorrência de um elemento textual para outro, um exemplo óbvio são frases tais como “veja nota 6”, “como discutido no Capítulo V”. SGML provê um mecanismo especial pelo qual qualquer ocorrência de um elemento pode receber um identificador especial, o qual pode ser usada para fazer referência em qualquer parte do texto.

Suponha, por exemplo, que nós queremos incluir uma referência onde a nota de um poema se refere a outro poema. Primeiramente é necessário prover uma maneira de colocar um rótulo para cada poema. Isto pode ser feito definindo um atributo para o elemento **<poema>**, como sugerido abaixo.

```
<!ATTLIST      poema
         id          ID                #IMPLIED >
```

Aqui nós definimos um atributo *id* que deve possuir um valor do tipo *ID*. Não é necessário que o tipo *ID* tenha o nome *id*, porem é uma convenção muito utilizada. Note que nem todas as instâncias de poema necessitam receber um atributo *id* pois o analisador saberá seguramente ignorar os poemas sem identificador. Somente os poemas que queremos referenciar devem possuir o identificador. Para cada poema deve-se incluir um *start-tag* com identificador único, por exemplo:

```
<POEMA ID=Poema1>
      Texto do poema com identificador “Poema1”
</POEMA>
<POEMA ID=P40>
      Texto do poema com identificador “P40”
</POEMA>
<POEMA>
      Este poema não tem identificador
</POEMA>
```

Depois precisamos definir um novo elemento para sua referência cruzada. Esse elemento não deve ter conteúdo, deverá ser somente um ponteiro.

```
<!ELEMENT      poemaref    - O   EMPTY >
<!ATTLIST      poemaref          target IDREF #REQUIRED >
```

O elemento **<poemaref>** não precisa de um *end-tag* pois ele não tem conteúdo. Ele possui um simples atributo chamado *target.* O valor deste atributo deve ser do tipo IDREF. Com base nisto, podemos codificar uma referência para o poema com *id Poema1* da seguinte maneira:

```
Isto pode ser encontrado no Primeiro poema <POEMAREF TARGET=Poema1>
```

Quando o analisador SGML encontrar esse elemento vazio ele simplesmente verificará que um elemento existe com o identificador *Poema1.*

4.2.5 Entidades SGML

Os aspectos SGML discutidos até agora estão todos concentrados na estrutura dos elementos em um documento. SGML também provê um simples e flexível método para codificar e rotular partes do conteúdo do documento de uma maneira portável. Em SGML a palavra *entidade* tem um sentido especial: significa o nome de uma parte de um documento *markup*, independente de qualquer consideração estrutural. Uma entidade pode ser um *string* de caracteres ou um arquivo texto. Para incluir uma entidade em um documento, nós usamos uma estrutura conhecida como *referência para entidade*. Por exemplo, na seguinte declaração:

```
<!ENTITY abc “Automatic Bank Check Group” >
```

define uma entidade cujo nome é *abc* e cujo o valor é a *string* “*Automatic Bank Check Group* ”. Isso é uma instância de uma *declaração de entidade*, a qual declara uma *entidade externa.* A seguinte declaração, por outro lado, declara uma *entidade do sistema:*

```
<!ENTITY CapDois SYSTEM “arq2.txt” >
```

Isso define uma entidade do sistema com o nome *CapDois* e cujo o seu valor é um texto associado com o identificador do sistema, neste caso, o identificador do sistema é o nome de uma arquivo do sistema operacional e o texto a ser trocado pela entidade é o conteúdo deste arquivo.

Uma vez que a entidade foi declarada, ela pode ser referenciada em qualquer lugar do texto. Isto é feito pelo fornecimento do nome da entidade com um E

comercial (&) na frente e um ponto-e-vírgula no final. O ponto-e-vírgula pode não existir se a referência da entidade for seguida por um espaço em branco ou um fim de linha.

Quando o analisador SGML encontra uma referência para entidade, ele imediatamente substitui o valor declarado para o nome da entidade. Deste modo a frase "Membro do &abc" depois de interpretada por uma analisador SGML seria lida como "Membro do Automatic Bank Check Group". Isso elimina o trabalho da digitação, e simplifica a tarefa de manter consistência em um conjunto de documentos. Outra aplicação é o mecanismo de substituição de *string*.

4.3 ODA (*Open/Office Document Architecture)*

O modelo ODA propõe um modelo hierárquico orientado a objetos para a representação de documentos. Os componentes de um documento são representados por objetos e seus atributos os quais fornecem as informações sobre os objetos. Este modelo está fundamentado na idéia de que um documento é definido através de um conteúdo e por duas estruturas diferentes que são: estrutura lógica e estrutura física, como mostramos na Figura 4.1.

Figura 4.1: Composição de um documento dentro do modelo ODA.

As estruturas são independentes, mas complementares. Cada estrutura é uma árvore cuja folhas representam o conteúdo do documento, o qual recebe o nome de *content fragment,* e que detalharemos a seguir.

4.3.1 Estrutura Genérica

O modelo ODA propicia a definição de estruturas de objetos genéricas, com o auxílio de um grande número de atributos, a fim de permitir descrever os níveis

lógico e físico de um máximo de tipos de documentos. O nível lógico do modelo ODA é composto de três classes de objetos:

- O documento
- Os objetos lógicos compostos
- Os objetos lógicos simples

Um documento pode ser composto de um ou vários objetos lógicos compostos ou ainda de um ou vários objetos lógicos simples. Da mesma forma, um objeto composto pode ser composto de vários outros elementos lógicos compostos ou de vários elementos lógicos simples. Um objeto lógico simples por sua vez, não pode ser decomposto em outros elementos lógicos simples, mas faz referência a uma ou mais partes do conteúdo do documento. Podemos citar por exemplo, os capítulos das seções de um documento como objetos lógicos compostos, e seus parágrafos como objetos lógicos simples.

No nível físico, o modelo ODA dispõe de cinco classe de objetos:

- O documento
- O conjunto de páginas
- A página
- A região
- O bloco

Como no nível lógico, os tipos de classes de objetos físicos permitem a construção da estrutura física de diferentes tipos de documento.

Um conjunto de páginas permite o re-agrupamento das páginas usando as propriedades que as mesmas apresentam em comum, como por exemplo, as referências bibliográficas, a tabela de índices, etc. Portanto o conjunto de páginas pode ser constituído de uma ou várias páginas e/ou um ou vários conjuntos de páginas subordinadas.

Uma página é uma superfície retangular correspondente a unidade de representação da média utilizada, por exemplo uma folha de papel. Seu canto superior esquerdo é o ponto de referência para o posicionamento dos objetos dentro da página.

A região é uma superfície retangular paralela as bordas da página. Ela pode ser compostas de outras regiões ou ainda de um ou mais blocos. Seu canto superior esquerdo permite posicioná-la dentro de uma página, e serve ao mesmo tempo como ponto de referência para posicionar seus objetos subordinados.

O bloco descreve uma entidade homogênea do documento e é representado na forma de uma superfície retangular paralela as bordas da página. Ele constitui o objeto básico da estrutura física e pode conter uma ou várias partes do conteúdo do documento.

O conteúdo e a sua arquitetura, o qual descreveremos mais adiante, é comum para ambas estruturas (lógica e física). Na verdade o conteúdo serve como ligação entre as duas estruturas., como mostramos na Figura 4.1 (página 80) e mais detalhadamente na Figura 4.2.

Na Figura 4.2 mostramos os objetos genéricos lógicos e físicos definidos no modelo ODA, enquanto que na Figura 4.3 ilustramos a relação entre as estruturas lógica e física e a estrutura do conteúdo do documento.

De fato, um parágrafo por exemplo, pode estar contido dentro de duas páginas diferentes, e deste modo seu conteúdo será dividido em duas partes associadas em blocos diferentes de duas páginas como mostramos na Figura 4.3.

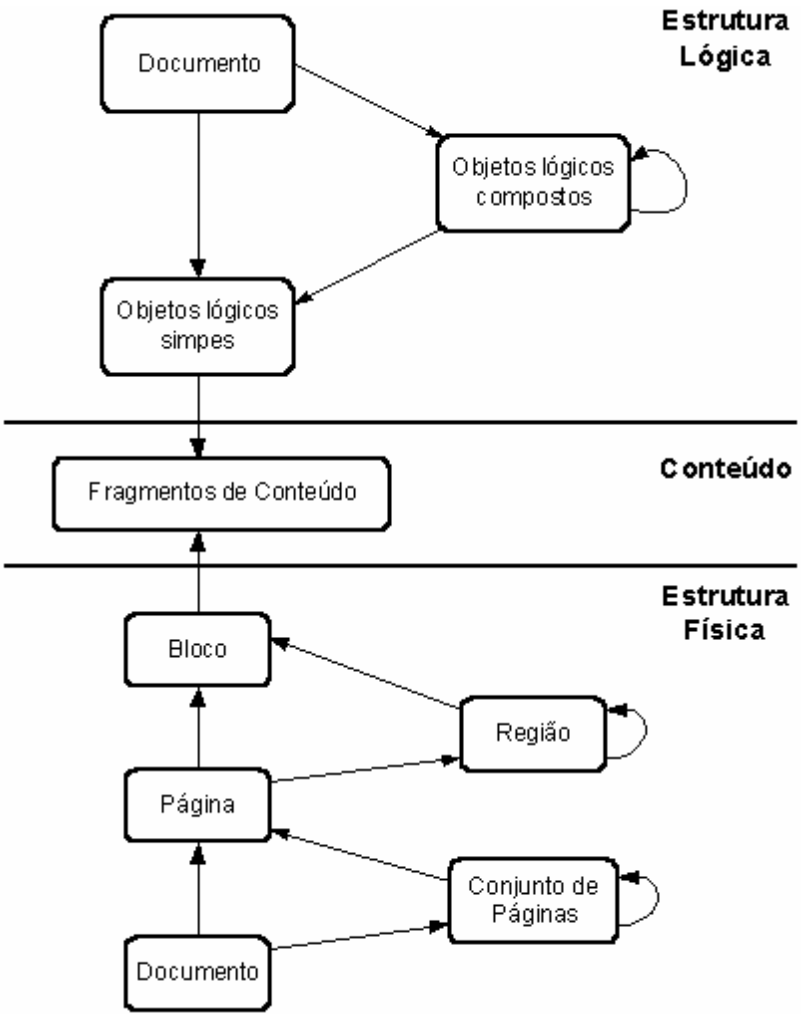

Figura 4.2: Objetos genéricos do modelo ODA.

Da mesma forma, um bloco pode conter as partes do conteúdo de vários objetos lógicos elementares. Por exemplo, um artigo é composto de vários objetos lógicos, como por exemplo, a palavra, sua ortografia, seu sentido, sua utilização, sua categoria gramatical, sua forma fonética, sua etimologia, etc. A seguir apresentamos os atributos que permitem descrever as classes de objetos e a relação entre esses objetos.

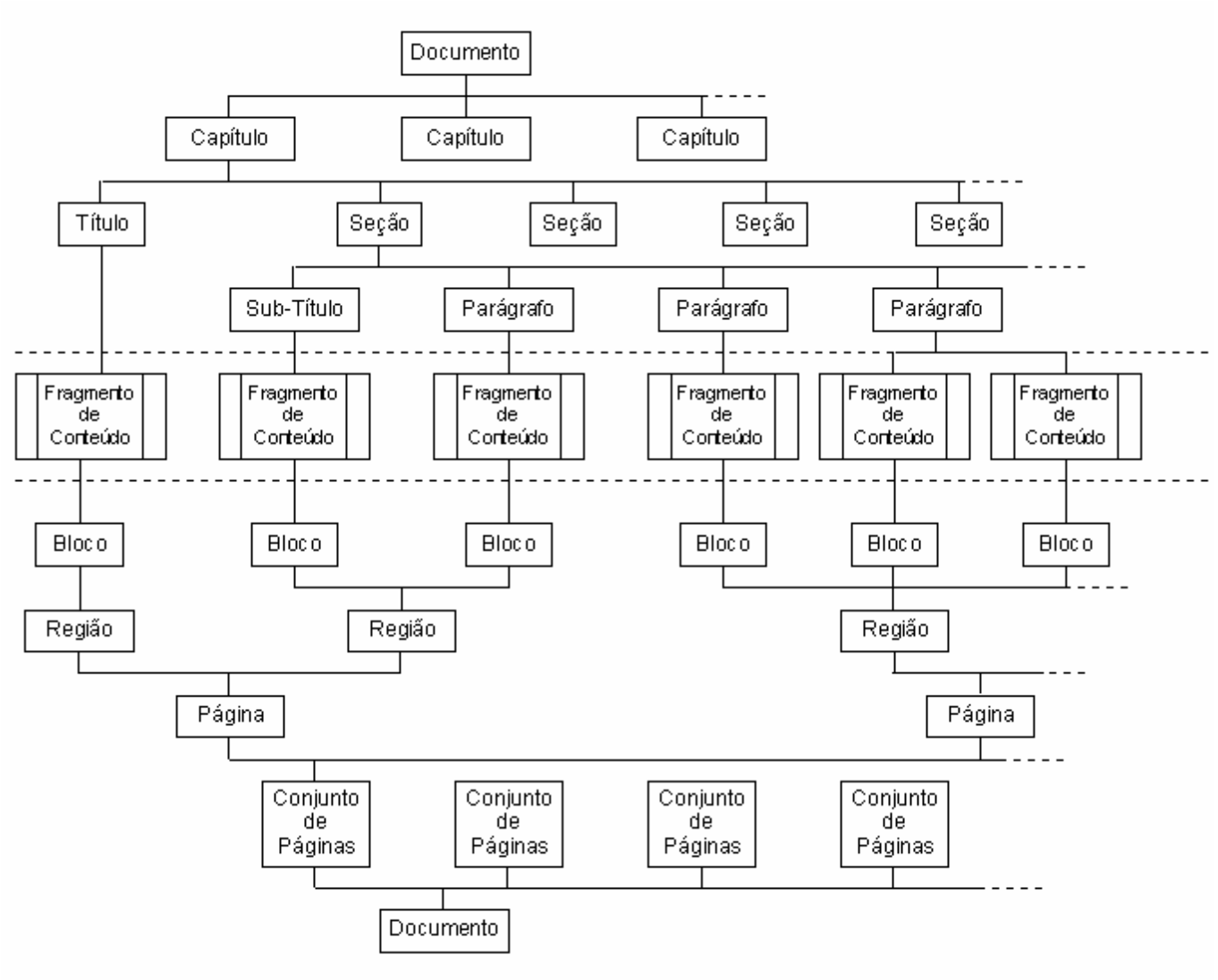

Figura 4.3: Inter-relação entre as estruturas lógica e física especificadas no modelo ODA.

4.3.2 Atributos das Classes de Objetos

O modelo ODA fornece um conjunto de atributos capazes de definir regras para a identificação, a representação e a construção de tipos de classes de objetos definidos acima. Eles permitem também especificar suas propriedades. Os atributos podem levar a forma de uma referência de um objeto definido anteriormente, uma expressão de avaliação em função de uma aplicação dada, ou simplesmente uma constante alfanumérica.

O modelo ODA propõe também os atributos para determinar o modo genérico da hierarquia entre os objetos da mesma estrutura. Os atributos são chamados de *geradores de objetos dependentes*. Os atributos permitem especificar se um conjunto de objetos pode ser considerado como:

- Uma seqüência (SEQUENCE) – aparece em uma ordem precisa.
- Um agregação (AGGREGATE) – aparece sem uma ordem precisa.
- Uma escolha (CHOICE) – escolha de um objeto dentro de um conjunto de objetos.

E ainda, os atributos permitem precisar se um objeto dependente é:

- Opcional (OPTIONAL) – zero ou uma ocorrência.
- Obrigatório (REQUIRED) – pelo menos uma ocorrência.
- Repetitivo (REPETITIVE) – várias ocorrências.
- Opcional e repetitivo (OPTIONAL e REPETITIVE) – zero ou várias ocorrências.

Com a ajuda dos atributos, as estruturas genéricas de um documento poderão ser definidas. A seguir mostraremos nas Tabelas 4.3 e 4.4 um exemplo de modelos genéricos determinando sua forma de uma gramática fora de contexto, onde os atributos são utilizados para construir as estruturas lógica e física de um documento do tipo relatório.

Tabela 4.3: Exemplo do modelo lógico genérico dentro do modelo ODA.

Classe		Descrição
Relatório	::=	SEQUENCE(Cabeçalho, Tabela de Índice, SEQUENCE(Capítulos))
Cabeçalho	::=	SEQUENCE(Nome de Autor, Título, Data)
Tabela de Índice	::=	SEQUENCE(Entrada de Tabela de Índice)
Entrada de Tabela de Índice	::=	SEQUENCE(Cabeçalho de Capítulo, Título do Capítulo, Referência de Página)

Capítulo	::=	SEQUENCE(Cabeçalho do Capítulo, Sumário do Capítulo(OPTIONAL), SEQUENCE(Seção))
Seção	::=	SEQUENCE(Cabeçalho de Seção, SEQUENCE (Elementos da Seção))
Elemento da Seção	::=	CHOICE(Parágrafo, Figura)
Figura	::=	AGGREGATE(*Picture,* Título)

Tabela 4.4: Exemplo do modelo físico genérico dentro do modelo ODA.

Classe		Descrição
Relatório	::=	SEQUENCE(Título, Conjunto de Páginas, Capítulo-Conjunto de Páginas)
Título	::=	SEQUENCE(Bloco-Autor, Bloco-Título, Bloco-Data, Organização da Disposição
Organização da Disposição	::=	SEQUENCE(Organização Bloco-Nome, Organização Bloco-Endereço, Bloco-Copyright)
Conjunto de Páginas	::=	SEQUENCE(Primeira Página, SEQUENCE(Página Continuação(OPTIONAL)))
Primeira Página	::=	SEQUENCE(Bloco-Titulo, Quadro de Conteúdo)
Página Continuação	::=	SEQUENCE(Continuação Cabeçalho Quadro, Quadro de Conteúdo)
Continuação Cabeçalho Quadro	::=	Continuação Cabeçalho Bloco
Quadro de Conteúdo	::=	SEQUENCE(Bloco de Conteúdo)
Capítulo Conjunto de Páginas	::=	SEQUENCE(Capítulo Primeira Página, SEQUENCE(Capítulo Continuação Página(OPTIONAL)))
Capítulo Primeira Página	::=	SEQUENCE(Capítulo Cabeçalho Bloco, Capítulo Conteúdo Bloco)
Capítulo Continuação Página	::=	SEQUENCE(Capítulo Conteúdo Quadro, Número de Bloco da Página
Capítulo Conteúdo Quadro	::=	SEQUENCE(Capítulo Conteúdo Bloco)

Na Figura 4.4 ilustramos os diversos conceitos sobre a estrutura lógica genérica de uma classe de documentos do tipo carta. O objeto CARTA é o tipo de documento e os objetos TÍTULO, CORPO, FIGURA, LISTA E RODA-PÉ são os objetos compostos, e os outros objetos apresentados são os objetos básicos.

