

---

# PROCESSAMENTO AUTOMÁTICO DE CHEQUES BANCÁRIOS: ARMAZENAMENTO E RECUPERAÇÃO DE INFORMAÇÕES

**Alessandro L. Koerich**

alekoe@livia.etsmtl.ca

**Luan Ling Lee**

zhao@decom.fee.unicamp.br

Laboratoire d'Imagerie, de Vision et d'Intelligence Artificielle, École de Technologie Supérieure  
Université du Québec

1100, rue Notre-Dame Ouest, H3C 1K3, Montréal, QC, Canada

Departamento de Comunicações, Faculdade de Engenharia Elétrica e de Computação

Universidade Estadual de Campinas

CP 6101, 13087-970, Campinas, SP, Brasil

---

**ABSTRACT** – This paper presents an automatic system for storage and retrieval of bankchecks. The bankchecks are digitized and their images are processed through an information extraction system that eliminates the redundant information and provides images containing only the user-input fields. The resulting images with the user-input fields are compressed and indexed using the information extracted from the MICR line. Finally, both compressed image and other useful information are stored in a database under a hybrid structure. The stored information can be recovered and the original bankcheck image can be reconstructed using both previously eliminated information and user-input field images. The experimental results by testing the proposed automatic system over real Brazilian bankchecks reveal that the proposed method provides images with good visual quality in addition to the advantage of efficient and automatic information storage, search and retrieval.

**KEYWORDS** – Bankcheck Processing, Document Storage and Retrieval.

**RESUMO** – Este trabalho apresenta um sistema automático para o armazenamento e recuperação de cheques bancários. Os cheques bancários são digitalizados e as imagens são processadas por um sistema de extração de informações que elimina as informações consideradas redundantes, fornecendo imagens que contêm somente as informações preenchidas pelos clientes. Estas imagens são comprimidas e indexadas usando palavras-chaves formadas por informações extraídas da linha MICR. Em seguida estas imagens são armazenadas em uma base de dados. A base de dados apresenta uma estrutura híbrida onde são utilizadas técnicas de árvore, tabelas hash e arquivo invertido. Tam-

bém são armazenadas na base de dados amostras dos padrões de fundo dos cheques e outras informações necessárias no processo de reconstrução. As informações armazenadas na base de dados podem ser recuperadas e a imagem do cheque original pode ser reconstruída através da inclusão das informações redundantes que foram anteriormente eliminadas. Os resultados experimentais a partir de testes realizados com cheques bancários brasileiros mostram que o método proposto fornece imagens reconstruídas com boa qualidade visual, além de permitir uma automatização nos processos de armazenamento, busca e recuperação de informações.

**PALAVRAS-CHAVE** – Processamento de Cheques Bancários, Armazenamento e Recuperação de Documentos.

## 1 INTRODUÇÃO

No Brasil os cheques bancários são amplamente utilizados, talvez até mais do que cartões magnéticos e cartões de crédito. Segundo a Câmara de Compensação do Banco Central, mensalmente são processados mais de 150 milhões de cheques bancários. As instituições financeiras são responsáveis pelo fornecimento de talonários de cheques bancários aos seus clientes. Quando um cliente emite um cheque bancário em favor de uma pessoa, normalmente, o favorecido pode receber a quantia relativa a este cheque de duas maneiras distintas: se dirigir diretamente ao caixa da agência bancária em que o cliente-emitente possui a conta e trocar este cheque pelo respectivo montante em dinheiro; ou depositar este cheque em sua própria conta bancária. A maioria das transações envolvendo cheques bancários se realiza na forma de depósito na conta bancária do favorecido. Neste caso o respectivo montante em dinheiro não fica disponível imediatamente ao depositante, pois, este cheque bancário depositado junto à instituição financeira onde o favorecido mantém uma conta, deve ser trocado pelo respectivo montante em dinheiro com a instituição financeira de origem do cheque. Este processo é conhecido

---

<sup>0</sup>Artigo submetido em 21/12/98

1a. Revisão em 14/04/99; 2a. Revisão em 14/01/00; 3a. Revisão em 28/11/00

Aceito sob recomendação do Ed. Cons. Prof. Dr. Jacques Szczupak

como compensação de cheques bancários.

O Banco Central mantém diversas Câmaras de Compensação em todo o país, onde as diferentes instituições financeiras realizam a troca de documentos. Após a instituição financeira de origem ter recebido o cheque emitido pelo seu cliente, ela retira o montante correspondente da conta do seu cliente—emitente e o repassa à instituição financeira onde o favorecido mantém a conta bancária, esta, por sua vez, repassa o montante ao seu cliente—favorecido. Os cheques bancários após serem compensados devem retornar à agência de origem ficando a disposição do cliente que os emitiu. As instituições financeiras ao pagarem um cheque bancário deveriam devolvê-lo ao cliente—emitente para fins de controle e prova de pagamento, porém, isto normalmente não é feito, pois, a lei permite que os estabelecimentos bancários enviem periodicamente à seus clientes os extratos de conta onde são reproduzidos os créditos e os débitos. Ainda assim, estes cheques pagos devem ficar nas agências a disposição dos clientes—emitentes por 60 dias. Após transcorrido este período, os cheques podem ser destruídos desde que a instituição financeira mantenha uma cópia microfilmada do cheque para comprovação do movimento da conta. Após transcorrido o período de doze meses as cópias microfotográficas também podem ser destruídas. Assim, o armazenamento de cheques bancários é uma tarefa realizada diariamente pelas instituições financeiras e que implica em um trabalho enorme envolvendo etapas de seleção, classificação e arquivamento. Além disso, existe a necessidade de uma grande quantidade de espaço para o armazenamento.