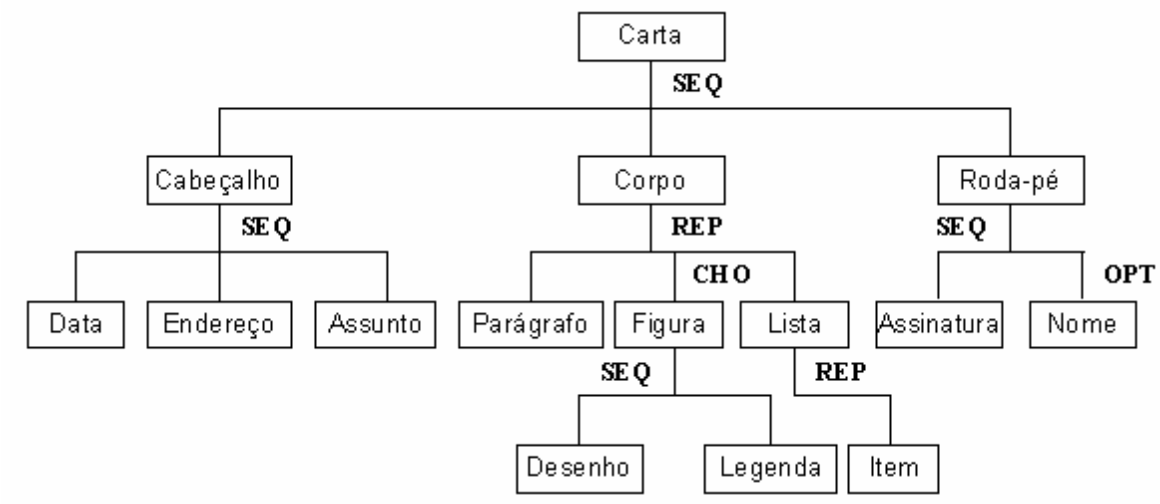

Figura 4.4: Exemplo de uma estrutura lógica genérica no modelo ODA.

4.3.3 Arquitetura do Conteúdo

O modelo ODA faz uma clara distinção entre as estruturas lógica e física do documento e o seu conteúdo. As estruturas descrevem unicamente a arquitetura do documento, enquanto que o conteúdo é descrito em uma estrutura interna chamada de *arquitetura do conteúdo.* Esta arquitetura é constituída de um conjunto de regras descritas na estrutura interna dos objetos elementares do documento. Ela é especificada através dos atributos de apresentação associados aos objetos elementares e de funções de controle que dependem da natureza do conteúdo considerado. O modelo ODA define três tipos de arquitetura de conteúdo:

- Texto: É uma cadeia de caracteres e de funções de controle. Os atributos utilizados pelas regras determinam o posicionamento e a forma de cada caracter dentro de um bloco, fornecendo assim a sua colocação na página. Geralmente os caracteres são colocados da esquerda para a direita em uma linha começando no alto de um bloco. O modelo ODA distingue dois tipos de atributos para o

conteúdo do tipo texto. O primeiro gerencia o transbordamento de uma parte entre dois blocos vizinhos e determina:

- sentido da progressão dos caracteres em sua linha
- sentido de progressão das linhas dentro do bloco
- tipo de justificação ou alinhamento das linhas em um bloco
- A indentação da primeira linha em um bloco

O segundo determina:

- início e o fim dos índices e das notas
- A fonte que deve ser utilizada para formar o caractere

- Figura: É um conjunto de pixels que permite formar uma imagem. Cada pixel é caracterizado por um conjunto de atributos que determina sua forma, seu tamanho, sua cor, sua intensidade e a sua posição. As funções de controle são utilizadas para definir a resolução da imagem e para marcar o fim do bloco do tipo Figura.

- Geométrico: Esse tipo é descrito de acordo com o padrão CGM (*Computer Graphics Metafile)* [ISO87]. Os objetos geométricos primitivos são compostos por linhas, texto, etc. Os objetos são posicionados dentro das coordenadas do sistema virtual (*VDC – Virtual Device Coordinates*) e seu aspecto é controlado por atributos que determinam a apresentação de uma linha (largura e cor) e por atributos que determinam o tamanho da imagem.

4.3.4 Formato de Troca

ODA é um padrão que serve para qualquer um que queira compartilhar documentos sem sacrificar o controle sobre o conteúdo, estrutura e *layout* desses documentos. Ele é projetado para resolver dificuldades criadas pela variedade de formatos de documentos existentes. Um documento ODA pode ser aberto, mudado, trocado, armazenado e reproduzido por qualquer programa compilador-ODA [CAR91].

ODA foi desenvolvido para facilitar a interoperabilidade entre diferentes sistemas processadores de documentos. A permuta de documentos ocorre quando uma pessoa envia um documento para outro. Usuários podem preferir a troca eletrônica de documentos por várias razões.

- A velocidade da transmissão que pode ser rápida.
- A distribuição eletrônica é mais eficiente, se existe mais que um receptor.
- O receptor pode querer armazenar o documento em um sistema para recuperá-lo posteriormente.
- O documento pode conter dados que não são possíveis de serem colocados no papel, como por exemplo, voz.

Como ODA foi desenvolvido para ser extensível, o padrão ODA foi escrito para ter um escopo muito grande. Na prática, entretanto, não existe dispositivo que suporte todas as características possíveis. Como então garantir que o emissor e o receptor de um documento suportem um conjunto compatível de características.

A resposta para este problema é um conjunto de definições, o *Document Application Profiles* (DAP). Ele é construído em níveis de funcionalidade, ou seja, cada DAP define uma lista de características suportadas, na qual qualquer sistema em um mesmo nível ou maior deve ser capaz de aceitar ou interpretar corretamente um documento.

Alguns conjuntos de regras existentes tais como FOD011, FOD026 e FOD036 são os que permitem que a permuta de documentos seja efetuada.

4.3.4.1 ODA - *Simple document structure* - FOD011

Este conjunto provê a permuta de documentos que contêm somente caracteres. O conteúdo do documento é representado na forma de uma seqüência de parágrafos. Esses parágrafos podem ser arranjados em grupos de um ou mais parágrafos para permitir diferentes *layout's* e apresentação de características para ser especificados por cada grupo.

4.3.4.2 ODA - *Enhanced document structure* - FOD026

Este conjunto provê a permuta de simples documentos multimídia entre sistemas processadores de documentos dentro de um ambiente integrado de escritório. Os documentos podem conter caracteres, diagramas e outras formas de ilustrações. Este nível foi desenvolvido para unir as capacidades da indústria de pacotes de processamento de texto bem como suprir o uso do conjunto internacional de caracteres, a variedade de fontes e padrões de tamanhos de páginas tais como ISO A4 e *North Americam Letter*. Este conjunto também provê características para escrita vertical e documentos orientais.

O FOD026 possui todas as características suportadas pelo FOD011.

4.3.4.3 ODA - *Extended document structure* - FOD036

Este conjunto provê a permuta de documentos com características tais como tabelas e ilustrações sobrepostas. Este nível foi desenvolvido para suprir as necessidades das aplicações de editoração eletrônica.

O FOD036 possui todas as características suportadas pelo FOD026.

4.4 Comparação entre os modelos ODA e SGML

Os modelos ODA e SGML foram criados para facilitar a troca de documentos entre diferentes sistemas de tratamento de documentos. É claro que os seus domínios de aplicação não são os mesmos. As suas diferenças se situam no nível físico, pois o SGML não se prende a construir a estrutura física por completo (os modelos DSSSL e SPDL completam esta lacuna). Entretanto as duas normas procuram um formalismo de estruturação que possa ser utilizado em sistemas de Análise e Reconhecimento de Documentos. Akindele em [AKI95] coloca alguns pontos fortes e fracos de cada um do modelos:

- **Modelo ODA**

Os pontos fortes são:

- A obrigação de uma descrição rigorosa das estruturas física e lógica ao mesmo tempo.
- A separação do conteúdo da estrutura do documento: Um editor de texto especializado e orientado para a especificação da estrutura mantém a coerência do conjunto, ou seja do conteúdo e das estruturas física e lógica.
- A definição de atributos de maneira precisa e a normalização da linguagem utilizada para a geração dos objetos subordinados. Isto que limita as fantasias de apresentação e oferece a um sistema de Análise e Reconhecimento de Documentos um contexto bem definido para pesquisa.
- Um único arquivo ODA pode encapsular um documento composto de vários arquivos, fazendo com que a distribuição deste se torne mais simples.

Os pontos fracos são:

- Os tipos de objetos físicos possíveis em ODA não permitem a descrição de documentos artisticamente complexos. Os objetos físicos são sempre retangulares e paralelos entre si, o que impede, por exemplo, definir regiões poligonais.
- A indexação de arquivos é difícil, pois o arquivo resultado não é composto por *markup's* .

- **Modelo SGML**

Os pontos fortes são:

- Os arquivos SGML são editáveis, pois a estrutura está incorporada ao conteúdo em forma de *markup*. Portanto, é fácil para um sistema de Análise e Reconhecimento de Imagem criar um arquivo compatível SGML.
- SGML é uma metalinguagem, sendo possível estabelecer um padrão para escrever *markup* respectivos a estrutura ODA.
- A facilidade de indexar arquivos através de *markup' s*.

Os pontos fracos são:

- Não é possível definir a estrutura física precisamente como no modelo ODA, o que é muito importante para alguns sistemas de Análise e Reconhecimento de Documentos.
- A ausência de uma estrutura pré-determinada, pode levar o sistema a criar estruturas falhas. Isso torna ainda mais necessário o uso do conhecimento da estrutura do documento.
- Um documento SGML é composto de no mínimo dois arquivos, ou seja, o seu código fonte e o seu DTD.

No SGML o *markup* descreve a estrutura do documento e outros atributos, além de especificar como este documento deverá ser processado. Por exemplo, para incluirmos um título em uma artigo, basta colocar o *tag* <titulo> e não é necessário dizer “Centralizado, Negrito, 16pt Times New Roman”. A descrição do <titulo> está contida no DTD – *Document Type Definition.*

O ODA suporta esta funcionalidade usando um mecanismo chamado *Document Classes.* O modelo ODA foi criado primeiramente com o objetivo de prover a troca eletrônica de documentos, pois um simples arquivo ODA pode encapsular vários documentos. A distribuição em ODA requer que um simples arquivo seja transmitido, enquanto que em SGML, são necessários no mínimo dois arquivos.

SGML usa uma representação baseada em ASCII, a qual fornece algumas limitações para construção de arquivos. Existe um subterfúgio, que permite embutir dados binários em um documento desde que esses elementos terminem com uma seqüência especial de caracteres, e que essa seqüência seja possível em qualquer arquivo binário. É possível evitar isso pelo uso de seqüência de *escapes,* porém isto não é definido pela norma ISO SGML. O método aceito pela norma é referenciar os arquivos como entidades externas. ODA usa arquivos binários expressados pela norma ASN.1. É interessante notar, que uma das principais razões pelas quais o SGML é muito usado, é o fato de gerar uma representação ASCII, o que torna muito simples “vasculhar” em um arquivo SGML usando um simples editor de texto. Os arquivos ODA são codificados no padrão conhecido como ODIF (*Open Document Interchange Format).*

4.5 Neste Capítulo

Neste capítulo apresentamos um estudo teórico sobre os dois principais padrões para armazenamento das estruturas lógica e física de um documento. Primeiramente discutimos o SGML, o qual através de um exemplo, demonstramos a sua potencialidade. Logo após apresentamos a teoria do padrão ODA, e como este são definidos os modelos físicos e lógicos, bem como o relacionamento entre esses modelos. Foi mostrado ainda os formatos de troca suportados pelo padrão ODA.

Finalizando este capítulo, apresentamos um comparativo entre os dois padrões, apresentando pontos fortes e fracos de cada um deles.

No capítulo seguinte mostraremos as etapas do processamento do cheque bancário desde a sua aquisição, até a obtenção do arquivo do cheque bancário codificado.

Capítulo V
Implementação

5.1 Introdução

Neste capítulo mostraremos o processamento necessário para se obter as estruturas lógica e física do cheque bancário a partir de uma imagem digital em níveis de cinza. Abaixo descrevemos as etapas desde a aquisição da imagem do cheque, passando pela segmentação, reconhecimento e codificação até a obtenção do arquivo do cheque bancário codificado.

Veremos também como as informações do cheque bancário podem ser codificadas utilizando os padrões SGML e ODA.

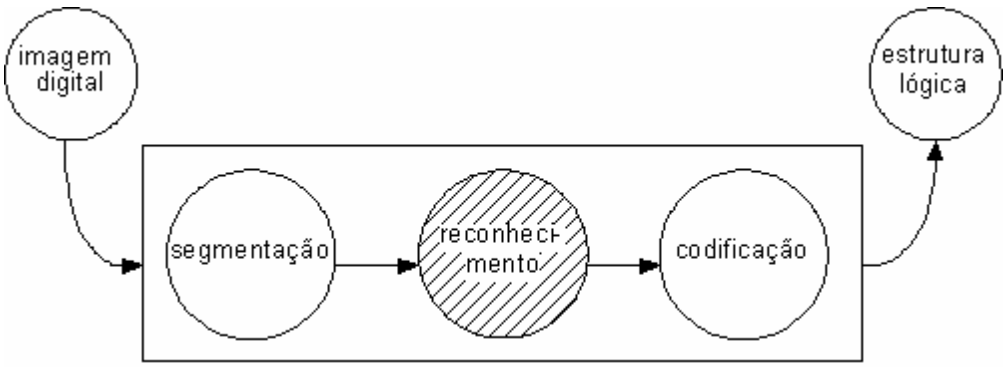

Figura 5.1: Etapas do processamento do cheque.

5.2 Aquisição da Imagem

As aquisições das imagens dos cheques foram feitas utilizando um Scanner Genius modelo ColorPage-SP2. Essas imagens tem as seguintes características: definição de 300 dpi, resolução aproximada de 1350x600 pixels e 256 níveis de cinza.

5.3 Segmentação

Esta parte do processo visa obter uma imagem digital binária de boa qualidade, a qual será o parâmetro de entrada para o processo de reconhecimento. A obtenção de uma imagem de má qualidade nessa fase do processo implicará em uma estrutura lógica errada.

As diferentes etapas da segmentação estão descritas na Figura 5.2

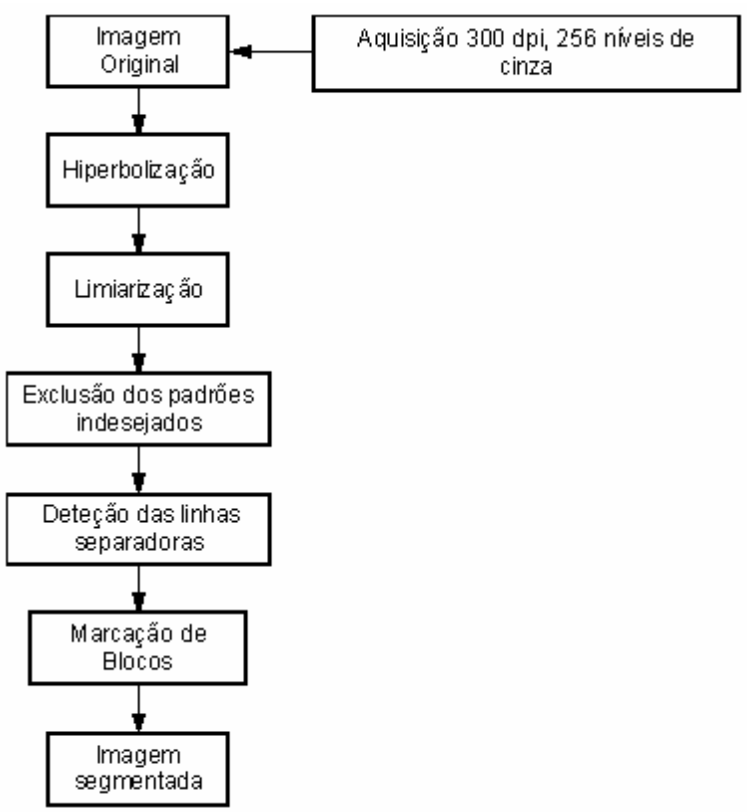

Figura 5.2: Etapas na segmentação do cheque bancário brasileiro.

Abaixo abordamos o problema da limiarização de imagens em níveis de cinza no cheques bancários, descrevemos a etapa de eliminação dos padrões indesejados, bem como a detecção automática das linhas separadoras dos campos e a marcação de blocos.

5.3.1 Limiarização da imagem

O objetivo da limiarização é encontrar uma imagem binária de boa qualidade. Como vimos no Capítulo II, o processo da equalização por hiperbolização quadrática do histograma [COB96] suaviza os padrões do fundo do cheque, tornando o processo da limiarização mais eficiente. Partindo-se destes conceitos, aplicamos os seguintes processos na imagem original do cheque: equalização por hiperbolização quadrática do histograma [SAN97] seguida da limiarização pelo método de Otsu [SAH88].

Após este processo, temos três imagens. A imagem 1 que é a imagem original (Figura 5.3), a imagem 2 que é o resultado da equalização por hiperbolização (Figura 5.4), e a imagem 3 que é a imagem binária (Figura 5.5), a qual será a base das operações daqui para frente.

5.3.2 Exclusão dos padrões indesejados

Para que possamos marcar os blocos que deverão ser reconhecidos, primeiramente devemos excluir todos os padrões indesejados no formulário do cheque. Os principais padrões não necessários para o trabalho são: as linhas verticais, linhas horizontais e os nomes dos campos. As linhas servem na maioria das vezes como delimitadoras dos campos, sendo que as linhas horizontais ainda tem a função de linhas de base para o preenchimento do cheque.

Para realizar esta tarefa aplicamos duas visões, as quais chamamos de global e local. A análise global aplica a morfologia matemática no documento como um todo, não levando em consideração nenhum conhecimento prévio da estrutura. Obtemos um resultado muito satisfatório, porém restrito a um tipo de formulário, ou seja, a mesma seqüência de operações morfológicas em outro tipo de cheque traz um resultado muito ruim. Se a solução buscada for somente para um tipo de cheque, a análise global pode fornecer bons resultados. Por outro lado, a análise local divide o formulário do cheque em duas partes distintas: o cabeçalho e o corpo do cheque. O cabeçalho é composto da primeira faixa do cheque, enquanto o corpo é composto das

demais faixas. Como o nome diz, as operações morfológicas são aplicadas em cada uma das faixas separadamente. Para a divisão do formulário em faixas, se faz necessário o uso da estrutura física do formulário do cheque bancário, que por sua vez é bem definida pelas normas do Banco Central [BCO83]. A seguir mostraremos a seqüência de operações morfológicas para as duas visões.

5.3.2.1 Análise Global

Para a exclusão das linhas verticais utilizamos a teoria da reconstrução em imagens binárias. As seguintes operações foram realizadas:

- Erosão binária da imagem 3 (Figura 5.5) usando o elemento estruturante linha horizontal com uma iteração, visando a remoção das linhas verticais. Esse processo gerou a imagem número 4 (Figura 5.6).

- Reconstrução híbrida rápida da imagem 4 (Figura 5.6), usando um elemento estruturante quadrado e a imagem 3 (Figura 5.5) como máscara. Esse processo gerou a imagem número 5 (Figura 5.7).

Como podemos perceber na Figura 5.7, alguns padrões mais fracos da imagem começam a ser eliminados, tais como as inscrições menores do cabeçalho do cheque. Poderíamos através da reconstrução recuperar esses padrões, mas não é o nosso objetivo.

Para a exclusão das linhas horizontais utilizamos a mesma idéia apresentada acima, com diferença apenas no elemento estruturante aplicado. As operações realizadas foram as seguintes:

- Erosão binária da imagem 5 (Figura 5.7) usando o elemento estruturante linha vertical com uma iteração, visando a remoção das linhas horizontais. Esse processo gerou a imagem número 6 (Figura 5.8).

- Reconstrução híbrida rápida da imagem 6 (Figura 5.8), usando um elemento estruturante quadrado e a imagem 5 (Figura 5.7) como máscara. Esse processo gerou a imagem número 7 (Figura 5.9).

É possível notar na Figura 5.9 que não existem mais sinais das linhas na imagem do cheque, porém ainda existe um número grande de padrões indesejados na imagem do cheque, os quais nos causarão problemas na próxima etapa se não forem eliminados agora. Deste modo, mostraremos abaixo os passos necessários para eliminar a maioria desses padrões.

Aplicamos as seguintes operações neste último passo da eliminação dos padrões indesejados:

- Erosão binária da imagem 7 (Figura 5.9) usando o elemento estruturante quadrado com uma iteração. Esse processo gerou a imagem número 8 (Figura 5.10).

- Reconstrução híbrida rápida da imagem 8 (Figura 5.10), usando um elemento estruturante quadrado e a imagem 5 (Figura 5.7) como máscara. Esse processo gerou a imagem número 9 (Figura 5.11).

Na Figura 5.11 mostramos o resultado final da segunda parte do processo de segmentação pela análise global.

O grande problema encontrado nesta análise, é que em alguns tipos de cheques, os nomes dos campos são impressos em fontes maiores ou fontes mais fortes do que as informações que queremos recuperar, consequentemente, a tentativa de excluir esses padrões excluem juntamente os padrões desejados. Para contornar este problema, apresentamos abaixo a análise local.

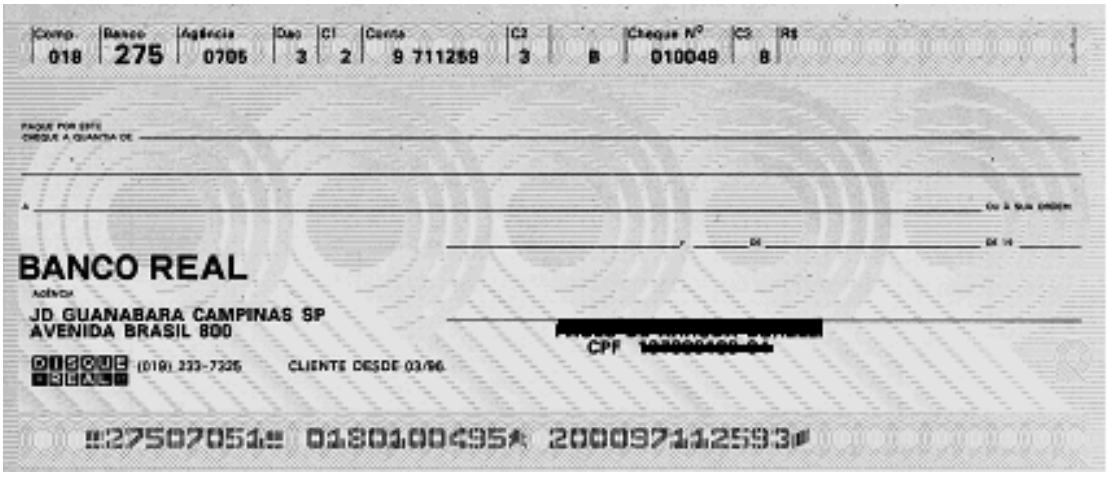

Figura 5.3: Imagem original do cheque.

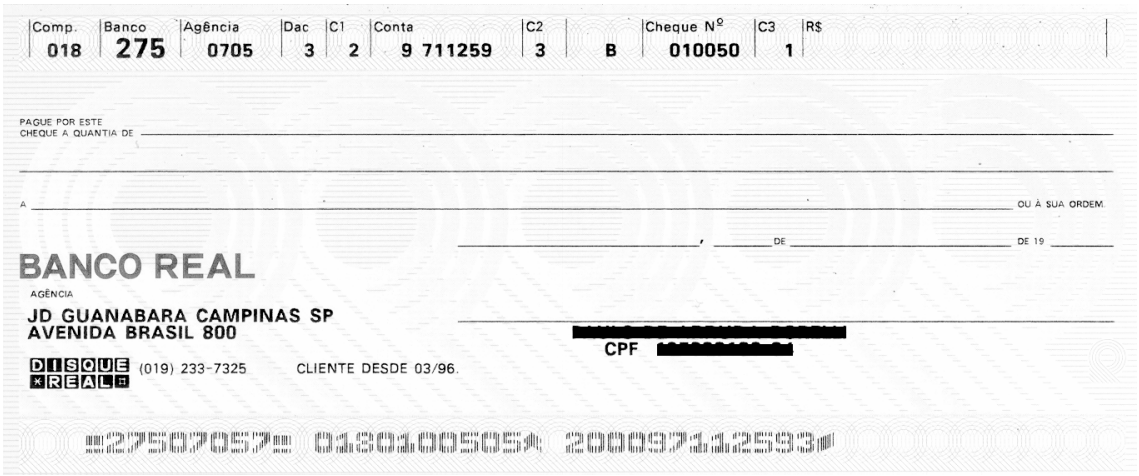

Figura 5.4: Resultado da equalização por hiperbolização quadrática do histograma.

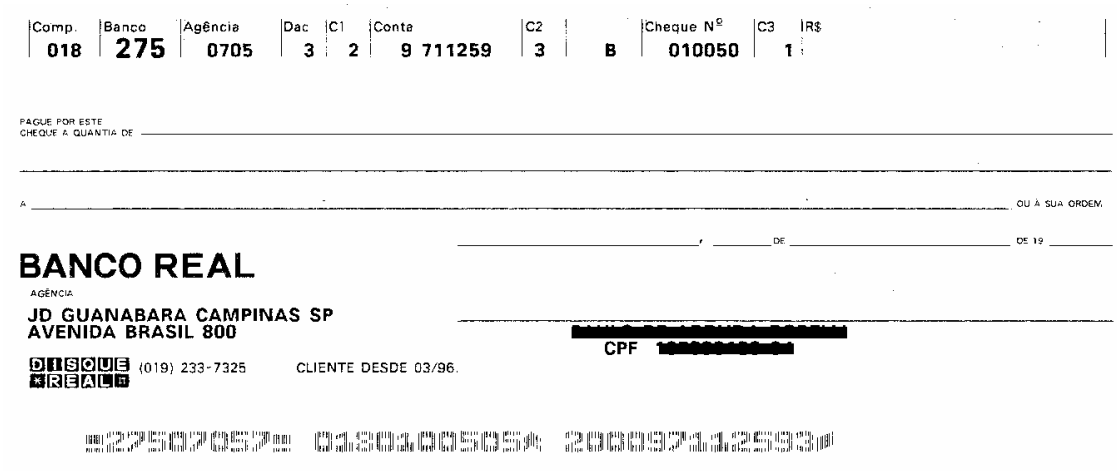

Figura 5.5: Resultado do processo de limiarização pelo método de Otsu.

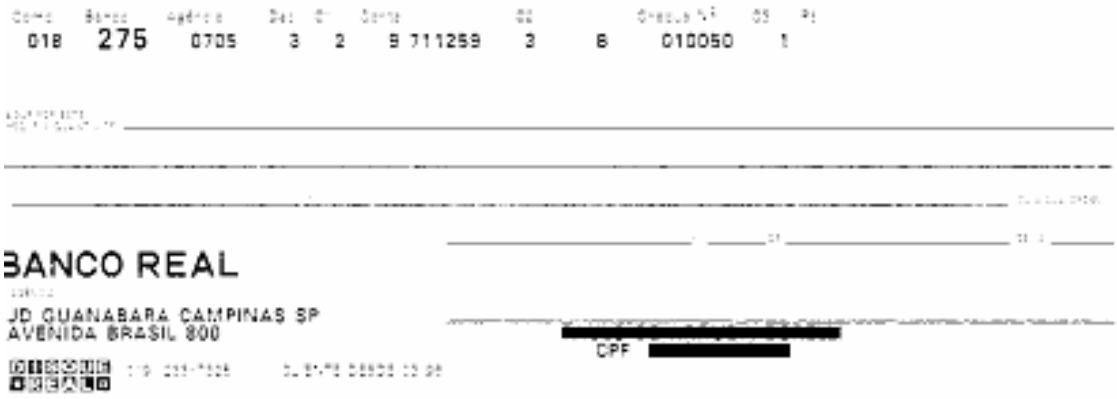

Figura 5.6: Imagem do cheque sem as linhas verticais.

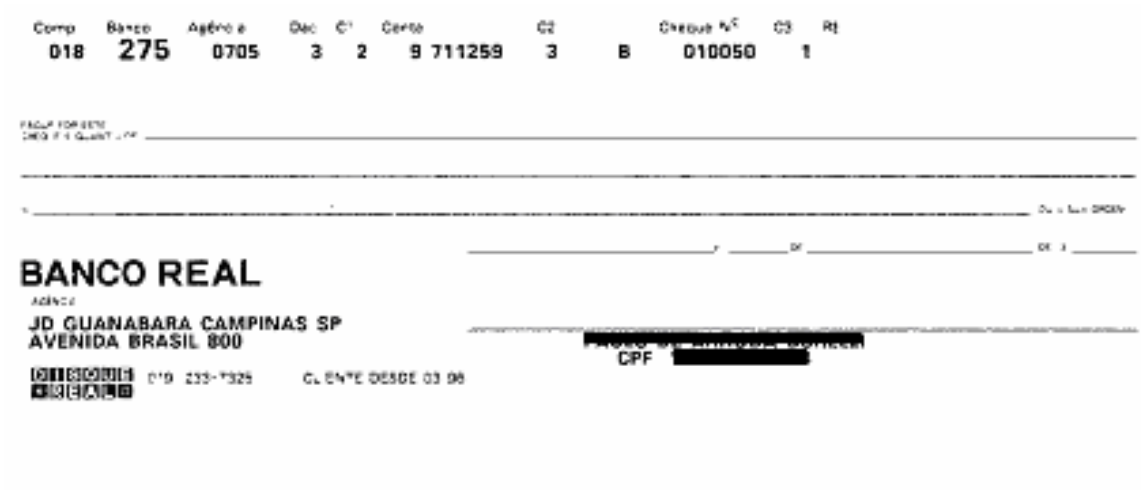

Figura 5.7: Imagem do cheque sem as linhas verticais após a reconstrução.

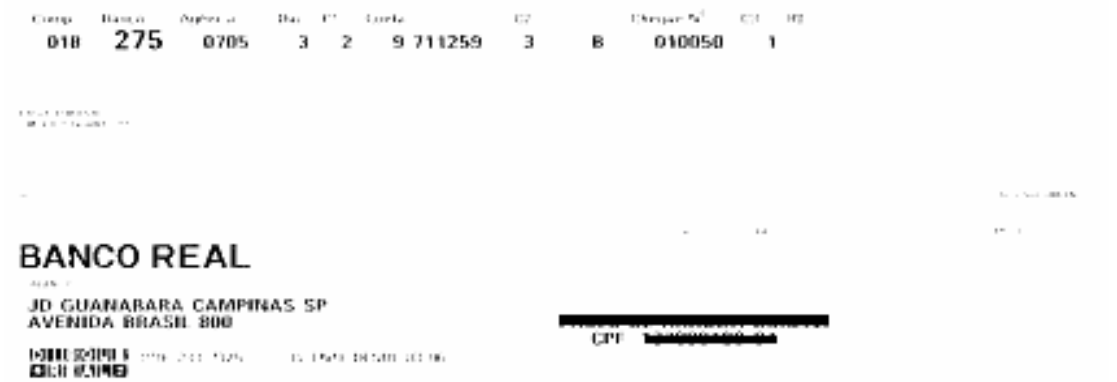

Figura 5.8: Imagem do cheque sem as linhas horizontais.

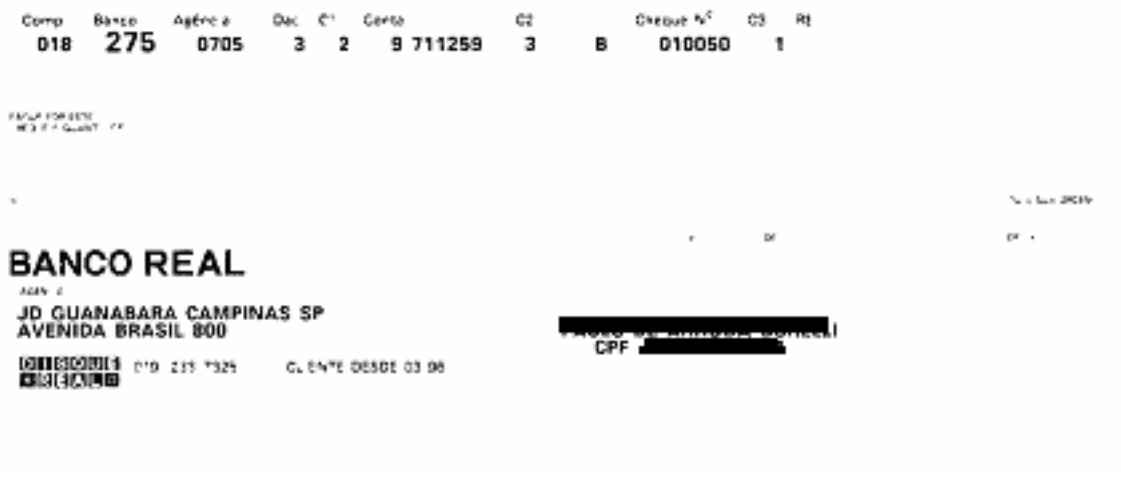

Figura 5.9: Imagem do cheque sem as linhas horizontais após a reconstrução.

Figura 5.10: Imagem do cheque após o processo de erosão binária.

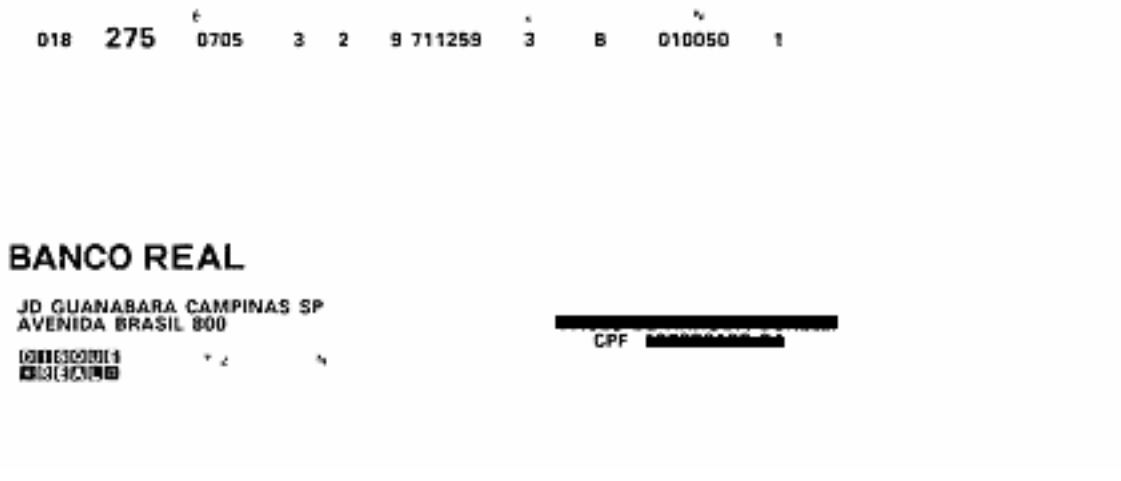

Figura 5.11: Imagem do cheque após a reconstrução da Figura 5.10.

5.3.2.2 Análise Local

A análise local consiste em separar o cheque em 4 faixas distintas. A primeira faixa corresponde aos nomes dos campos do cabeçalho do cheque que se estende até uma linha que chamamos de “delimitador1”, a segunda faixa corresponde ao cabeçalho do cheque e vai até uma linha que chamamos de “delimitador2”, a terceira faixa corresponde ao corpo do cheque e vai até uma linha que chamamos de “delimitador3”, e quarta faixa corresponde ao rodapé do cheque. Podemos visualizar esta divisão na Figura 5.12.

A remoção das linhas verticais (horizontais) faz-se por erosão suave em duas iterações com o elemento estruturante linha horizontal (linha vertical) seguida de uma reconstrução com o elemento estruturante quadrado. Essa reconstrução faz se necessária para conservar intactas as informações do cabeçalho e do corpo do cheque. Infelizmente, não é sempre possível eliminar totalmente os nomes dos campos do cabeçalho sem degradação das informações relevantes. A localização do “delimitador1” é feita buscando a primeira linha totalmente vazia após a região dos nomes dos campos do cabeçalho, porém se esta linha não for encontrada, ela será criada baseada no conhecimento do *layout* do cheque bancário e em um limiar heurístico, o qual limita a região de procura em 20 linhas após o começo dos nomes do campos do cabeçalho.

Da mesma maneira, a linha “delimitador2” é obtida buscando a primeira linha totalmente vazia após a região do cabeçalho do cheque. A linha “delimitador3” é obtida buscando a primeira linha totalmente vazia antes do rodapé.