A micrografia constitui-se na técnica que utiliza a miniaturização da imagem como seu principal agente e que nas últimas décadas transformou-se numa importante ferramenta para o armazenamento de documentos. O processo de microfilmagem atualmente soluciona um dos maiores problemas das instituições financeiras: a redução no tamanho dos documentos a serem armazenados. A redução no tamanho dos documentos permite uma grande economia de espaço, pois, geralmente, um documento microfilmado ocupa um espaço inferior a 2% do espaço ocupado pelo documento original em papel. Esta solução, porém, não se mostra adequada se pensarmos no processo de armazenamento como um todo. Se o cliente desejar obter a cópia de algum documento que já foi destruído, o banco faz uma procura manual dentre milhares de documentos armazenados em rolos de filme até encontrar a seção que contém a imagem do cheque. O filme é levado à um dispositivo especial que faz a ampliação e a impressão do documento em papel. A imagem final apresenta uma baixa qualidade em relação à imagem do documento original. O Banco Central estabelece uma série de normas e procedimentos para a microfilmagem, armazenamento e manutenção de cheques. Estas normas estão contidas na Resolução número 885 do Banco Central (1983). As tarefas de geração dos microfilmes, armazenamento, manutenção, busca e recuperação de informações não parecem ser nem um pouco eficientes, pois, estes processos são realizados de maneira predominantemente manual. Além disso, existem inúmeras limitações na maneira como estes documentos são manipulados, como por exemplo o custo da transformação destes documentos da forma de papel para uma forma microfotográfica, o acesso demorado e ineficiente às informações armazenadas, vulnerabilidade do microfilme à condições ambientais, cuidados especiais no armazenamento, etc.

Com a evolução dos equipamentos de aquisição de imagens e processadores e o aumento da capacidade dos dispositivos de armazenamento, os computadores podem oferecer soluções alternativas para o problema de armazenamento de cheques bancários

os, aumentando a confiabilidade e facilitando a manipulação das informações, além de propiciar uma redução nos custos operacionais das instituições financeiras. As técnicas de processamento de imagens e reconhecimento de padrões podem oferecer soluções eficientes para a manipulação e o armazenamento dos cheques bancários. Um cheque bancário pode ser facilmente digitalizado através de um dispositivo óptico de aquisição (scanner). Esta imagem pode, então, ser transformada para algum formato comercial (GIF, JPEG, TIFF, BMP) e armazenada em diferentes tipos de meios magnéticos ou ópticos. Esta imagem pode ainda ser enviada com rapidez para praticamente qualquer local servido por uma rede de comunicação. Porém, simplesmente digitalizar e armazenar milhares de imagens, não resolve algumas das questões colocadas anteriormente, pois, cada uma das imagens a serem armazenadas deve ser identificada, bem como organizadas em uma base de dados de modo que tarefas como inclusão, busca, recuperação e apagamento sejam realizadas de maneira fácil e rápida. Além disso, devido a grande quantidade de informação, representar as imagens de uma maneira compacta também é altamente desejável.

Os trabalhos publicados tratam principalmente do processamento e armazenamento de formulários pré-impressos em geral (Akiyama e Hagita, 1990; Satoh et al., 1995). Casey et al. (1992) apresentam um sistema inteligente para o processamento de formulários que tem a capacidade de capturar informações de imagens de documentos digitalizados e indexá-los automaticamente para o armazenamento e recuperação. Yasuda et al. (1980) apresentam uma técnica para a compressão de imagens de cheques bancários, onde inicialmente são extraídos os pixels pertencentes aos caracteres impressos através de técnicas de segmentação. Em seguida estes pixels são separados em planos de bits e níveis de cinza. Uma codificação preditiva adaptativa é aplicada ao plano de níveis de cinza. As informações dos cheques são armazenadas como uma combinação de códigos derivados do plano de bits e do plano de níveis de cinza. Este método apresenta um fator de compressão de 80% para imagens digitalizadas em 8 níveis de cinza.

Assim, neste trabalho utilizaremos um sistema anteriormente proposto, responsável pela extração das informações de interesse dos cheques bancários, no caso, as informações preenchidas pelo clientes, que são: valor numérico, valor por extenso, nome do favorecido, local e data de emissão e assinatura (Lee et al., 1997). Estas informações são fornecidas na saída do sistema de extração na forma de uma imagem binária. Esta imagem será, então, manipulada pelo sistema proposto de modo que ela seja comprimida, indexada e armazenada em um banco de dados. O banco de dados possui uma estrutura híbrida onde são utilizadas tabelas hash, árvores e arquivos invertidos. Além disso, apresentamos a utilização de ferramentas de busca e recuperação de informações bem como a reconstrução das imagens originais dos cheques bancários. O principal objetivo deste trabalho é apresentar uma solução para o armazenamento automático de imagens de cheques bancários que seja eficiente tanto em termos da qualidade visual dos documentos armazenados e recuperados, como na rapidez para a recuperação e visualização dos cheques bancários.

A organização deste trabalho é a seguinte. A seção 2 apresenta uma descrição geral dos sistemas de extração e armazenamento de informações. Na seção 3 são descritas as operações de compressão e indexação das imagens dos cheques bancários. A estrutura da base de dados é apresentada na seção 4. Na seção 5 apresentamos os procedimentos usados para a recuperação e

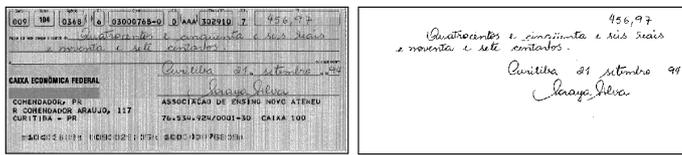


Figura 1: Exemplo de um cheque bancário e as informações preenchidas extraídas

reconstrução das imagens dos cheques bancários. Os resultados experimentais e a análise do desempenho do sistema são apresentados na seção 6. Finalmente, na última seção são apresentadas uma discussão e a conclusão.

## 2 DESCRIÇÃO GERAL DO SISTEMA

O sistema de extração de informações como apresentado em Koerich e Lee (1997) e Lee et al. (1997), é responsável pela extração das informações preenchidas pelos clientes nos cheques bancários. Este sistema é composto por um scanner óptico, responsável pela transformação do documento em papel para uma representação digital, um scanner MICR (Magnetic Ink Character Recognition), responsável pela leitura dos caracteres codificados e impressos pelo processo MICR utilizando uma fonte CMC-7 (Caracteres Magnéticos Codificados em 7 Barras). Além destes dispositivos de entrada o sistema também possui uma série de algoritmos responsáveis pelo processamento das imagens digitalizadas dos cheques bancários fornecendo na saída somente as informações preenchidas presentes nos cheques no formato de uma imagem binária. A figura 1 apresenta um exemplo de um cheque bancário digitalizado e as informações preenchidas extraídas deste cheque.

Em trabalhos anteriores (Lee et al., 1997; Koerich e Lee, 1997) analisamos o layout dos cheques bancários e observamos que as informações impressas e o padrão de fundo são similares para todos os cheques emitidos por um mesmo banco. Fizemos uso deste conhecimento e desenvolvemos um sistema capaz de extrair as informações preenchidas pelos clientes, visando realizar o reconhecimento automático destas informações (verificação de assinaturas, reconhecimento do valor, etc.). Porém, podemos aproveitar uma grande parte da metodologia empregada para a extração de informações para tratarmos o problema do armazenamento de cheques bancários. Assim, utilizando como entrada a imagem contendo somente as informações preenchidas fornecida pelo sistema de extração e também as informações lidas pelo scanner MICR, podemos desenvolver um sistema para o armazenamento e recuperação de cheques bancários. A figura 2 apresenta uma visão geral do sistema de armazenamento onde aparecem somente os principais módulos.

A imagem com as informações preenchidas fornecida pelo sistema de extração é passada a um módulo onde ela sofre um processo de compressão. A imagem comprimida é indexada utilizando-se as informações da linha MICR fornecidas pelo módulo de extração. Em seguida esta imagem é armazenada em um banco de dados, ficando disponível para futuras consultas. Para recuperar esta imagem do banco de dados é necessário digitar em um terminal uma palavra-chave idêntica à utilizada na indexação. A partir desta palavra-chave, a base de dados é acessada e são selecionados a imagem com as informações preenchidas, o padrão de fundo original do cheque e diversos parâmetros e informações necessárias para a geração de uma imagem contendo as informações impressas presentes no cheque original. Estes parâmetros e informações são levados a um módulo

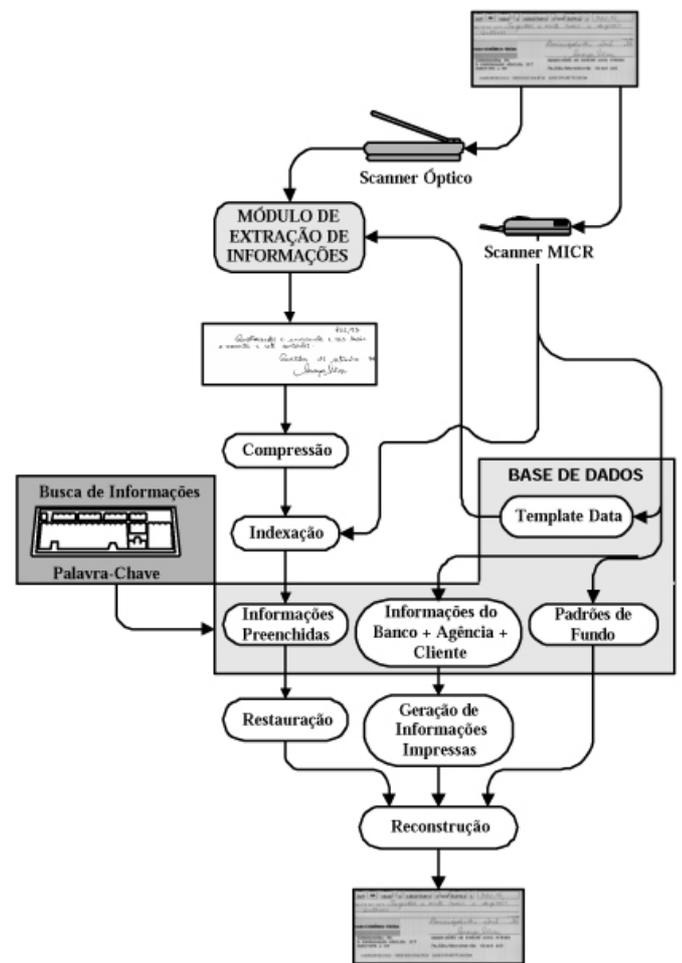


Figura 2: Principais elementos do sistema de armazenamento e recuperação de informações

de geração de informações impressas onde é gerada uma imagem binária correspondente às informações impressas presentes no cheque. A imagem gerada, a imagem com as informações preenchidas e a imagem do padrão de fundo são levadas a um módulo de reconstrução onde é feita a reconstrução da imagem original do cheque. Em seguida esta imagem reconstruída pode ser direcionada a diferentes dispositivos para a visualização, como impressoras, terminais de vídeo, terminais de atendimento automático (ATMs) ou transmitidas via uma rede de comunicação. A seguir apresentaremos em detalhes os principais módulos do sistema de armazenamento e recuperação de cheques bancários.