Todas as informações localizadas antes do “delimitador1” são os nomes dos campos, portanto padrões indesejados e são eliminados facilmente. Todas as informações localizadas entre o “delimitador1” e o “delimitador2” são essenciais e portanto extraídas.

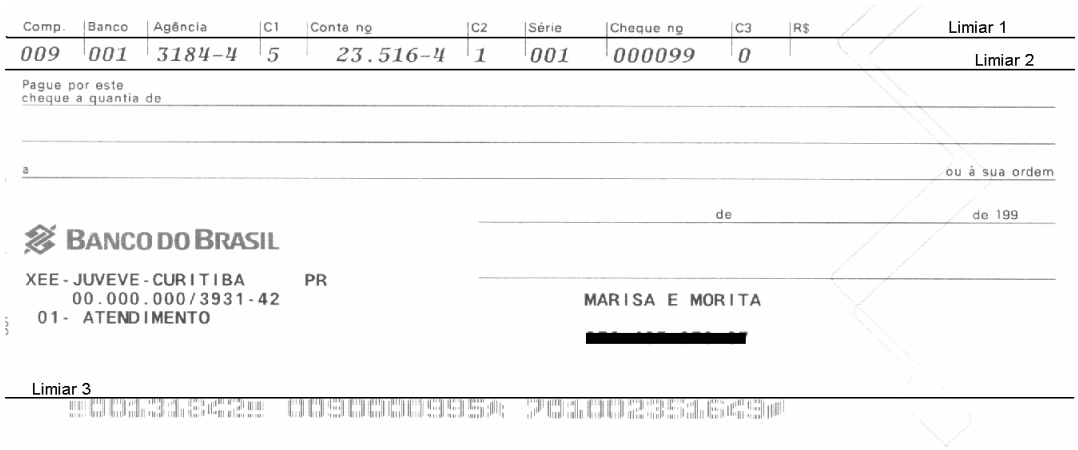

Figura 5.12: Delimitadores do cheque na análise local.

Na Figura 5.13 mostramos a faixa extraída para o processamento.

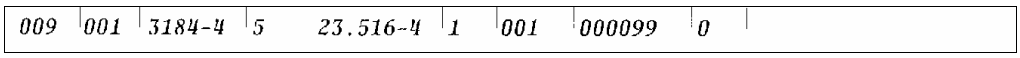

Figura 5.13: Imagem dos campos da segunda faixa.

Na terceira região localizada entre o “delimitador2” e o “delimitador3”, a eliminação das linhas horizontais é fácil (erosão suave com elemento estruturante linha vertical). A exclusão de informações não necessárias para o nosso processamento, tais como *“pague por este cheque a quantia de” ou ainda “ou a sua ordem”* é realizada com o uso do conhecimento da estrutura física do documento padronizada pelo Banco Central do Brasil [BCO83], [TAN91]. Após o processamento acima descrito, obtemos como resultado a imagem da Figura 5.14.

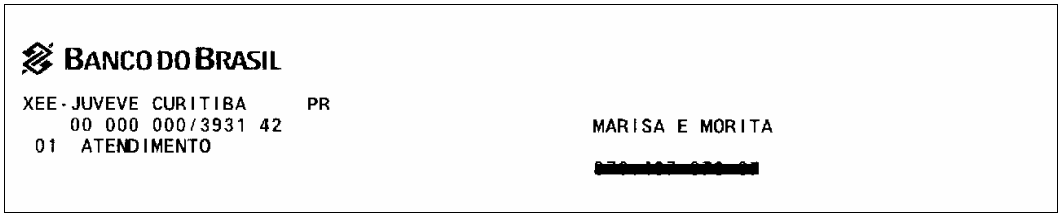

Figura 5.14: Imagem dos campos da terceira faixa.

O processamento da quarta faixa é o mais simples. A única operação que deve ser realizada é a exclusão de todas as linhas abaixo do “delimitador3”.

Após o processamento local de cada uma das partes da imagem, realizamos uma operação lógica OU entre o resultado do processamento da segunda faixa e o resultado do processamento da terceira faixa, com o intuito de criar uma imagem resultante final, conforme Figura 5.15.

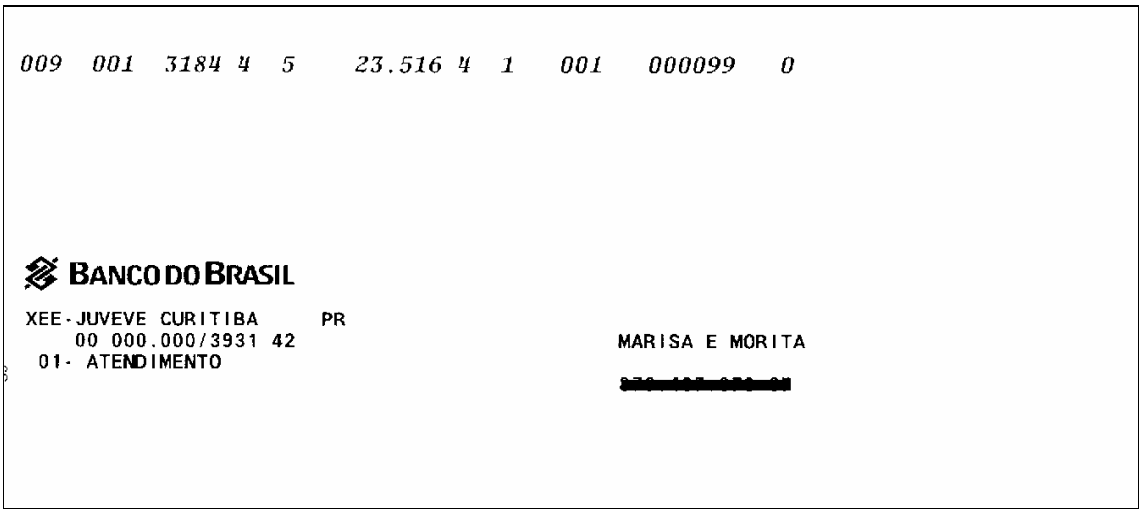

Figura 5.15: Resultado final na análise local (equivalente a Imagem 9 da análise global).

Na Figura 5.16 mostramos o resultado do mesmo cheque que exemplificamos a análise local, usando a análise global.

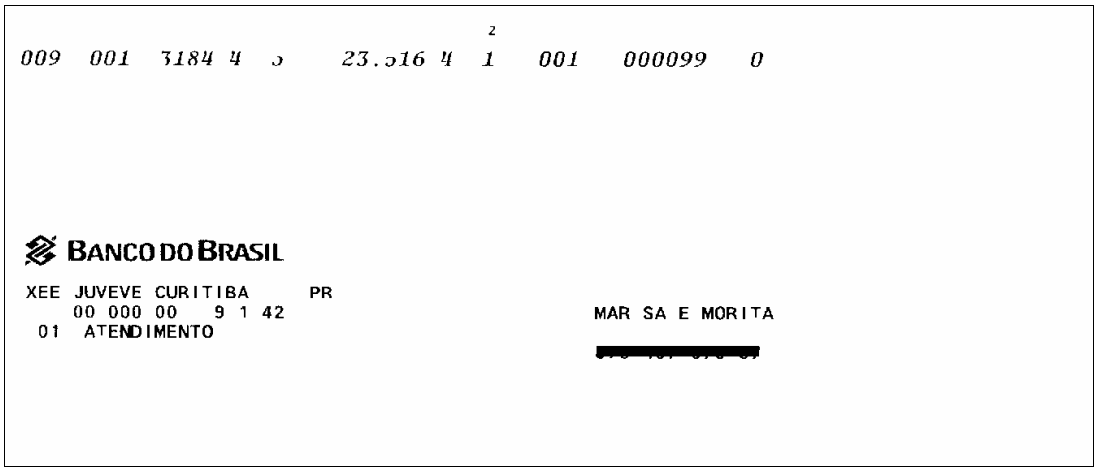

Figura 5.16: Resultado final na análise global.

Podemos notar que alguns padrões que desejávamos recuperar foram perdidos, como por exemplo, uma parte dos dados da agência, a letra “I” do nome “MARISA”, e alguns padrões indesejáveis continuaram após o processamento, como por exemplo, um pequeno pedaço do nome de um campo na primeira faixa do cheque.

Como podemos notar, os padrões mais prejudicados na análise global são os que se parecem com linhas verticais ou horizontais, como por exemplo a letra “I”, os caracteres “/” e “-“. A análise global tem a vantagem de ser um processo mais rápido, porém a sua aplicação é restrita a tipos específicos de cheques, enquanto que a análise local garante uma aplicabilidade para uma gama maior de tipos de cheque, isso graças em grande parte ao uso do conhecimento da estrutura física do documento.

5.3.3 Detecção dos separadores no cabeçalho do cheque

No cabeçalho do cheque bancário brasileiro, os blocos podem ser delimitados por linhas verticais ou horizontais (Figura 5.17). Portanto, a primeira dificuldade encontrada é descobrir que tipo de separadores o cabeçalho do cheque contém.

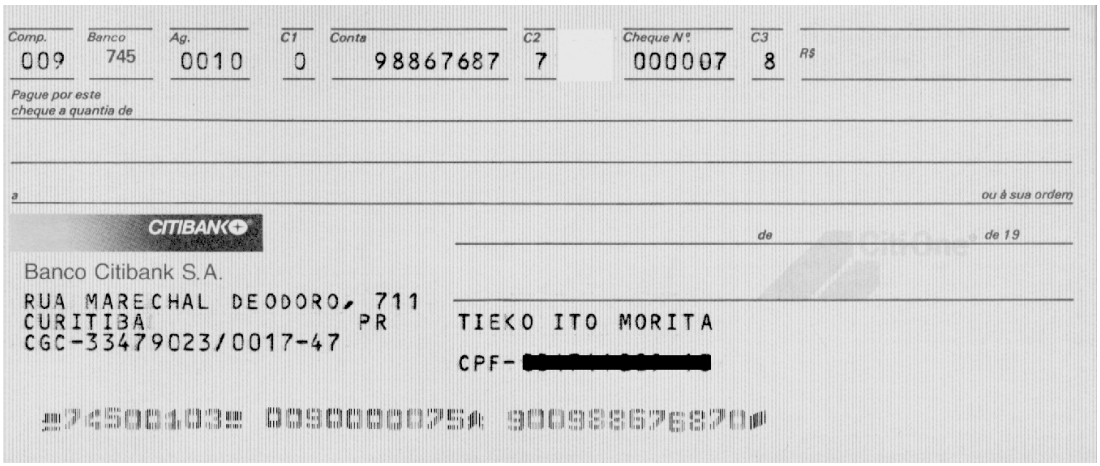

Figura 5.17: Cheque com delimitadores horizontais.

Resolvemos isso contando os resíduos de uma erosão vertical que visa preservar as linhas verticais. Se os separadores forem verticais, a erosão não elimina

tudo e o resultado dessa contagem é diferente de zero. Caso contrário (separadores horizontais), o resultado dessa contagem é zero.

A detecção automática dos separadores verticais faz-se por erosão agressiva com elemento estruturante vertical, de maneira a eliminar tudo que não é vertical, seguida de uma grande dilatação com o mesmo elemento estruturante de maneira a gerar separadores verticais de grande tamanho.

No caso da detecção de separadores demarcados por linhas horizontais, precisamos deduzir as linhas verticais a partir das horizontais. Visando isso, executamos o processo agressivo de erosão seguida da reconstrução, as duas com elemento estruturante horizontal. Obtemos somente linhas horizontais que são dilatadas com o elemento estruturante vertical de tal maneira a criar grandes blocos. Os lados verticais de cada bloco constituem os separadores verticais. Podemos observar que a abordagem utilizada para esta detecção necessita de poucos parâmetros heurísticos.

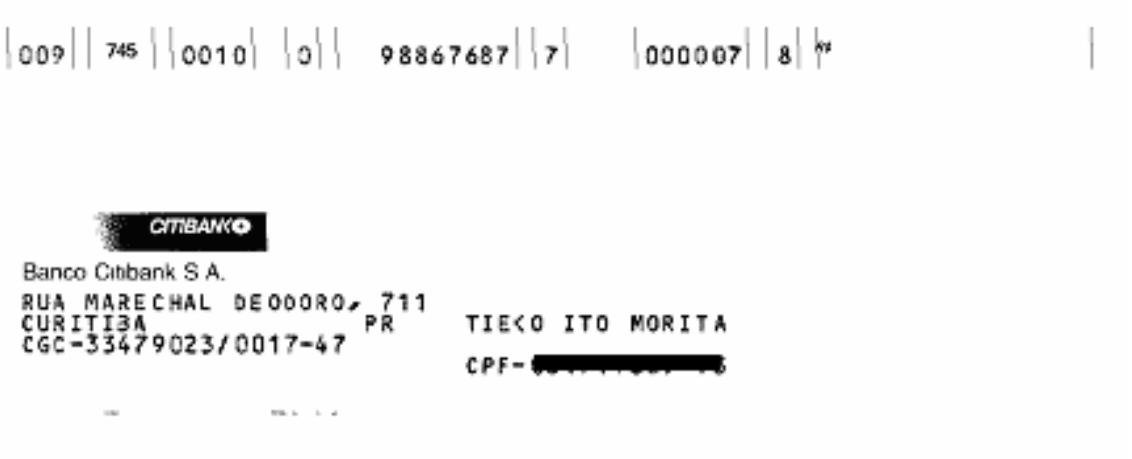

Figura 5.18: Delimitadores verticais obtidos a partir de delimitadores horizontais.

5.3.4 Marcação dos blocos

De posse dos separadores dos campos no cabeçalho do cheque, o próximo passo é criar os blocos. O nosso objetivo nesta parte da segmentação é marcar todos os blocos que deverão ser reconhecidos. A proposta é que cada bloco seja uma sub-imagem que será passada ao processo de rotulação e reconhecimento. Para isso, aplicamos a técnica *Top-Down* de RLSA (Run-Length Smoothing Algorithm) [WON82] que apresentamos no Capítulo II, baseado em processos morfológicos. Aplicamos as seguintes operações para realizar esta parte da segmentação:

- Dilatação binária da imagem 9 (Figura 5.11) com um elemento estruturante linha horizontal e 10 iterações, gerando a imagem número 10 (Figura 5.19).

- Fechamento binário da imagem 10 (Figura 5.19) com um elemento estruturante quadrado e 3 iterações, gerando a imagem número 11 (Figura 5.20).

- Dilatação binária da imagem 9 (Figura 5.11) com um elemento estruturante linha vertical e 10 iterações, gerando a imagem número 12 (Figura 5.21).

- Fechamento binário da imagem 12 (Figura 5.21) com um elemento estruturante quadrado e 3 iterações, gerando a imagem número 13 (Figura 5.22).

- Operação lógica AND entre as imagens 11 (Figura 5.20) e 13 (Figura 5.22) gerando a imagem 14 (Figura 5.23).

- Dilatação binária da imagem 14 (Figura 5.23) com um elemento estruturante quadrado e 5 iterações, gerando a imagem número 15 (Figura 5.24).

- Dilatação binária da imagem 16 (Figura 5.24) com um elemento estruturante linha vertical e 6 iterações, gerando a imagem número 16 (Figura 5.25).

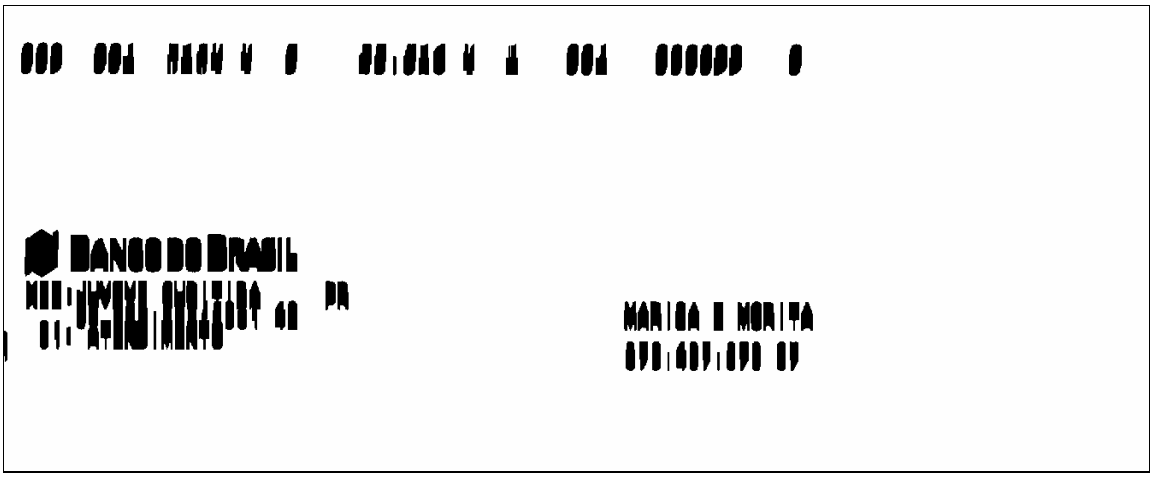

Figura 5.19: Dilatação binária com elemento estruturante linha horizontal.

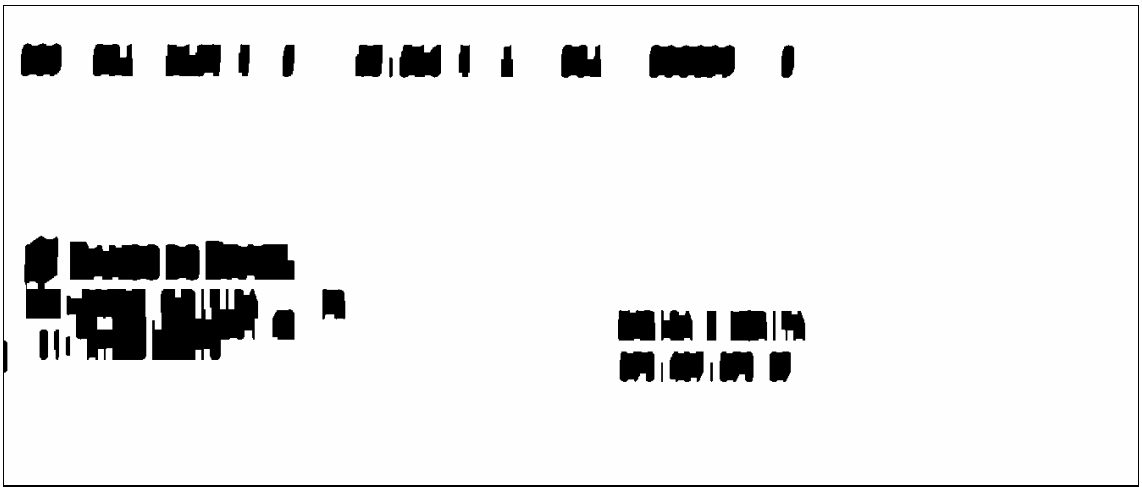

Figura 5.20: Fechamento binário da imagem da Figura 5.19.