## 3 COMPRESSÃO E INDEXAÇÃO

A operação de compressão consiste em representar uma imagem por um conjunto de dados que possua o menor tamanho possível. Um dos problemas inerentes aos sistemas de processamento de documentos é justamente a representação compacta da informação. Existem vários algoritmos que fazem a compressão de uma imagem (Aarps e Truong, 1994; Howard e Vitter, 1994), porém, como o tipo de documento que estamos tratando segue normas gerais e possui um layout padronizado, podemos pensar em uma esquema de compressão que explore estas características particulares.

Casey et al. (1992) propõe um método alternativo para o armazenamento de documentos pré-impressos onde cada documento é analisado e somente certos campos onde as informações de-

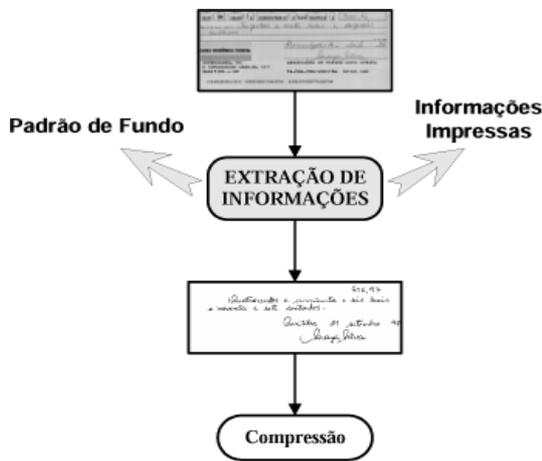


Figura 3: Compressão de cheques bancários em 2 níveis

sejadas aparecem são extraídos e armazenados. A idéia básica desta técnica é remover as partes comuns existentes em todos os formulários do mesmo tipo e armazenar somente os dados variáveis. Como os dados preenchidos possuem um volume menor de informação em relação à imagem integral do documento, devemos esperar um ganho no sentido da razão de compressão, pois, o número de bits necessários para o armazenamento da imagem seria reduzido, principalmente em documentos com fundos ilustrativos e grande quantidade de campos.

Os cheques bancários podem ser considerados como documentos baseados em formulários pré-impressos, isto é, eles possuem informações impressas comuns a todos os cheques do mesmo tipo e campos a serem preenchidos. Como temos conhecimento prévio do layout dos cheques bancários, podemos considerá-lo como sendo uma informação redundante, sendo que os dados preenchidos pelos clientes sobre os cheques são as informações de interesse. Eliminando as informações redundantes, isto é, o padrão de fundo e as informações impressas, podemos obter um esquema de compressão eficiente, pois, estas informações são as que ocupam maior espaço. Esta solução é plausível, pois, geralmente os cheques emitidos por um mesmo banco, possuem em comum o padrão de fundo e as informações impressas. Como resultado, não é necessário ocupar espaço adicional em memória porque somente uma amostra do padrão de fundo do cheque necessita ser armazenada. As informações impressas podem ser armazenadas uma única vez em uma base de dados na forma de texto ASCII. Os procedimentos apresentados até aqui já foram implementados no sistema de extração de informações (Koerich e Lee, 1997), assim, precisamos somente utilizar a imagem fornecida na saída deste sistema. Porém, mesmo após a eliminação das informações redundantes das imagens dos cheques bancários, ainda existe uma quantidade considerável de áreas nestas imagens que podem ser consideradas como redundantes.

Assim, a etapa de compressão de informações pode ser dividida em duas etapas conforme apresentado na figura 3: a primeira faz uso da informação estrutural dos cheques bancários, eliminando as partes consideradas comuns entre os cheques de uma mesma instituição financeira; a segunda etapa consiste de um operação de compressão propriamente dita, ou seja, algoritmos que consideram somente a distribuição/frequência de pixels. Como a primeira etapa de compressão fornece uma imagem binária, diversas técnicas podem ser utilizadas na segunda etapa: LZW, Run Length, CCITT Group 3 e Group 4, e Código de Huffman.