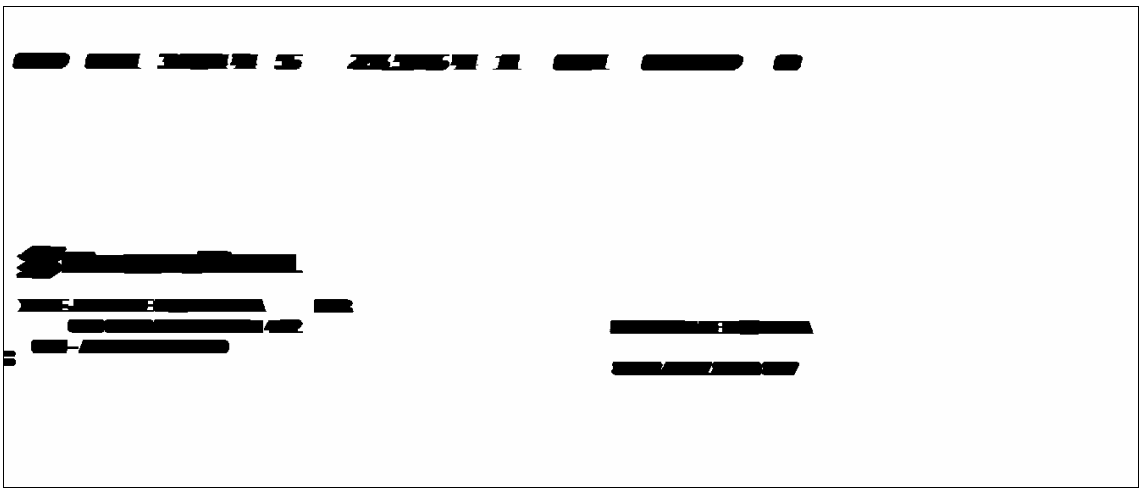

Figura 5.21: Dilatação binária com elemento estruturante linha vertical.

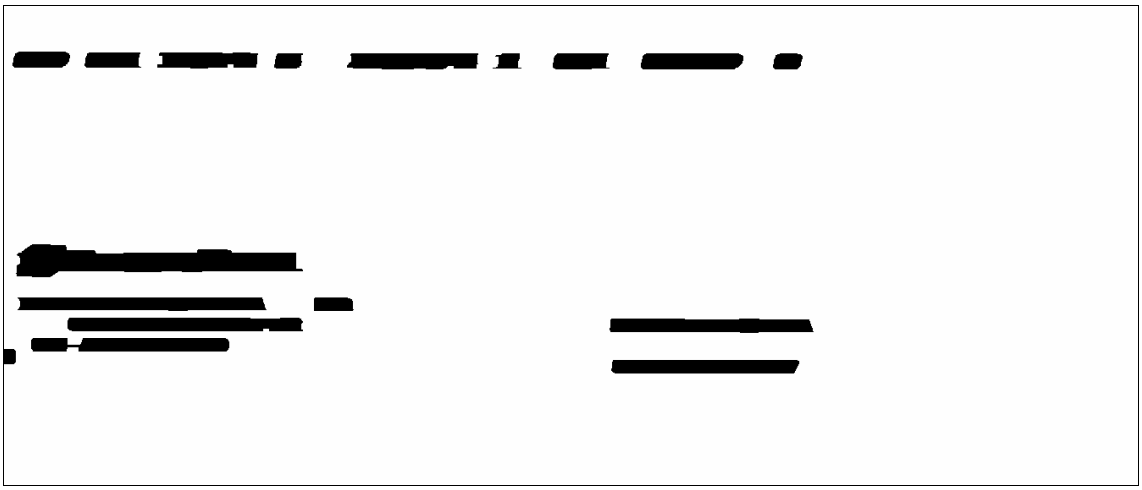

Figura 5.22: Fechamento binário da imagem da Figura 5.21.

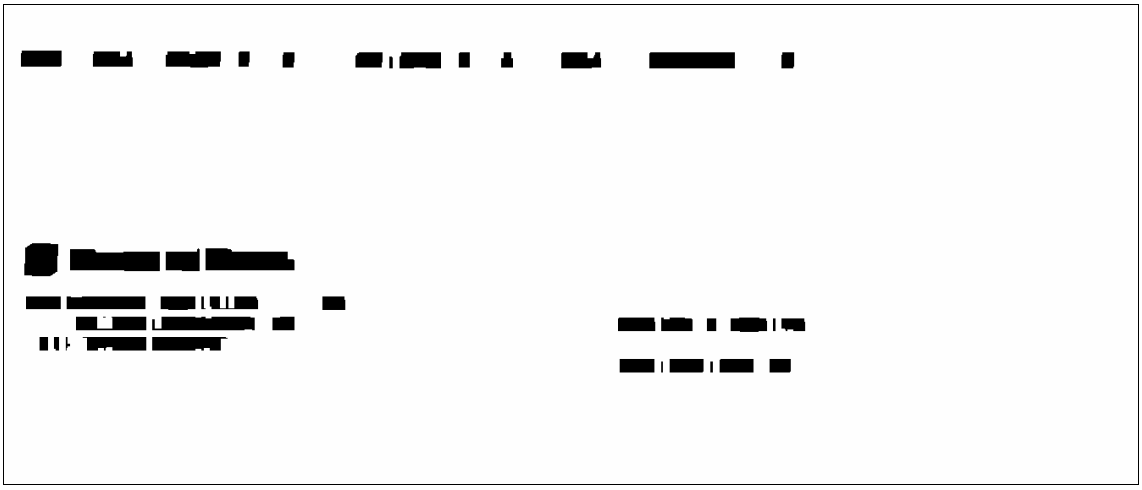

Figura 5.23: Operação lógica AND entre as imagens das Figuras 5.22 e 5.20.

Figura 5.24: Dilatação com elemento estruturante quadrado na imagem da Figura 5.23.

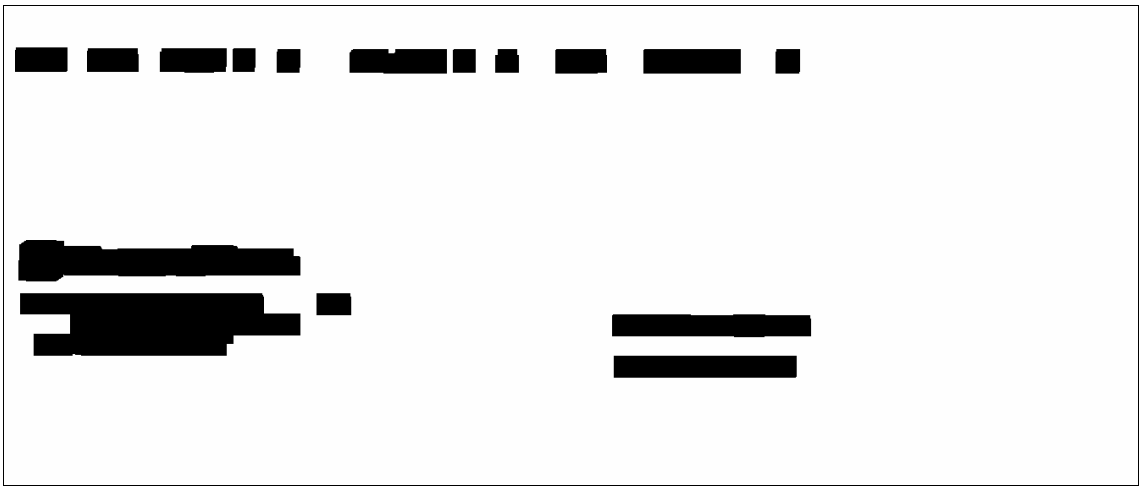

Figura 5.25: Dilatação com elemento estruturante linha vertical.

Após a execução dos processos acima descritos, conseguimos blocos uniformes, porém muitas vezes pode ocorrer a união desses blocos. Usamos os separadores verticais obtidos anteriormente para separar perfeitamente todos os blocos. Fazemos a diferença entre a imagem dos blocos demarcados e a imagem dos separadores.

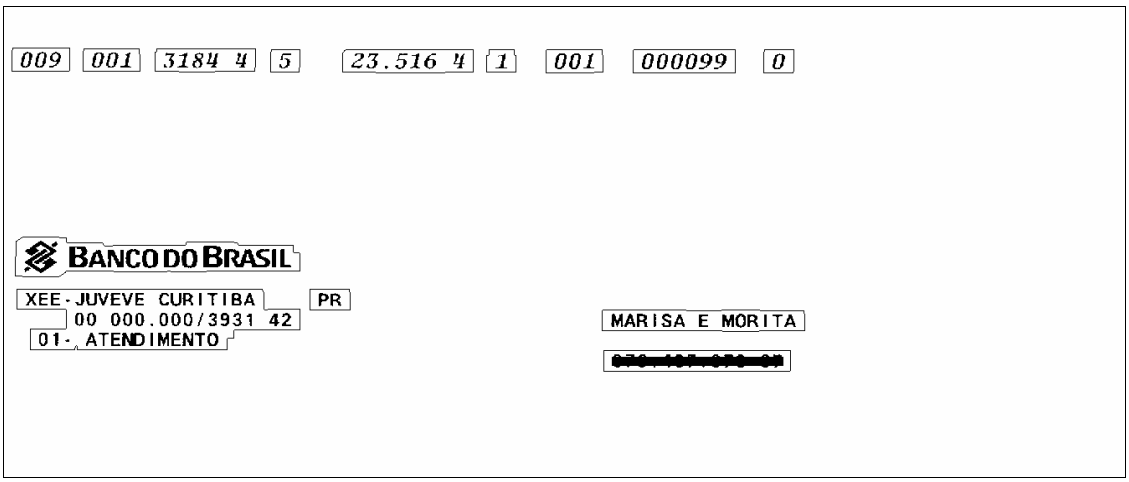

Figura 5.26: Blocos marcados para a fase de reconhecimento.

Na Figura 5.26 temos todos os blocos que deverão ser reconhecidos e então codificados, gerando a nossa estrutura lógica.

5.4 Reconhecimento

Como citamos anteriormente, cada bloco marcado deve ser reconhecido. Para executar essa tarefa usamos o software *WordScan* versão 3.1 da *Calera Recognition Systems.* Gostaríamos de salientar que não faz parte do escopo do nosso trabalho o desenvolvimento de um módulo de reconhecimento tendo em vista inúmeros softwares disponíveis no mercado bem como o estado avançado das pesquisas nesta área.

Como temos um grande conhecimento da estrutura do cheque bancário, é interessante usarmos isso nesta fase do processamento. Para cada bloco passado ao *software* de reconhecimento, temos condições visuais de saber se o resultado está parcialmente correto ou não. Dizemos parcialmente, pois é conhecido que o primeiro bloco a ser reconhecido é o número da câmara da compensação, consequentemente o retorno deverá ser composto de números somente. É importante salientar que seria necessário somente reconhecermos a primeira faixa de informações do cheque, e que com base nela e em uma base de dados conseguiríamos as informações restantes. Por exemplo, de posse da base de dados de um banco, o reconhecimento do número da conta corrente nos fornece todos os dados restantes do cliente tais como nome, cpf/cgc, endereço da agência entre outros, aumentado ainda mais o grau de confiabilidade do processo.

Aplicamos o processo da análise local no cheque da Figura 5.2. Com base no conhecimento que temos da estrutura do documento, passamos os principais blocos para o software de reconhecimento e obtemos os seguintes resultados:

```
018
275
0705
3
2
9 711259
3
B
010049
8
JD CUANABARA CAMPINAS SP
AVENIDA BRASIL 800
```

```
PAULO DE ARRUDA BORELLI
CPF XXXXXXXXX XX
```

Com base neste resultado e no conhecimento da estrutura, o próximo passo é a codificação da estrutura lógica do documento.

5.5 Codificação SGML

Mostraremos agora como as informações retornadas pelo módulo de reconhecimento podem ser armazenadas em uma estrutura lógica utilizando SGML.

Basicamente, as informações que deverão ser codificadas se encontram na primeira e terceira faixas, como descrevemos no Capítulo III – Estrutura do Cheque Bancário.

Conforme vimos no Capítulo IV, devemos elaborar as regras para o nosso documento, ou seja, o formulário do cheque bancário brasileiro. O conjunto de regras é a seguinte:

- Um cheque é composto de cabeçalho e corpo
  - cabeçalho do cheque é composto pelos seguintes campos:
  - Comp (Compensação)
  - Banco (Número do Banco)
  - Agenc (Número da Agência)
  - Dv (Dígito Verificador – Facultativo)
  - C1 (Dígito Verificador)
  - Conta (Número da Conta)
  - C2 (Dígito Verificador da Conta)
  - Rserie (Registro da Série – Facultativo)
  - Numer (Número do Cheque)
  - C3 (Digito Verificador do Número do cheque)
- corpo do cheque é composto dos seguintes campos
  - Nomeag (Nome da Agência)

- Endag (Endereço da Agência)
- Cliente (Nome do Cliente)
- Cpfcgc (Documento Cpf ou Cgc)

A descrição mais detalhada dos campos que compõe o formulário do cheque está descrita no Capítulo III – Estrutura do Cheque Bancário.

5.5.1 O DTD para o cheque

Com base nas regras acima colocadas, o próximo passo é escrever essas regras de uma maneira formal. A maneira de formalizar as regras em SGML, é através do DTD (*Document Type Definition).* Abaixo mostramos o DTD para o formulário de cheque brasileiro.

```
<!-- Document Type Definition for Bank Check -->
<!ELEMENT check   - -  (cabec, corpo)>
<!ELEMENT cabec   - -  (comp, banco, agenc, dv?, c1,
                        conta, c2, rserie?, serie, numer, c3)>
<!ELEMENT comp    O -  (#PCDATA)>
<!ELEMENT banco   O -  (#PCDATA)>
<!ELEMENT agenc   O -  (#PCDATA)>
<!ELEMENT dv      O O  (#PCDATA)>
<!ELEMENT c1      O -  (#PCDATA)>
<!ELEMENT conta   O -  (#PCDATA)>
<!ELEMENT c2      O -  (#PCDATA)>
<!ELEMENT rserie  O O  (#PCDATA)>
<!ELEMENT serie   O -  (#PCDATA)>
<!ELEMENT numer   O -  (#PCDATA)>
<!ELEMENT c3      O -  (#PCDATA)>
<!ELEMENT corpo   - -  (nomeag,endag,cliente,cpfcgc)>
<!ELEMENT nomeag  O -  (#PCDATA)>
<!ELEMENT endag   O -  (#PCDATA)>
<!ELEMENT cliente O -  (#PCDATA)>
<!ELEMENT cpfcgc  O -  (#PCDATA)>
```

Como pode-se notar, todos os elementos componentes da estrutura dos cheque, são do tipo PCDATA, exceto os elementos *check, cabec* e *corpo*, que são elementos composto por outros elementos. Também está explícito nessas declarações que todos os elementos necessitam de um *end-tag*, menos os elementos opcionais, que são o *dv* e o *rserie* (seguidos do ponto de interrogação), e ainda que os únicos elementos que tem necessidade do *start-tag* são os elementos compostos.

5.5.2 Elementos do Cheque Bancário Escritos em SGML

Uma vez construído o DTD para a estrutura do cheque, podemos escrever o texto em SGML para demonstrar a aplicabilidade do DTD acima definido. Respeitando as regras até então colocadas, o texto codificado é o seguinte:

```
<!doctype check SYSTEM>
<check>
   <cabec>
      018</comp>
      275</banco>
      005</agenc>
      3
      2</c1>
      9711259</conta>
      3</c2>
      B</serie>
      010049</numer>
      8</c3>
   </cabec>
   <corpo>
      JD CUANABARA CAMPINAS SP</nomeag>
      AVENIDA BRASIL 800</endag>
      PAULO DE ARRUDA BORELLI</cliente>
      XXXXXXXXX-XX</cpfcgc>
   </corpo>
</check>
```

Este código foi validado pelo analisador SGML VM2.EXE (Tabela 5.1, página 117), gerando os seguintes resultados:

```
VM105-> Pass 1 of 1.
VM100-> Entity is *SGMLDOC in D:\SGML\CHECK.SGM.
VM115-> 001 021: Document type definition started for CHECK.
VM116-> 002 000: Document type definition completed for CHECK; 0
SHORTREF maps defined.
VM111-> 002 007: CHECK start-tag found.
VM111-> 003 010: CABEC start-tag found.
VM111-> 004 009: COMP start-tag implied by data.
VM110-> 004 009: 3 data characters: [018]
VM112-> 004 016: COMP end-tag found; CABEC element resumed.
VM111-> 005 009: BANCO start-tag implied by data.
VM110-> 005 009: 3 data characters: [275]
VM112-> 005 017: BANCO end-tag found; CABEC element resumed.
VM111-> 006 009: AGENC start-tag implied by data.
VM110-> 006 009: 3 data characters: [005]
VM112-> 006 017: AGENC end-tag found; CABEC element resumed.
VM111-> 007 007: C1 start-tag implied by data.
```

```
VM110-> 007 007: 1 data character: [3]
VM114-> 008 006: Record end.
VM110-> 008 006: 6 data characters: [      ]
VM110-> 008 007: 1 data character: [2]
VM112-> 008 012: C1 end-tag found; CABEC element resumed.
VM111-> 009 013: CONTA start-tag implied by data.
VM110-> 009 013: 7 data characters: [9711259]
VM112-> 009 021: CONTA end-tag found; CABEC element resumed.
VM111-> 010 007: C2 start-tag implied by data.
VM110-> 010 007: 1 data character: [3]
VM112-> 010 012: C2 end-tag found; CABEC element resumed.
VM111-> 011 007: SERIE start-tag implied by data.
VM110-> 011 007: 1 data character: [B]
VM112-> 011 015: SERIE end-tag found; CABEC element resumed.
VM111-> 012 012: NUMER start-tag implied by data.
VM110-> 012 012: 6 data characters: [010049]
VM112-> 012 020: NUMER end-tag found; CABEC element resumed.
VM111-> 013 007: C3 start-tag implied by data.
VM110-> 013 007: 1 data character: [8]
VM112-> 013 012: C3 end-tag found; CABEC element resumed.
VM112-> 014 011: CABEC end-tag found; CHECK element resumed.
VM111-> 015 010: CORPO start-tag found.
VM111-> 016 030: NOMEAG start-tag implied by data.
VM110-> 016 030: 24 data characters: [JD CUANABA] ...
VM112-> 016 039: NOMEAG end-tag found; CORPO element resumed.
VM111-> 017 024: ENDAG start-tag implied by data.
VM110-> 017 024: 18 data characters: [AVENIDA BR] ...
VM112-> 017 032: ENDAG end-tag found; CORPO element resumed.
VM111-> 018 029: CLIENTE start-tag implied by data.
VM110-> 018 029: 23 data characters: [PAULO DE A] ...
VM112-> 018 039: CLIENTE end-tag found; CORPO element resumed.
VM111-> 019 015: CPFCGC start-tag implied by data.
VM110-> 019 015: 9 data characters: [XXXXXXXXX]
VM110-> 019 018: 3 data characters: [-84]
VM112-> 019 027: CPFCGC end-tag found; CORPO element resumed.
VM112-> 020 011: CORPO end-tag found; CHECK element resumed.
VM112-> 021 008: CHECK end-tag found; *DOC element resumed.
VM106-> Processing completed; 0 messages issued.
           1 entity declared with 0 characters of text.
          18 element types declared,
             with 50 model tokens and 0 exception groups.
           0 attributes with 0 group members
             and 0 characters of value text.
           0 IDs and 0 ID references specified.
           0 data content notations with 0 text characters.
           0 short reference maps declared.
         552 capacity points required (1% of 35000 permitted).
```

É importante salientar, que um texto SGML sempre estará acompanhado do seu DTD, caso contrário o analisador rejeitará o código por falta de parâmetros para a compilação. Este analisador requer ainda que o arquivo SGML.MSG esteja no

mesmo diretório que o arquivo do DTD. Resumidamente, para a compilação do código SGML são necessários os seguintes arquivos:

Tabela 5.1: Arquivos necessários para compilar um código SGML.