A vantagem em utilizar este esquema de compressão em dois ní-

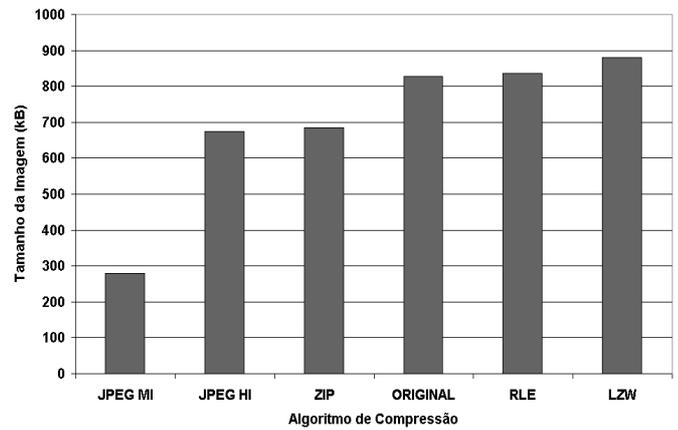


Figura 4: Tamanho da imagem original do cheque comprimida por diferentes métodos

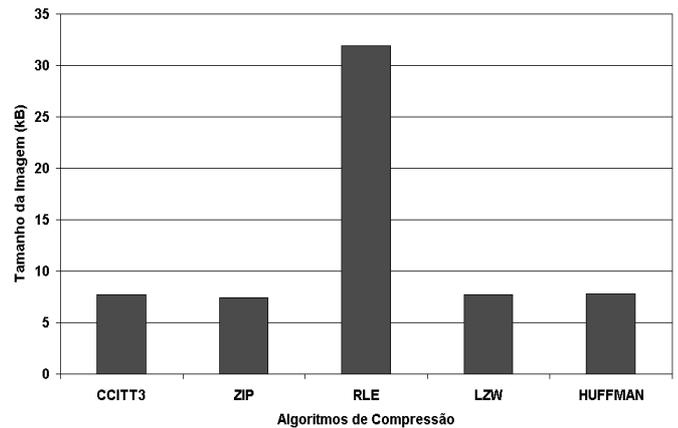


Figura 5: Tamanho da imagem com as informações preenchidas comprimida por diferentes métodos

veis em relação a aplicarmos diretamente um algoritmo de compressão sobre a imagem original do cheque, pode ser comprovado pela comparação dos tamanhos das imagens. A figura 4 apresenta o tamanho médio das imagens originais dos cheques comprimidas, enquanto a figura 5 apresenta o resultado da imagem do cheque comprimida utilizando-se o esquema de compressão em 2 níveis.

Com o esquema de compressão em 2 níveis, conseguimos reduzir de uma maneira eficiente o tamanho das imagens a serem armazenadas. Porém, devemos ainda identificar cada uma destas imagens, bem como, determinar a maneira com que elas serão armazenadas em uma base de dados.

### 3.1 Indexação de Imagens

A indexação de documentos é provavelmente a tarefa que consome mais tempo em sistemas de armazenamento de imagens. Geralmente dois ou mais campos devem ser digitados antes que um documento possa ser arquivado para posterior recuperação. Para tornarmos esta tarefa mais rápida e eficiente podemos fazer uso da linha MICR no processo de indexação das imagens. No momento da leitura da linha MICR as informações são passadas ao módulo de armazenamento de informações. Este módulo faz a interpretação das informações codificadas e gera as palavras-chave necessárias para a identificação e indexação das imagens com as informações preenchidas (figura 6). Assim, a partir da operação de indexação, cada imagem passa a ter associada a si

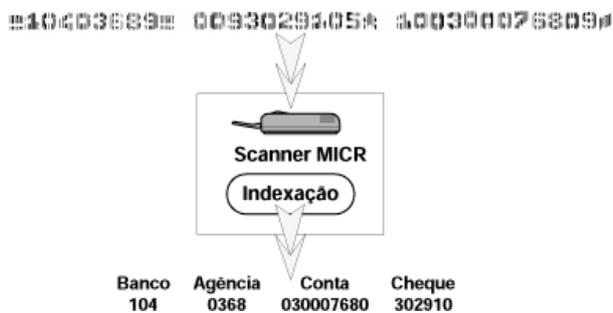


Figura 6: Processo de indexação de cheques bancários

Tabela 1: Ordem dos possíveis valores para os elementos que formam as palavras-chave.

Palavra-Chave	Número de Dígitos	Valores Possíveis
Banco	03	$10^3$
Agência	04	$10^4$
Número da Conta	10	$10^{10}$
Número do Cheque	06	$10^6$

uma palavra-chave única que a identifica em todo o sistema bancário. A palavra-chave para cada cheque terá 23 dígitos e será composta pela concatenação dos quatro campos extraídos da linha MICR, ou seja, pela concatenação do número do banco, do número da agência, do número da conta e do número do cheque.

## 4 BASE DE DADOS

Após o processamento visando a redução no tamanho da imagem e sua indexação devemos pensar na maneira como as imagens serão armazenadas, isto é, de que maneira elas serão organizadas em um meio magnético ou óptico e de que maneira poderemos posteriormente acessá-las e recuperá-las. Nosso objetivo é construir uma estrutura de dados que seja capaz de armazenar uma grande quantidade de cheques bancários (mais de 1.000.000) de uma maneira eficiente, de modo que os tempos de resposta (inserção e busca de informações) sejam razoáveis.

Estamos trabalhando com imagens representando informações extraídas de diferentes cheques bancários preenchidos por diferentes clientes e emitidos por diferentes bancos. Como as palavras-chave são dadas pela concatenação do número do banco, número da agência, número da conta e número do cheque, teremos que cada palavra-chave terá 23 dígitos e poderá assumir  $10^{23}$  diferentes valores. Como consequência, dois problemas podem surgir: a falta de espaço para o armazenamento; uma grande quantidade de posições desperdiçadas, pois, jamais seriam preenchidas. Assim, é extremamente interessante alocarmos um espaço em memória que tenha um tamanho compatível com os dados a serem armazenados. Considerando que as palavras-chave são formadas pela concatenação de 4 elementos, para cada parte da palavra-chave teremos uma faixa possível de valores conforme apresentado na tabela 1. Assim, utilizaremos uma estrutura em árvore com 4 níveis, onde cada nó deverá possuir uma estrutura adequada para manipular o grande volume de informações que neles serão inseridos.

### 4.1 Estrutura da Base de Dados

A maneira específica de implementarmos uma estrutura de dados está fortemente ligada às operações que serão realizadas, à quantidade e à forma dos dados a serem armazenados. Como estamos

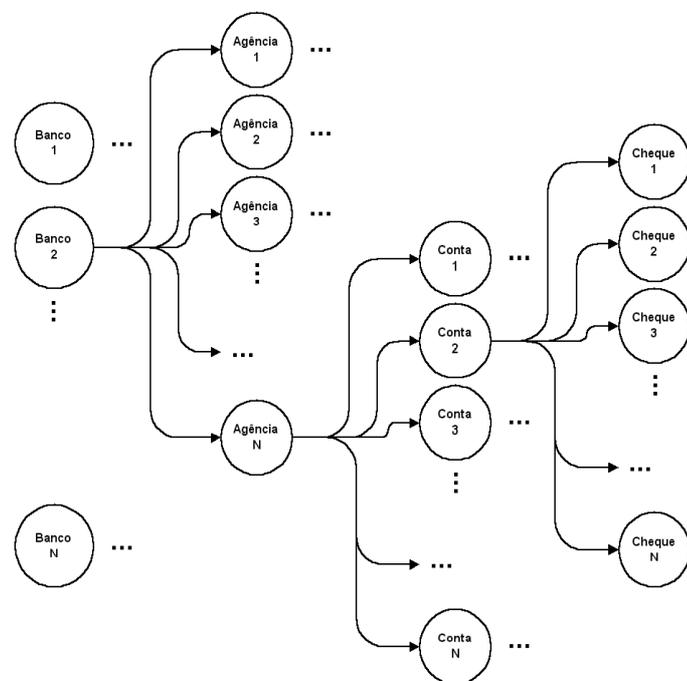


Figura 7: A estrutura de dados em forma de árvore

tratando de uma aplicação onde a quantidade de informações variáveis é considerável, iremos dividir o problema criando diversas estruturas de dados similares organizadas de maneira hierárquica. No intuito de generalizarmos ao máximo nossa aplicação, dividiremos a estrutura de dados em quatro níveis associados a uma árvore N-ária, onde cada nível desta árvore será dedicado ao armazenamento das informações dos *bancos*, *agências*, *contas* ou *cheques*.

Os dois primeiros níveis, por possuírem uma variabilidade muito baixa ao longo do tempo, podem ser representados por um array estático. Por outro lado, os dois níveis restantes são fortemente dependentes do tempo, apresentando variações diárias. Assim, estes dois níveis devem ser representados por estruturas dinâmicas que possam alterar seu tamanho em função da quantidade de informações.

O nível de informações dos *bancos* compreende todos os bancos que operam no sistema financeiro representados pelo seu número-código de inscrição no Serviço de Compensação, composto de 3 caracteres numéricos. O nível de informações das *agências* compreende todas as agências de um mesmo banco representadas pelo número código da agência, composto de 4 caracteres numéricos. O nível de informações das *contas* compreende todas as contas de uma mesma agência representado pelo número da conta do emitente, compondo-se de no máximo 10 caracteres numéricos. Por último, temos o nível dos *cheques* representado por todos os números de série dos cheques pertencentes a uma mesma conta, composto por 6 caracteres numéricos. Na figura 7 temos a organização da estrutura de dados na forma de uma árvore N-ária. Nesta figura, a árvore foi expandida somente para um banco, uma agência e uma conta, porém para os demais bancos, agências e contas, a estrutura da árvore é similar.

Esta divisão da estrutura de dados em 4 níveis propicia uma maior flexibilidade ao sistema, pois cada nível pode ser tratado independentemente sem afetar os demais níveis. Além disso, esta divisão minimiza o efeito das colisões de modo que o tempo de resposta do sistema não é afetado de maneira significativa com

o aumento da quantidade de informações inseridas na base de dados.

Portanto, resta-nos agora, determinar a estrutura interna de cada um dos nós desta árvore e de que maneira eles serão inteligidos aos níveis superiores e inferiores.

## 4.2 Arquivo Invertido

Para aumentar a eficiência das operações realizadas em nossa base de dados, utilizamos a técnica de arquivos invertidos. Assumindo-se um conjunto de imagens, onde, para cada imagem é atribuída uma palavra-chave, temos que um arquivo invertido é um lista de palavras-chave onde cada palavra-chave aponta para o endereço onde a imagem correspondente está armazenada. O uso de um arquivo invertido melhora a eficiência de procura em várias ordens de grandeza, pois não iremos procurar a imagem diretamente, mas sim, uma palavra-chave que está relacionada com a imagem armazenada no banco de dados. Como veremos a seguir, utilizamos a técnica de arquivos invertidos associada a tabelas hash e portanto teremos uma sucessão de palavras-chave, cada uma associada a um endereço, até chegarmos à imagem propriamente dita.

## 4.3 Níveis da Base de Dados

Conforme mencionamos anteriormente, os dois primeiros níveis de informação, não estão sujeitos a modificações dinâmicas, pois, o número de bancos e de agências não são alterados de uma maneira rápida. Assim, para ambos os níveis, a estrutura de dados pode ser do tipo estática e possuir um tamanho fixo. Este tamanho que é único para o array dos bancos, pode assumir diferentes valores para o nível das agências, pois, cada banco possui um número diferente de agências, variando de mais de 2.000 para os grandes bancos, até 2 ou 3 para instituições financeiras de menor porte.