ARQUIVO	DESCRIÇÃO
	SGML.
G	e contém as mensagens geradas pelo analisador.
TD	eque bancário.
GM	SGML.

O analisador SGML VM2.EXE pode ser livremente acessado, inclusive para operações de *download*, através do seguinte endereço na World Wide Web: **http://www.sgmlu.com**

5.6 Codificação ODA

Mostraremos agora a codificação do cheque bancário utilizando o modelo ODA. Para isso usaremos as mesmas regras especificadas para o modelo SGML por questões de compatibilidade entre os modelos.

Como vimos no Capítulo IV, o modelo ODA faz uma clara distinção entre as estruturas (física e lógica) e o conteúdo do documento. Abaixo na Tabela 5.2 vamos definir as estruturas lógica e física e a inter-relação entre essas duas estruturas através do conteúdo do documento.

Tabela 5.2: Modelo lógico para ao cheque bancário.

Classe		Descrição
`Check`	`::=`	`SEQUENCE(cabec, corpo)`
`Cabec`	`::=`	`SEQUENCE(comp, banco, agenc, dv(OPTIONAL), c1, conta, c2, rserie(OPTIONAL), série, numer, c3)`
`Corpo`	`::=`	`SEQUENCE(logo, nomeag, endag, cliente, cpf/cgc)`

A Tabela 5.2 especifica três classes de objetos: *Check, cabec* e *corpo.* O *Check* é composto por uma seqüência de *cabec* e *corpo.* O *cabec* é composto por vários atributos dos quais dois deles são opcionais, e o *corpo* é composto por uma outra série de atributos, sendo que todos são obrigatórios. Podemos fazer uma analogia da estrutura acima definida com a Figura 5.27, onde as informações do cheque estão logicamente distribuídas dentro de um formulário de cheque genérico.

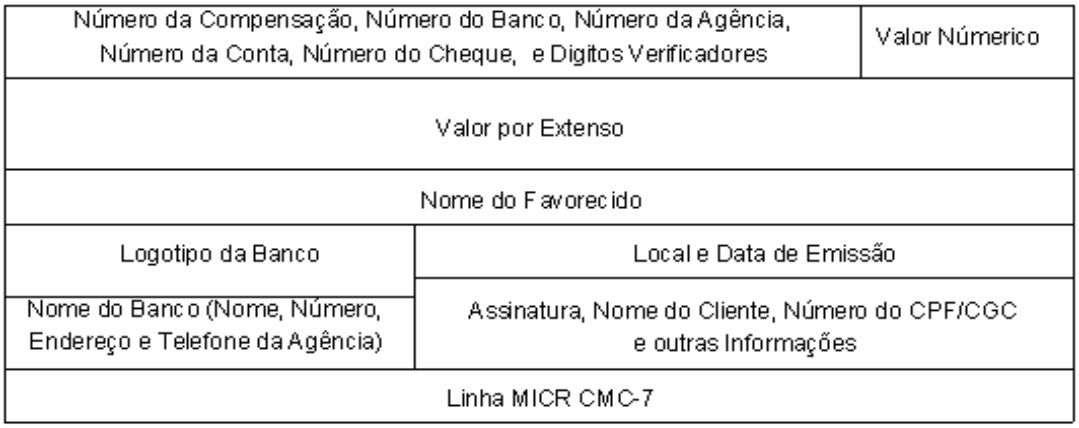

Figura 5.27: Modelo para a distribuição de informações em cheques bancários brasileiros.

Similar a estrutura lógica, definimos na Tabela 5.3 a estrutura física para o cheque bancário.

Tabela 5.3: Modelo físico para o cheque bancário.

Classe		Descrição
`Check-form`	`::=`	`SEQUENCE(cabec-frame, corpo-frame)`
`cabec-frame`	`::=`	`SEQUENCE(comp-block, banco-block, agenc-block, dv-block(OPTIONAL), c1-block, conta-block, c2-block, rserie-block(OPTIONAL), série-block, numer-block, c3-block)`
`corpo-frame`	`::=`	`SEQUENCE(logo-block, nomeag-block, endag-block, cliente-block, cpf/cgc-block)`

Como podemos notar, o modelo de distribuição proposto na Figura 5.27 possui mais informações do que a estrutura lógica apresentada. Porém, vale ressaltar que foram codificadas as informações buscadas no processo de segmentação e marcação de blocos descritos neste capítulo.

Da mesma forma que fizemos a analogia na estrutura lógica, podemos fazer uma analogia para a estrutura física, porém com a Figura 5.28, a qual nos mostra uma divisão física dos campos do cheque bancário.

Na estrutura física proposta são definidas três grandes classes de objetos: *Ckeck-form, cabec-frame* e *corpo-frame*. O *Check-form* é o formulário do cheque propriamente dito e é composto de duas regiões. O *cabec-frame* é a primeira região e é composta pelos blocos na parte superior do formulário do cheque e o *corpo-frame* é a segunda região e é composta pelos blocos na parte do meio do cheque. A região destinada a magnetização não foi contemplada na nossa estrutura pois a mesma é descartada no momento da segmentação.

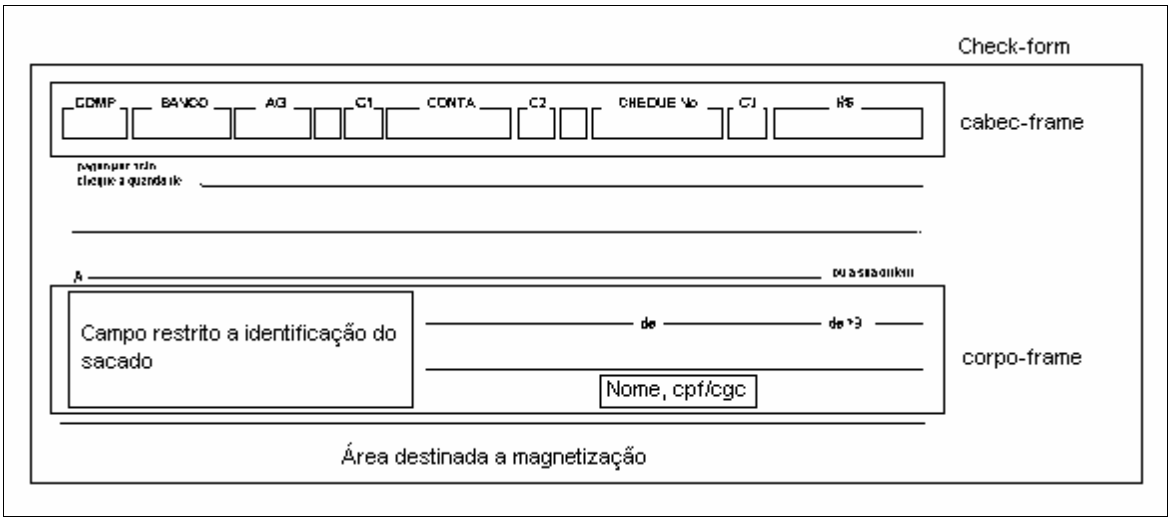

Figura 5.28: Distribuição física dos campos no formulário do cheque.

A Figura 5.28 mostra que fisicamente o menor componente da estrutura física é o bloco. Não poderia ser diferente, pois a definição do modelo ODA para bloco é a seguinte: O bloco é uma classe de objetos básica e é representado na forma de uma superfície retangular, paralelo as bordas da página, ou formulário no nosso caso.

Até então, definimos as estruturas lógica e física para o cheque bancário, porém o conteúdo do documento não foi mencionado. Isso acontece porque o modelo ODA faz uma clara distinção entre as estruturas lógica e física do documento e o seu conteúdo. O conteúdo é descrito pelo modelo ODA em uma estrutura chamada de *Arquitetura do Conteúdo*. Na Figura 5.29 é possível verificar como o conteúdo do documento permite fazer a inter-relação entre as estruturas lógica e física.

Como podemos verificar, o relacionamento das estruturas lógica e física se dá através do conteúdo do documento. Através do diagrama mostrado na Figura 5.29 podemos tirar várias conclusões como por exemplo: O correspondente lógico de um bloco físico, o conteúdo de um determinado bloco físico e assim por diante.

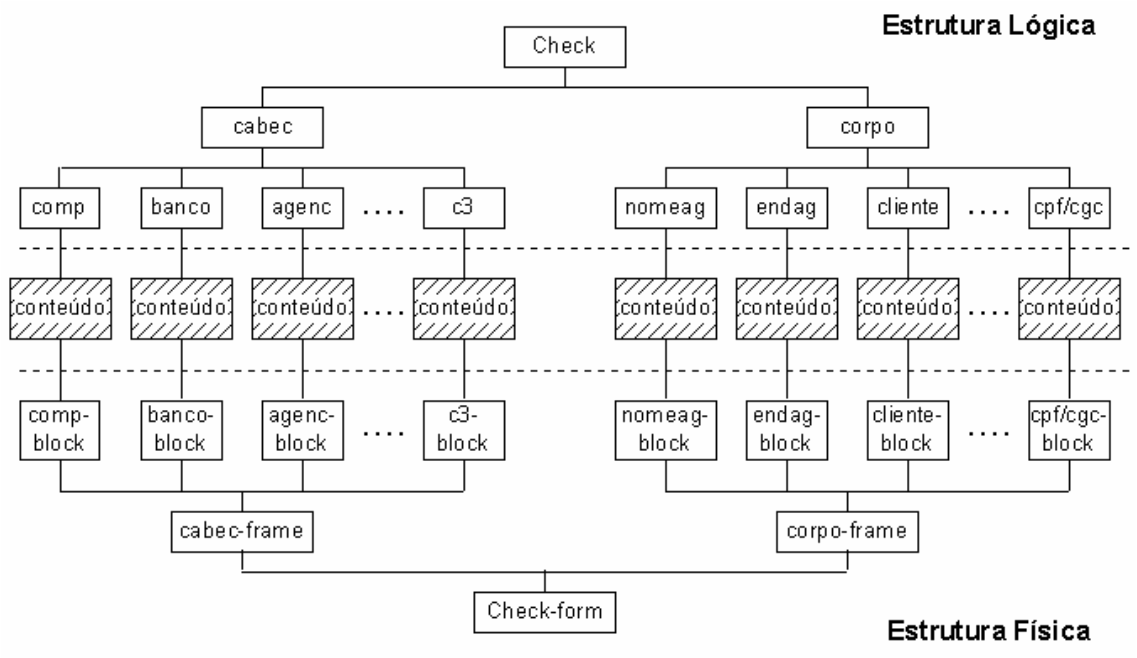

Figura 5.29: Inter-relacionamento entre as estruturas lógica e física.

5.7 Neste Capítulo

Neste capítulo vimos o processamento necessário para se obter as estruturas lógica e física a partir de uma imagem em tons de cinza. Dentro deste processamento, destacamos a segmentação, que tem o objetivo de fornecer uma

imagem binária de boa qualidade, além de marcar os blocos para o reconhecimento e a codificação.

Vimos como codificar as informações que retornam do módulo de reconhecimento nos padrões SGML e ODA. Dentro do modelo ODA, vimos as distinções entre as estruturas física e lógica bem como a comunicação entre essas estruturas utilizando o conteúdo do arquivo.

No capítulo seguinte apresentaremos algumas discussões sobre o trabalho desenvolvido, e também algumas considerações sobre os resultados encontrados.

Capítulo VI
Análise e Discussão dos Resultados

Os trabalhos em Análise e Reconhecimento de Documentos, nos mostram a necessidade de dispor de um modelo de estruturas de documentos para guiar o reconhecimento e melhorar os resultados em um sistema de análise [DEN88][BEL89].

Com o intuito de diminuir a proliferação de modelos para a representação de estruturas de documentos, os órgãos internacionais de normalização e as indústrias fizeram esforços comuns afim de propor padrões para representação de documentos. Os principais resultados foram conseguidos pela ISO (*International Standard Organization),* através das normas ISO 8613 e ISO 8879, respectivamente ODA e SGML.

Dentro do domínio de escritórios, a questão colocada era a seguinte: como fazer a troca automática de documentos (memorandos, relatórios, cartas, etc..) entre diferentes sistemas de uma mesma empresa, ou entre empresas diferentes. O principal problema, no entanto, era a incompatibilidade entre o *hardware* e os sistemas operacionais utilizados dentro dos escritórios. Para resolver este problema, a norma ODA (*Open/Office Document Architecture)* foi proposta. Neste trabalho de pesquisa, a principal dificuldade no estudo do modelo ODA, foi a falta de publicações sobre o assunto. Por exemplo: A tentativa de adquirir um livro através da internet chamado "*Document Architecture in Open Systems, The ODA Standard*", porém, após 4 meses de espera, fomos informados que o título em questão não era mais publicado. Outro fato interessante: a Microsoft possuía um produto chamado "*WinWord 6.0 ODA (FOD-26) Converter Kit* ", disponível para plataforma Windows e Windows NT. Este produto habilitava o Microsoft Word a ler e gravar arquivos no formato ODA, porém este produto deixou de ser distribuído pela Microsoft.

Kirstein em [KIR96] mostrou que as implementações utilizando ODA não estão sendo buscadas de forma enérgica pelos fabricantes. Os produtos não tem uma

funcionalidade adequada para atender os usuários, e existe uma grande probabilidade de que isso não mude.

Kohsari em [KOH96] faz referência ao "*Bull Office Team View*". Este software foi desenvolvido pela ABC-BULL para permitir acessar, compartilhar e gerenciar documentos, além de suportar documentos ODA. Quando entramos em contado com a ABC-BULL a fim de adquirir uma cópia deste software, fomos informados que a empresa acabara de substituí-lo pelo Lotus Notes, porém o Lotus Notes não manipula arquivos ODA.

O que podemos deduzir baseado na literatura encontrada, é que o modelo ODA teve um avanço considerável logo após a sua concepção, e que a falta de estímulo quanto ao seu uso se deve em grande parte a complexidade imposta pelo modelo.

Por outro lado em [TRA95] podemos observar que sistemas baseados em SGML estão sendo implementados em larga escala, e em diferentes tipos de organizações. Desde a aprovação formal através da ISO 8879 em 1986, os desenvolvedores e conhecedores de SGML tem crescido. As ferramentas que manipulam SGML estão multiplicado-se e tornando-se cada vez mais sofisticadas, bem como as aplicações desenvolvidas em SGML.

Podemos constatar o grande crescimento do SGML, através do grande número de publicações, e também da variedade de informações a respeito de SGML disponíveis na Internet, informações essas que vão desde introdução ao SGML até oportunidades de emprego para profissionais em SGML.

No estudo que mostramos no capítulo IV, podemos constatar que o modelo SGML é bem mais simples de ser implementado do que o modelo ODA, o que torna o SGML bem mais atrativo aos desenvolvedores.

É importante salientar que a norma ODA é mais completa do que o SGML, pois a mesma é capaz de gerar as estruturas física e lógica do documento, o que a

norma SGML não é capaz de fazer. Para gerar a estrutura física, o SGML necessita de outros padrões tais como DSSSL (*Document Style Semantics and Specification Language*) – ISO 10179.2 e SPDL (*Standard Page Description Language*) – ISO 10190.

Como o objetivo do nosso trabalho é codificar o conteúdo do cheque bancário brasileiro, usando para isso as estruturas física e lógica, acreditamos que o modelo SGML é o mais indicado, pois é capaz de armazenar todo conteúdo do cheque bancário, além de fornecer mecanismos de indexação muito satisfatórios [TRA95]. Para codificação da estrutura física do documento, levando em consideração que optamos pelo uso do SGML, o DSSSL e o SPDL são os padrões que deverão ser pesquisados futuramente.

Para que possamos recuperar o conteúdo do cheque bancário brasileiro, utilizamos técnicas de Processamento Digital de Imagens, como descrevemos no capítulo V. O processamento desenvolvido neste trabalho é uma mostra do que já é possível se fazer com a imagem de qualquer tipo de documento.

Nesse capítulo faremos uma análise em relação aos métodos propostos e discutiremos os problemas e as dificuldades que foram encontradas no desenvolvimento desses métodos. Esses foram implementados utilizando a ferramenta Pdimagem desenvolvido por Britto em [BRI95]. Para a implementação dos mesmos foram utilizados a linguagem C++ (Visual C++ 4.0) em ambiente *Windows* 95.

O primeiro passo, foi buscar técnicas que nos permitissem eliminar os padrões do fundo do cheque, porém sem perder informações relevantes ao nosso processamento. Encontramos a solução apresentada em [SAN97], a qual nos forneceu resultados satisfatórios, entretanto, não conseguimos definir um número adequado de iterações para o processo de equalização através da hiperbolização quadrática do histograma. Sendo assim, ficou inviável um processo de limiarização totalmente automático. Nas Figuras 6.1 e 6.2 podemos visualizar o resultado do processo de limiarização após a equalização através da hiperbolização quadrática do histograma.

Figura 6.1: Imagem limiarizada após 1 iteração da equalização.

Figura 6.2: Imagem limiarizada após 2 iterações da equalização.

Como podemos notar na Figura 6.2, informações relevantes ao nosso processamento foram perdidas, isto se deve as 2 iterações aplicadas no processo de equalização. Por outro lado, algumas vezes uma única iteração não é suficiente para suavizar o fundo do cheque, fazendo com que parte do fundo permaneça na imagem limiarizada, o que torna o processamento inviável.

Constatamos também que, quando as imagens são bastante claras, o processo de equalização pela hiperbolização quadrática do histograma não nos traz benefícios.

Após o processo da limiarização, testamos dois métodos para a extração dos componentes do cheque bancário brasileiro. Chamamos esses métodos de análise global e análise local. Na análise global aplicamos as operações morfológicas no formulário do cheque como um todo. Constatamos que os resultados não foram satisfatórios, pois eram funcionais em uma determinada região do cheque, e em outras regiões não agia com eficácia. Tendo isso em vista, elaboramos um processamento local nas regiões do cheque o qual chamamos de análise local. Esse tipo de processamento é viável quando se tem conhecimento da estrutura do documento, conhecimento esse que nos permite separar o documento em regiões. Aplicamos esse método sobre a base de dados formada por Koerich em [KOE97], 250 cheques de 10 diferentes bancos brasileiros. No anexo A mostramos os resultados do processamento. É importante salientar que as imagens de cheques processadas nesse trabalho são livres de inclinação (*skew*). Santos em [SAN97] descreve algumas técnicas para correção de inclinação.

Conforme pudemos constatar no capítulo V, os resultados da análise local se mostraram bem mais eficientes, porém consomem mais tempo de processamento (em torno de 15%), pois é necessário despender processamento para separar a imagem em sub-imagens com o intuito de processá-las separadamente, e então agrupá-las para o passo da marcação de blocos.

Na Tabela 6.1 podemos comparar os resultados obtidos através da análise global e da análise local. Utilizamos dois critérios para concluir se o resultado do processamento foi satisfatório:

- Os blocos marcados devem conter somente as informações que devem ser codificadas, ou seja, o conteúdo dos campos e não os nomes dos campos;
- O resultado do reconhecimento deve apresentar um índice de 90% ou mais de sucesso por cheque processado.

É importante salientar que os resultados obtidos dependem fortemente do *software* de reconhecimento utilizado.

Tabela 6.1: Comparativo entre as análises global e local.

Banco	Número de Imagens	Percentagem de Acerto da Análise Global	Percentagem de Acerto da Resultado da Análise Local
Banco do Brasil	20	0%	90%
Banespa	25	0%	0%
Banestado	28	100%	100%
BCN	30	0%	93%
Caixa Econômica Federal	28	0%	89%
Citibank	13	0%	100%
HSBC Bamerindus	31	0%	85%
Itaú	25	0%	100%
Real	25	0%	100%
Rural	25	100%	100%
Total	250	21,2%	85,7%

Para a marcação de blocos, utilizamos uma metodologia *top-down*, a qual apoia-se num conhecimento muito forte da estrutura do documento a ser analisado. Pesquisamos os métodos *bottom-up*, e constatamos que o documento necessita um nível de detalhe maior do que os métodos *top-down,* exigindo maior resolução e definição da imagem. Todo o processamento empregado na fase de marcação de blocos teve o intuito de criar blocos retangulares, com o propósito de facilitar a extração dos mesmos para a fase de reconhecimento.

Tentamos não nos prender a limiares heurísticos durante o processamento, porém em determinados casos, o número de iterações para as operações morfológicas foi determinado heurísticamente através de uma quantidade massiva de testes.

Finalizando este capítulo, a avaliação dos resultados conseguidos através do processamento proposto tem a interação humana, apesar de que uma metodologia de avaliação e controle de qualidade das imagens resultantes traria benefícios a qualquer ambiente de processamento de cheques bancários.

Capítulo VII
Conclusão e Trabalhos Futuros

Com o objetivo de extrair e armazenar os elementos pré-impressos de imagens de cheques bancários brasileiros, efetuamos um estudo sobre padrões de documentos e técnicas de processamento digital de imagens que tornaram possível a execução desta tarefa. Analisando os resultados obtidos no processo de segmentação elaborado no Capítulo V deste trabalho, concluímos que a segmentação de documentos impressos apresentam grandes perspectivas futuras.

Conforme citamos em várias partes deste trabalho, o uso do conhecimento da estrutura do documento é de fundamental importância para um sistema de Análise e Reconhecimento de Documentos. A observação das disposições espaciais dos elementos componentes no documento é fundamental para uma caracterização clara do que deve ser buscado no documento analisado.

A remoção do padrão de fundo do cheque bancário brasileiro apresentou bons resultados através dos processos de suavização e limiarização, respectivamente, equalização pela hiperbolização quadrática do histograma e limiarização pelo método de Otsu. Porém ressaltamos que não conseguimos elaborar um processo totalmente automático nesta fase, devido a dificuldade em suavizar os padrões do fundo, conforme descrevemos no Capítulo VI.

Quanto aos métodos utilizados no processamento da imagem, o método que trata as diferentes faixas do cheque separadamente, o qual chamamos de análise local, mostrou-se mais eficiente e a sua aplicação viabilizada em função do conhecimento da estrutura do documento.

No tocante aos padrões e normas estudados neste trabalho, constatamos que o SGML tem se tornado um padrão de fato devido a sua grande aceitação na comunidade científica e no mercado. Concluímos que o SGML é suficiente para a codificação da estrutura lógica do cheque bancário brasileiro, além de fornecer bons

mecanismos de pesquisa. Podemos concluir também que o ODA fornece uma estrutura muito mais complexa que o SGML, e este é um dos motivos pelo qual a norma ODA não teve uma aceitação grande como o SGML.

O pré-requisito para uma boa marcação de blocos e por conseqüência um bom reconhecimento dos mesmos é uma imagem limiarizada. Acreditamos que muitos esforços deverão ser feitos na busca de soluções eficazes para a segmentação das imagens de cheques bancários brasileiros. Como trabalhos futuros que poderão dar seqüência a este, acreditamos que as áreas de pesquisas abaixo relacionadas são de extrema importância para um processo totalmente automático na busca das estruturas de uma imagem de cheque bancário brasileiro.

- Estudo aprofundado no problema da suavização do padrão de fundo do formulário do cheque;
- Elaboração de um método automático que mensure a qualidade alcançada pelo processo de segmentação em uma base de dados grande;
- Elaboração de um método que determine o número de iterações necessárias nas operações morfológicas, eliminando ao máximo o uso de limiares heurísticos no processamento. Um segmento da morfologia matemática que entendemos que possa trazer bons resultados é o da granulometria binária.

Acreditamos que esses pontos colocados são bastante complexos, porém necessários na busca de soluções efetivas para o problema da segmentação de imagens de cheques bancários.

Referências

[AKI95] Akindele T. O. “Vers un système de construction automatique de modèles génériques de structures de documents”. PhD Thesis. Université Henri Poincaré Nancy I. França. 1995.

[BAI90] Baird H., Jones S., Fortune S. “Image segmentation by shape-directed covers”. ICDAR´90. pp 820-825. 1990.

[BEL89] Belaid A., Chenevoy Y., Lâasri H. “Vers un système à base de connaissance pour la compréhension de l’écriture” Actes du 7. Congrès AFCET de Reconnaissance des Formes et Intelligence Artificielle. 1989.

[BCO83] Banco Central do Brasil, Manual de Normas e Instruções, 1983.

[BRI95] Britto Jr. A.S. “Segmentação por cor utilizando a Transformada de Karhunen-Loève” Dissertação de Mestrado, CEFET-Pr. Curitiba. 1995.

[BRO89] Brown H. “Standarts for structured documents”. The Computer Journal. pp 505-514. 1989.

[BUN84] Bunke H., Sagerer G. “Use and representation of knowledge in image understanding based on semantic networks”. Procs of 7th International Conference on Patterns Recognition”. Volume 2, pp 1135-1137. Montreal, Canada. 1984.

[CAR91] Carlis J.V., Drake J.M. “Whats’s a document?. A Data model based on ISO 8613”. University of Minnesota. 1991.

[COB96] Cobra D. T., “A generalization of the method of quadratic hyperbolization of image histograms”. The IEEE Circuits and Systems Society. 1996.

[DEN88] Dengel A., Barth G. “High level document analisys guided by geometric management systemd”. SPIE, Vol.1293 Aplications of Artificial Intelligence VIII. 1990.

Referências

[DOS84] Doster W. “Different states of a document’s content on the way from the Gutembergian world to the eletronic world”. Procs of 7th ICPR. pp 982-874. 1984.

[FAC96] Facon J. “Morfologia Matemática: teoria e exemplos”. Editora Universitária Champagnat. Curitiba. 1996.

[FUR89] Furuta R. “Concepts and models for structured documents”. Cambridge University Press. 1989.

[FOX92] Fox E. “ Building a user-centered database from the ACM literature”, Procs of Symposium on Document Analysis and Information Retrieval. 1992.

[GAR96] Gartland A., Houghton D. “Tool survey report in the context of the DMU-JEDI project”. The Joint Eletronic Document Interchange Project – Ukerna. 1996.

[GON92] Gonzalez R., Woods C., Richard E., “Digital Image Processing”. Addison-Wesley. 1992.

[HU94] Hu T. “New methods for robust and efficient recognition of the logical structures in documents”, PhD Thesis. Instituit d’Informatique. Universite de Fribourg. Suisse, 1994.

[ISO87] International Standart Organization. “Information processing. Open systems, Computer Graphics – Metafile for the storage and transfer of picture description information”. Parts 1-3, ISO 8632. 1987.

[ISO91] International Standart Organization. “Information processing. Text and office systems – Office Document Architecture and interchange formats” Part 10. ISO 8613. 1991.

Referências

[KOE97] Koerich, A. L., “Uma proposta para o processamento de imagens de cheques bancários: Extração e armazenamento de infromações“. Dissertação de Mestrado. Universidade Estadual de Campinas. 1997.

[KOH96] Kohsari G.M. “The current status of ODA and SGML Converters in the context of the UCL-JEDI Project”. The Joint Eletronic Document Interchange Project – Ukerna. 1996.

[KIR96] Kirstein P. “A Final Report of JEDI-Project”. The Joint Eletronic Document Interchange Project – Ukerna. 1996.

[McQ97] McQueen C. M. S., Burnard L. “Guidelines for eletronic text encoding and interchange”. 1997.

[NAG91] Nagy G., Seth S. “Hierarchical representation of optically scanned documents”. Proc of 7th ICPR IEEE CS Pres. Los Alamitos, California. pp 347-349. 1991.

[NIY94] Niyogi D. “A knowledge-based approach to deriving logical structure from document images”. PhD Thesis. University of Buffalo. New York. 1994.

[O’GO92] O’Gorman L. “The document spectrun for page layout analysis”. IAPR International Workshop on Structural and Syntatic Pattern Recognition”. Bern, Switzerland. 1992.

[O’HÁ96] O’Hare G.M.P., Jennings N.R., “Foundations of distribuited artificial intelligence”. Wiley-Interscience Publication New York, NY. 1996.

[PEE85] Peels A. J., Jansen N.J., Nawijn W. “Document architecture and text formating”. ACM Transactions on Office Information Systens. pp 347-369. 1985.

[PRA78] Pratt W. K. “Digital image processing”. John Wiley & Sons. New York. 1978.

Referências

[RAB92] Rabbani M. “Digital Image Enhancement”. SPIE-Video Short Course Notes. 1992.

[SAN97] Santos J.E.B. “Estudo sobre métodos e técnicas para a segmentação de imagens de cheques bancários”. Dissertação de Mestrado, CEFET-Pr. Curitiba. 1997.

[SER82] Serra J. “ Image Analysis and Mathematical Morphology”. Academic Press. London. 1982.

[SAH88] Sahoo, P.K., Soltani S., Wong A.K.C, “The survey of Thresholding techniques.” Computer Vision, Graphics and image Processsing. pp 233-260. 1988.

[SHA85] Shapiro L. G. “ The role of AI in computer vision”. Procs of 2nd IEEE Computer Society Conference on Artificial Intelligence Aplications. pp 76-81. 1985.

[SRI89] Srihari S.N., Govindaraju V. “Textual image analysis using the Hough Transform”. International Journal of Machine Vision and Aplications. pp 141-153. 1989.

[TAN91] Tang Y. Y., Suen C.Y., Yan C. D., Cheriet M. “Document analysis and understanding: A Brief Survey”, Procs of the 1st International Conference on Document Analysis and Recognition, Vol 2, St-Malo, France. 1991.

[TAY92] Taylor S., Lipshutz M., Weir C., “Document structure interpretation integrating multiple knowledge sources”, Procs of Symposium on Document Analysis and Information Retrieval. 1992.

[TRA95] Travis B., Waldt D. “The SGML Implementation Guide”, Springer-Verlag, Berlin Heidelberg. 1995.

[VIN93] Vincent L. “Morphological grayscale reconstruction in image analysis: Aplications and efficient algorithms”. IEEE Transactions on Image Processing. pp.176-201. 1993.

[WAT92] Watanabe T., Fukumura T., “A framework of layout recognition for document understanding”, Procs of Symposium on Document Analysis and Information Retrieval. 1992.

[WON82] Wong K., Casey R., Wahl F. “Document analysis system”, Procs of 6th International Conference on Pattern Recognition. pp 19-22. Munich. 1982.

Anexo A

Resultados do processamento através da análise local

Apresentamos nesse anexo alguns resultados obtidos através do processamento da análise local, proposto no Capítulo V.

A Figura 1 é a imagem original do cheque do Banco do Brasil e a Figura 2 é o resultado após o processamento da análise local. A Figura 3 é a imagem original do cheque do Banco do Real e a Figura 4 é o resultado após o processamento da análise local. A Figura 5 é a imagem original do cheque do BCN e a Figura 6 é o resultado após o processamento da análise local. A Figura 7 é a imagem original do cheque do Banco HSBC Bamerindus e a Figura 8 é o resultado após o processamento da análise local. A Figura 9 é a imagem original do cheque do Banco Itaú e a Figura 10 é o resultado após o processamento da análise local. A Figura 11 é a imagem original do cheque do Banco do Citibank e a Figura 12 é o resultado após o processamento da análise local. A Figura 13 é a imagem original do cheque do Banco Banespa e a Figura 14 é o resultado após o processamento da análise local. A Figura 15 é a imagem original do cheque da Caixa Econômica Federal e a Figura 16 é o resultado após o processamento da análise local. A Figura 17 é a imagem original do cheque do Banco Banestado e a Figura 18 é o resultado após o processamento da análise local. A Figura 19 é a imagem original do cheque do Banco Rural e a Figura 20 é o resultado após o processamento da análise local.

Figura 1: Imagem original Banco do Brasil

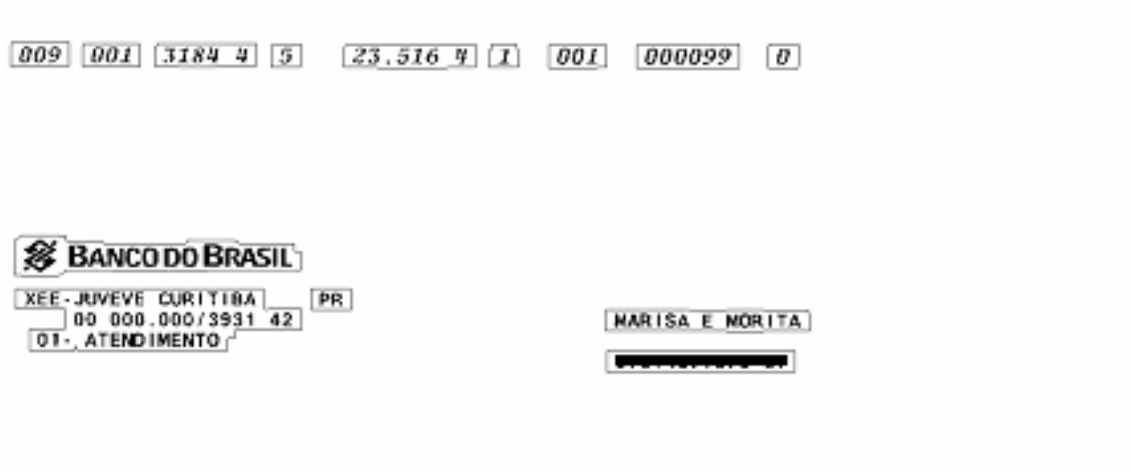

Figura 2: Resultado do processamento da Figura 1.

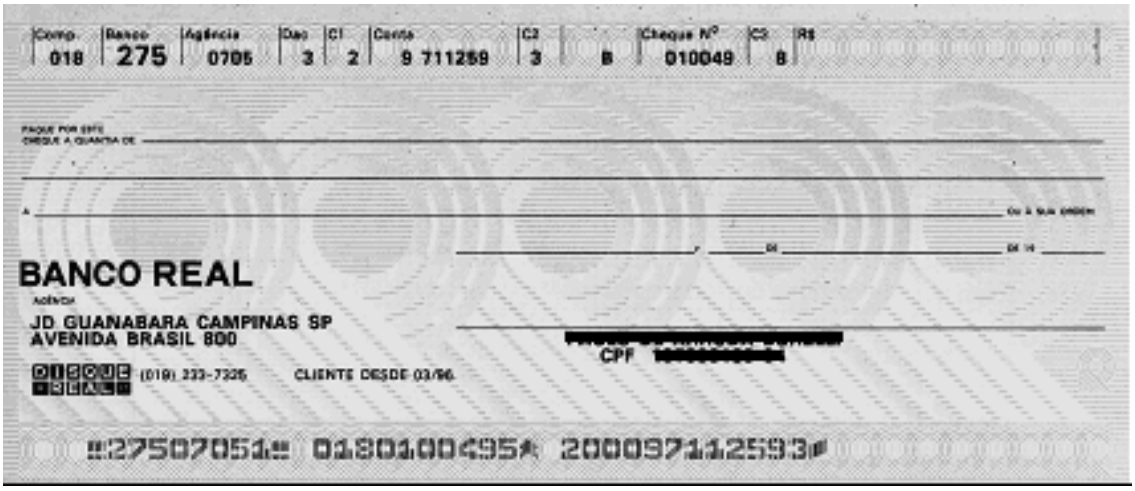

Figura 3: Imagem original Banco Real

Figura 4: Resultado do processamento da Figura 3.