Por outro lado, a utilização de arrays de tamanho fixo para o nível das contas e principalmente para o nível dos cheques, onde a variação na quantidade de informações é bem mais dinâmica, torna-se inviável. Para estes dois níveis utilizamos tabelas hash implementadas na forma de lista dinâmicas. A estrutura interna para os nós correspondentes aos 4 níveis é apresentada de maneira simplificada na figura 8.

## 4.4 Tabelas Hash

Uma estrutura de dados do tipo hash é meramente uma matriz de tamanho fixo contendo palavras-chave. Cada palavra-chave é mapeada por algum número na faixa de 1 até  $ht$ , onde  $ht$  é o tamanho da tabela hash, e colocada na posição correspondente. A função que realiza o mapeamento entre as palavras-chave e os endereços hash é chamada de função hash. Este mapeamento deve ser simples e assegurar que quaisquer duas palavras-chave distintas sejam armazenadas em posições diferentes. Como há um número finito de posições e um número muito grande de possíveis palavras-chave, devemos procurar uma função hash que distribua as palavras-chave igualmente entre as posições. Além disso, devemos escolher uma função que decida o que fazer quando duas palavras-chave possuírem o mesmo endereço (colisão).

A decisão de utilizar tabelas hash está vinculada à critérios como a frequência das operações que serão realizadas, as características das informações a serem armazenadas, o tempo médio de operação e as necessidades de armazenamento. Após anali-

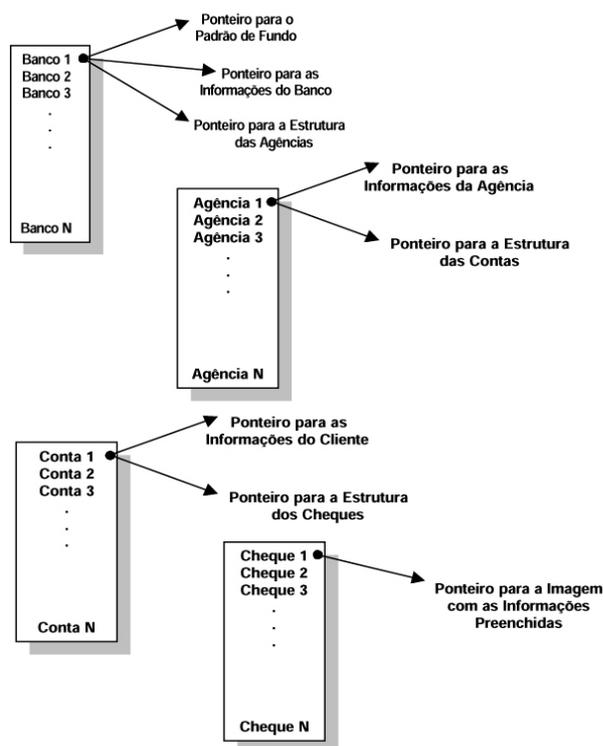


Figura 8: Estrutura interna dos nós da árvore N-ária de acordo com o tipo de nível

sarmos alguns tipos de estruturas de dados, optamos por utilizar internamente aos nós da árvore N-ária uma estrutura baseada em tabelas hash. Este tipo de estrutura de dados é a que melhor se adapta às características de nossa aplicação (Csabharwal e Bhatia, 1995; Cormen et al., 1990; Frakes e Yates, 1992), que são: informações de tamanho e forma limitadas, grande número de inserções, poucos apagamentos e poucas atualizações. A principal característica da técnica hash é seu excelente desempenho para inserções e buscas. A maior desvantagem desta técnica é a possibilidade de grandes tempos para o pior caso.

### 4.4.1 Funções HASH

Sem dúvida, um dos elementos mais importantes de uma tabela hash, é a função hash, isto é, a função que faz o mapeamento entre as palavras-chave e as posições da tabela. Utilizamos a técnica chamada *rehashing*, que envolve a utilização de duas funções hash,  $ha_1$  e  $ha_2$ . Inicialmente as palavras-chave são mapeadas pela função  $ha_1$ . Caso ocorra um colisão, a função  $ha_2$  é aplicada sucessivamente até uma posição vazia da tabela ser encontrada. Assim, as funções hash utilizadas em nosso sistema são dadas por:

$$ha_1(k) = k \text{ mod } ht \quad (1)$$

$$ha_2(k) = 1 + \left(\frac{k}{ht}\right) \text{ mod } (ht - 1) \quad (2)$$

onde  $ha_1$  é a função hash primária,  $ha_2$  é a função hash secundária,  $ht$  é o tamanho da tabela hash (número primo),  $k$  é a palavra-chave e  $\text{mod}$  é o operador resto da divisão.

#### 4.4.2 Alteração Dinâmica do Tamanho

A determinação de um tamanho fixo para as tabelas hash é um dos principais problemas quando a quantidade de informações a serem inseridas varia rapidamente. As tabelas hash relativas as contas e principalmente aos cheques, sofrem alterações com uma grande velocidade, assim, a determinação do tamanho destas tabelas torna-se um fator crítico. Desta maneira, visando aumentar a confiabilidade do sistema e torná-lo quase imune a estouros do tamanho da tabela, definimos alguns critérios para ajustar dinamicamente o tamanho das tabelas hash.