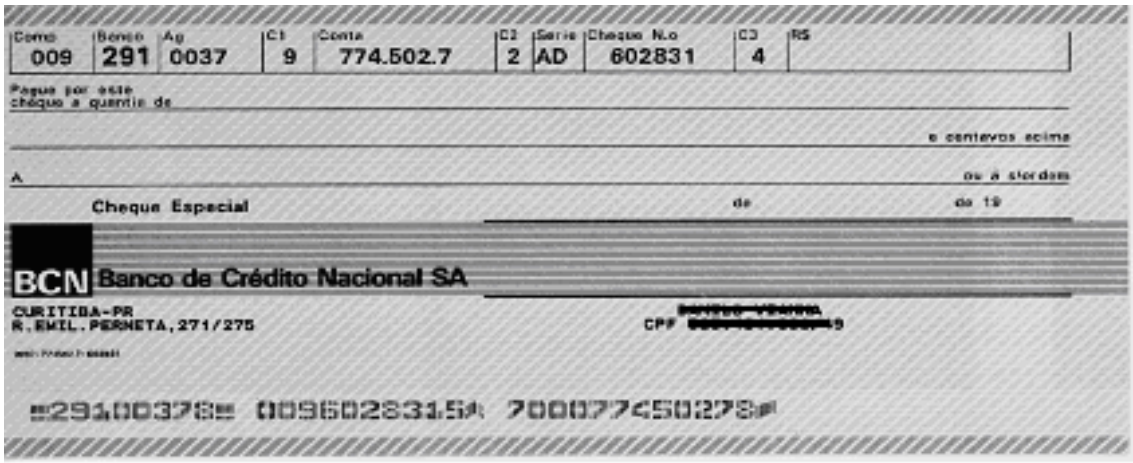

Figura 5: Imagem original Banco BCN.

Figura 6: Resultado do processamento da Figura 5.

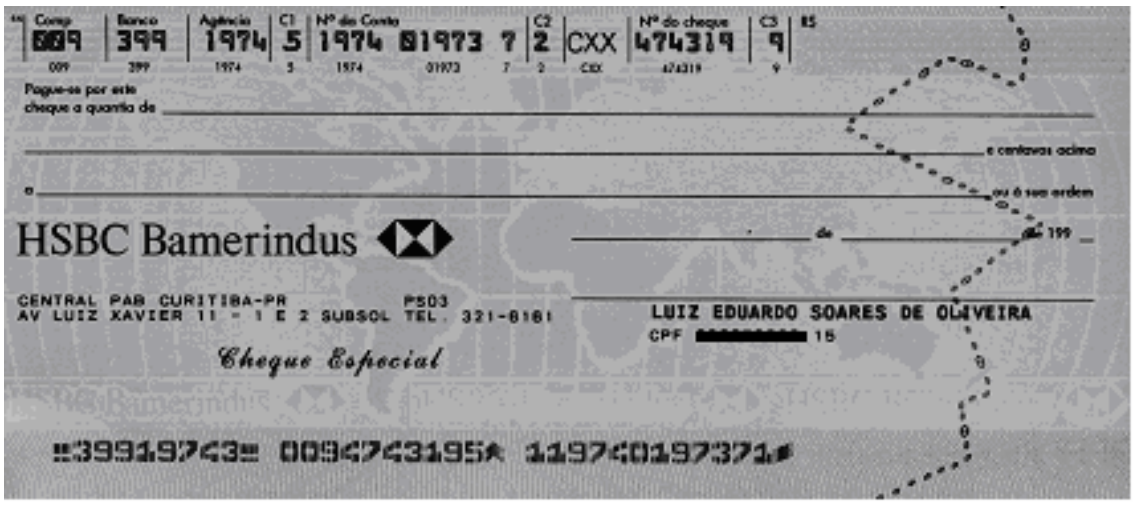

Figura 7: Imagem original Banco HSBC.

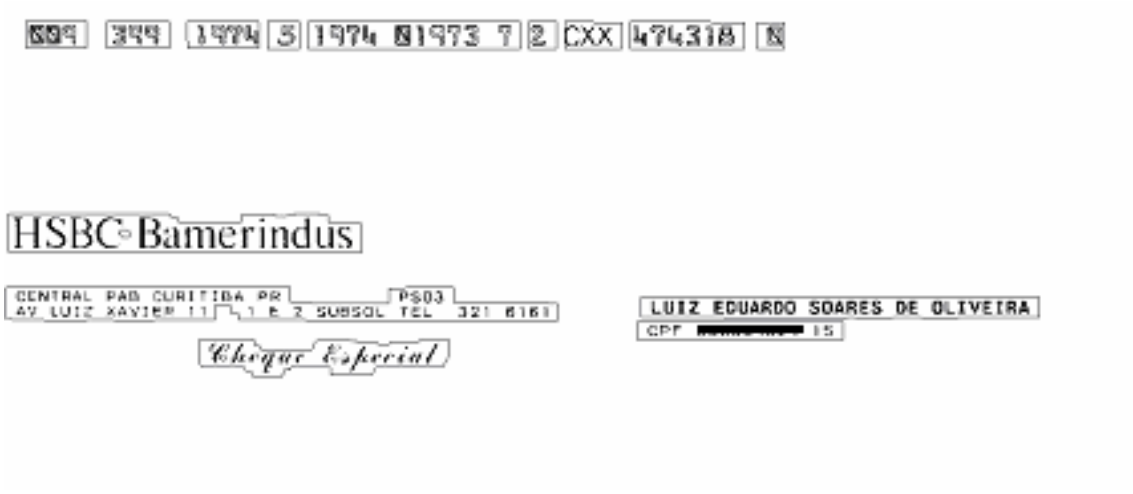

Figura 8: Resultado do processamento da Figura 7.

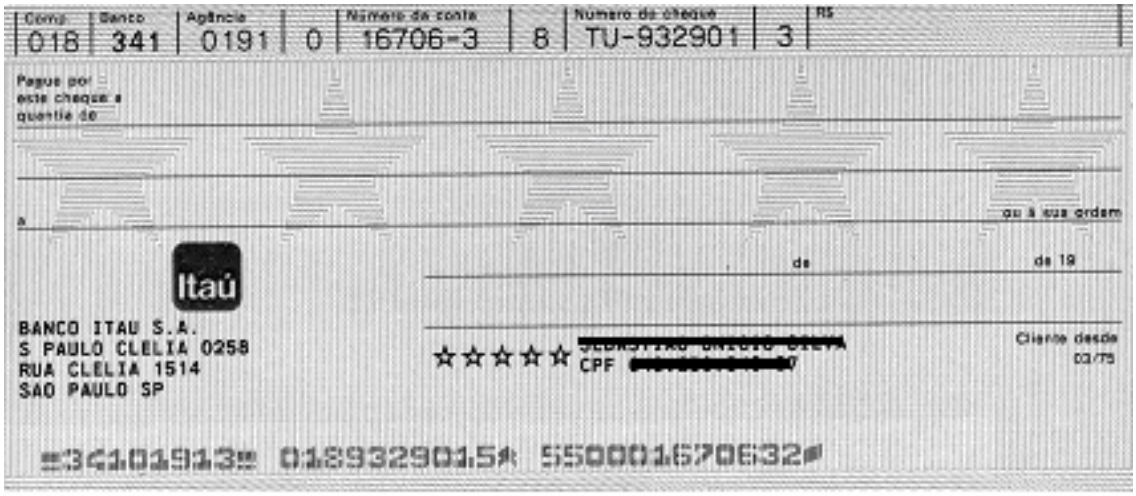

Figura 9: Imagem original Banco Itaú

Figura 10: Resultado do processamento da Figura 9.

Figura 11: Imagem original Banco Citibank

Figura 12: Resultado do processamento da Figura 11.

Figura 13: Imagem original Banco Banespa

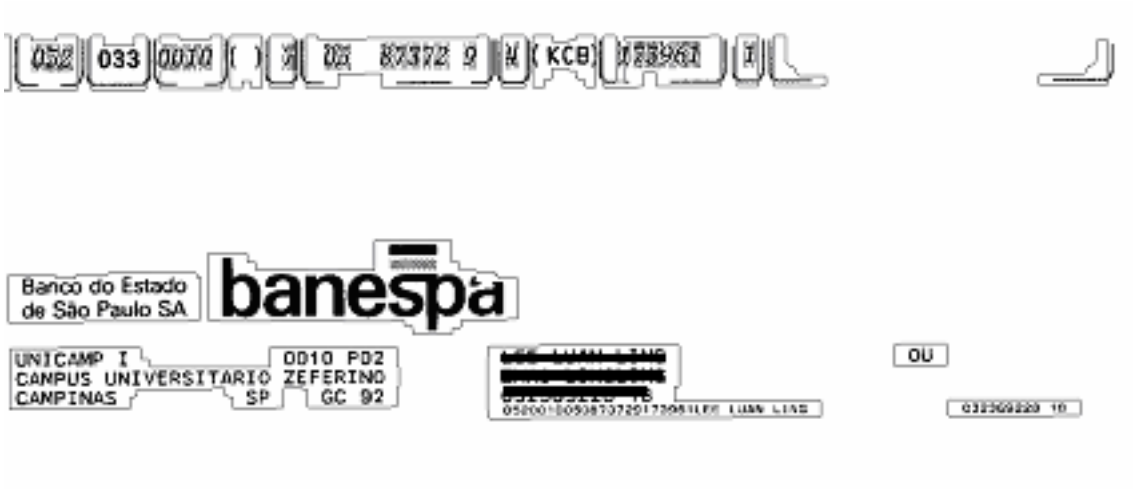

Figura 14: Resultado do processamento da Figura 13.

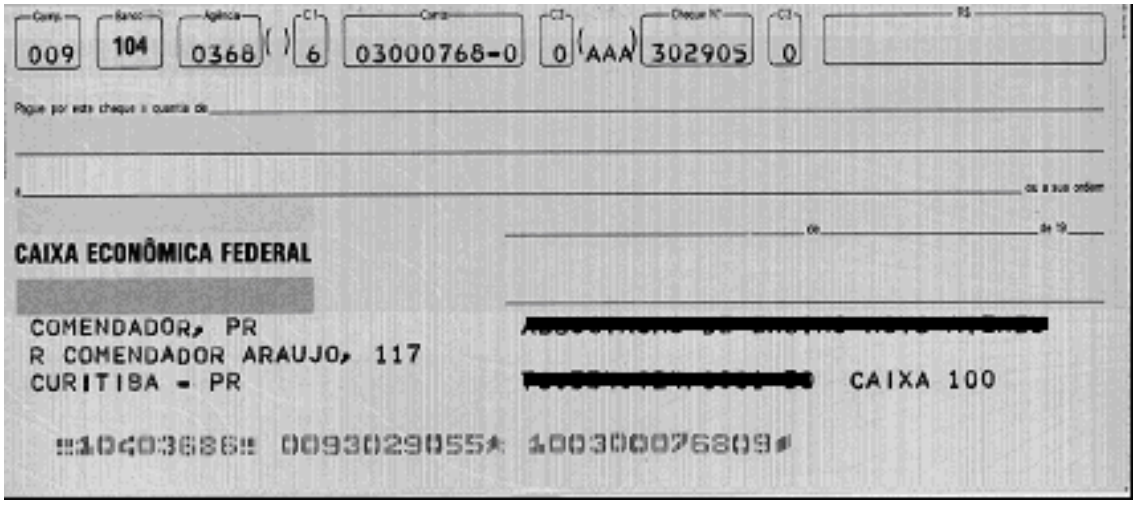

Figura 15: Imagem original Banco Caixa Econômica Federal

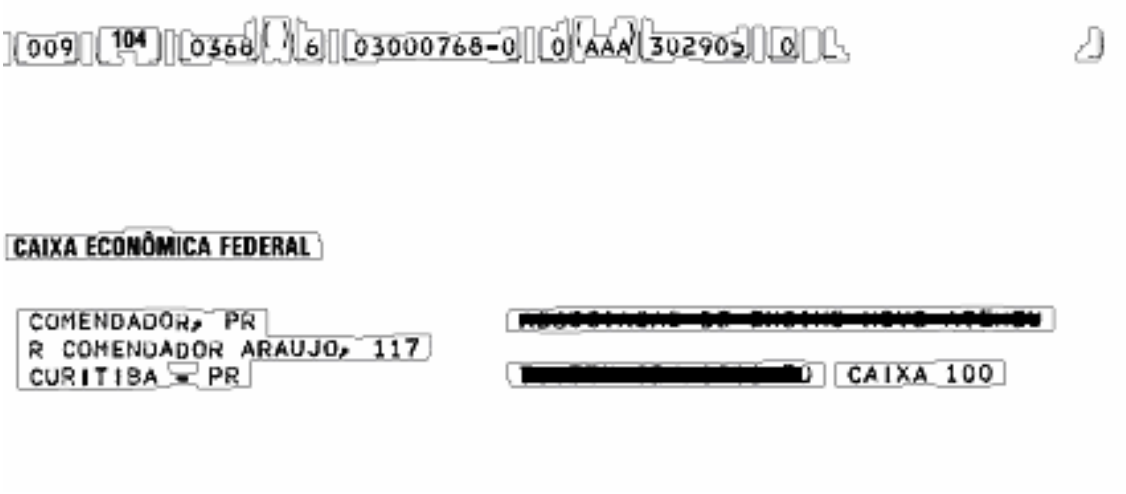

Figura 16: Resultado do processamento da Figura 15.

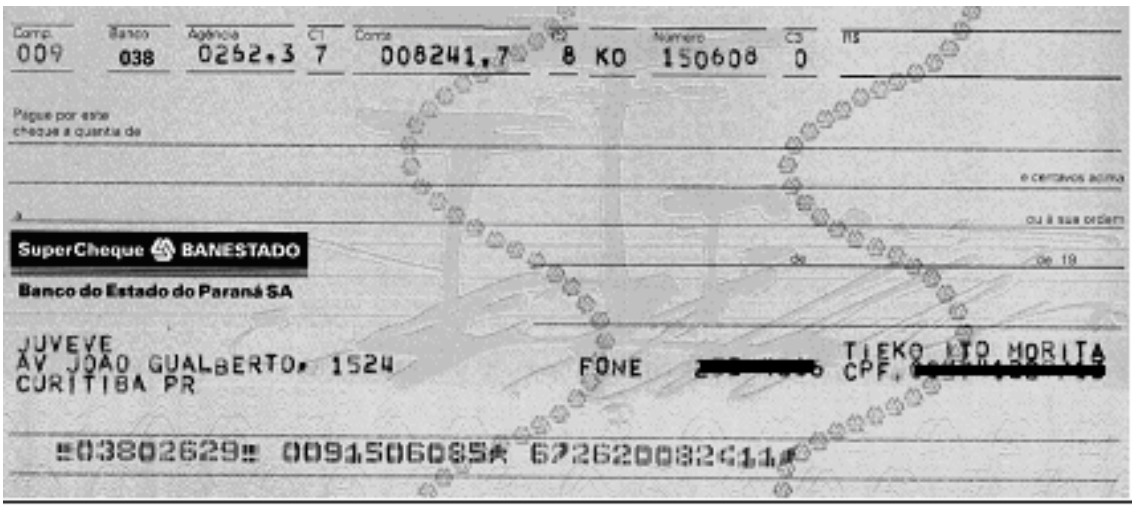

Figura 17: Imagem original Banco Banestado

Figura 18: Resultado do processamento da Figura 17.

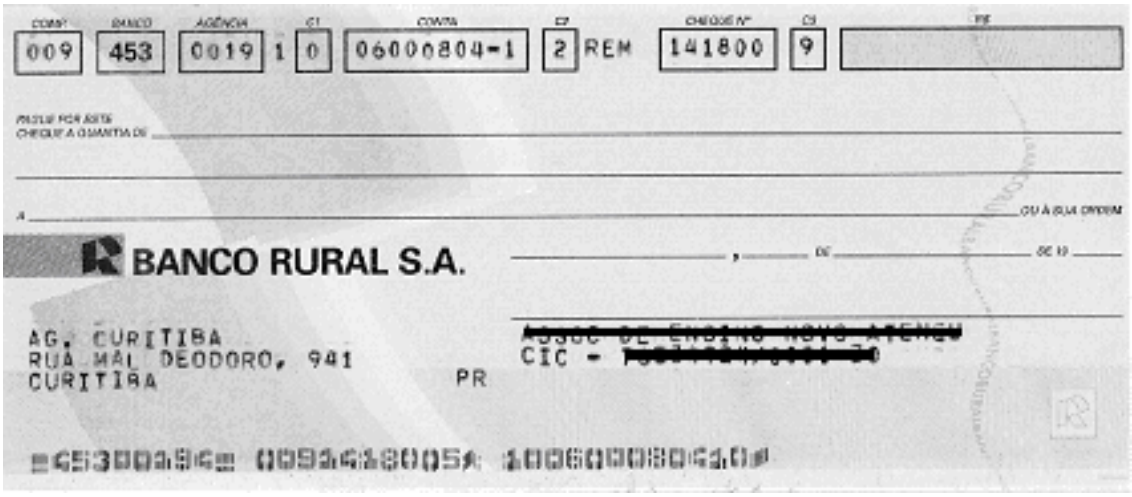

Figura 19: Imagem original Banco Rural

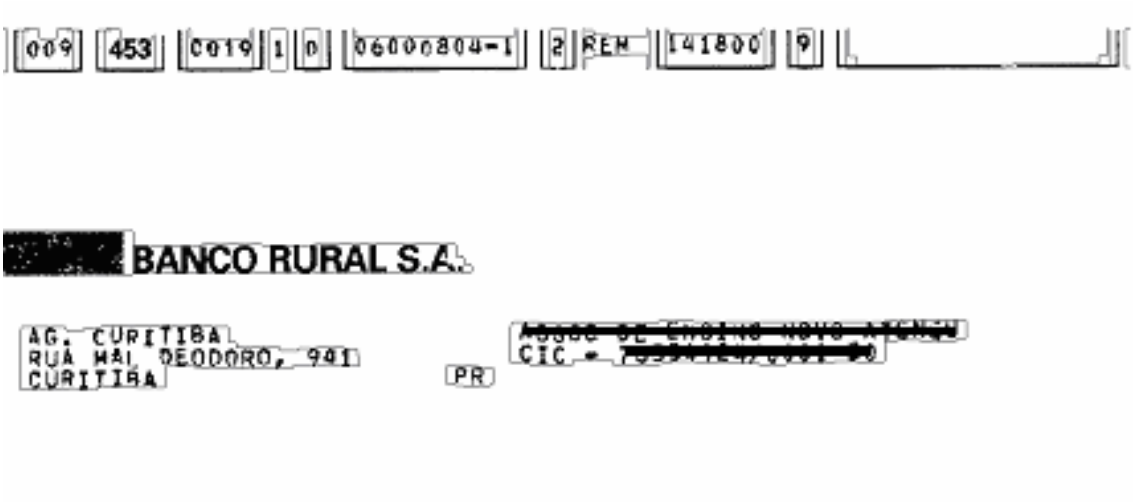

Figura 20: Resultado do processamento da Figura 19.