Os tamanhos iniciais das tabelas hash são definidos considerando um conhecimento a priori sobre o volume de contas e cheques associados a cada agência. Desta maneira, definimos o tamanho inicial como sendo:

$$ht = p \cdot \frac{nych}{t} \cdot tol \quad (3)$$

onde  $ht$  é o tamanho da tabela a ser determinado,  $p$  é o período pelo qual os cheques serão mantidos na base de dados (geralmente 12 meses),  $\frac{nych}{t}$  é o número de cheques inseridos por unidade de tempo ( $t$ ), ( $t$  geralmente corresponde ao período de um mês), e  $tol$  é o fator de tolerância.

Esta tabela hash possui uma característica especial. A sua última posição é reservada para, caso seja necessário, armazenar um ponteiro para uma sub-tabela. Esta sub-tabela será gerada somente quando houver alguma indicação de que a tabela-mãe atingiu seu limite. Assim, se para uma mesma palavra-chave, uma operação de inserção atinja o final da tabela-mãe por mais de duas vezes e o fator de carga ( $fc$ ) definido pela equação 4 seja superior a 99%, esta sub-tabela é gerada, e passa a funcionar como uma extensão da tabela original.

$$fc = \frac{nych}{ht} \quad (4)$$

Porém, isso implica também em uma mudança na política de endereçamento hash. Assim, a partir da criação desta tabela, todos os demais elementos a serem inseridos serão diretamente direcionados para esta sub-tabela. Como consequência, teremos também uma alteração no processo de busca nesta tabela. Caso o final da tabela-mãe seja atingido por mais de duas vezes para uma mesma palavra-chave a busca passa a ser feita na sub-tabela. O tamanho desta sub-tabela será dado por:

$$hts = \frac{nych}{tr} \cdot tol \quad (5)$$

onde  $\frac{nych}{tr}$  agora é o número de cheques inseridos em um período reduzido de tempo  $tr$ , de modo a incluir no tamanho da tabela informações do movimento atual da conta. Além disso, a última posição desta sub-tabela também está reservada para a criação de uma nova sub-tabela.

Na prática, a condição de preenchimento completo de uma tabela hash em nosso sistema é um evento raro, porém, possível de

ocorrer. Assim, omitiremos maiores detalhes sobre este tópico. Uma outra solução que foi também investigada e utilizada com sucesso, foi simplesmente, a construção de um nova tabela de maior tamanho e transferência dos dados da tabela original para esta nova tabela. Esta operação leva em média 1 minuto<sup>1</sup>.

Na seção 6 apresentaremos uma análise do desempenho do sistema implementado segundo as características aqui apresentadas.

## 5 RECUPERAÇÃO E RECONSTRUÇÃO DE IMAGENS

Um sistema de recuperação de informações promove o *matching* entre as perguntas do usuário e os documentos armazenados na base de dados. Um documento é um objeto de dados, geralmente textos que podem também conter outros tipos de informações como fotos, gráficos, desenhos, imagens, etc. Frequentemente, o próprio documento não é armazenado diretamente em um sistema deste tipo, mas, é representado por elementos substitutos, como por exemplo, palavras-chave. Um sistema de recuperação de informações deve suportar certas operações básicas, como modificar um documento na base de dados, apagá-lo, procurá-lo e apresentá-lo ao usuário. Nesta seção trataremos justamente da busca, recuperação e apresentação de informações.

### 5.1 Busca e Recuperação de Informações

A busca de informações em nosso sistema é feita utilizando-se palavras-chave. As palavras-chave devem conter informações semelhantes às usadas durante a indexação dos documentos. Devido à característica particular dos documentos armazenados na base de dados, não são admitidas entradas parciais ou diferentes das utilizadas na indexação, isto é, a entrada da palavra-chave deve seguir a mesma sintaxe usada anteriormente na indexação. A palavra-chave deve ser fornecida ao mecanismo de busca através da digitação no teclado de um terminal. A operação de procura envolve a comparação da palavra-chave com os registros armazenados na base de dados. O objetivo é determinar se existe dentre os registros da base de dados, um registro igual àquele da palavra-chave. Se a palavra-chave coincidir com um registro, então, um ponteiro para a imagem do cheque deve ser retornado.

Para encontrarmos um determinado cheque armazenado na base de dados devemos inicialmente entrar com a palavra-chave no sistema. Os três primeiros dígitos que correspondem ao número do banco são selecionados, seu endereço hash é calculado e é feita uma busca na tabela hash dos bancos. Sendo encontrada a informação que coincida com a primeira parte da palavra-chave, ela fornecerá um ponteiro que apontará para uma nova tabela hash, desta vez, no nível das agências. Agora, são selecionados os quatro dígitos seguintes da palavra-chave, correspondendo ao número da agência, o endereço hash é calculado e é feita uma busca na tabela hash das agências. Sendo encontrada a informação que coincida com a segunda parte da palavra-chave, ela fornecerá novamente um ponteiro para uma nova tabela hash, desta vez no nível de informação das contas. Dependendo do número do banco, representado pela primeira parte da palavra-chave, uma certa quantidade de dígitos que correspondem ao número da conta são selecionados. Para este número repetimos as operações de cálculo do endereço hash e busca, obtendo, assim, um novo ponteiro para a tabela hash dos cheques. São selecionados os últimos seis números da palavra-chave, que correspondem ao número do cheque, o endereço hash é calculado e teremos um

<sup>1</sup>Para o sistema em Windows, especificado na seção 6 e uma base de dados com capacidade para 1.000.000 de cheques.

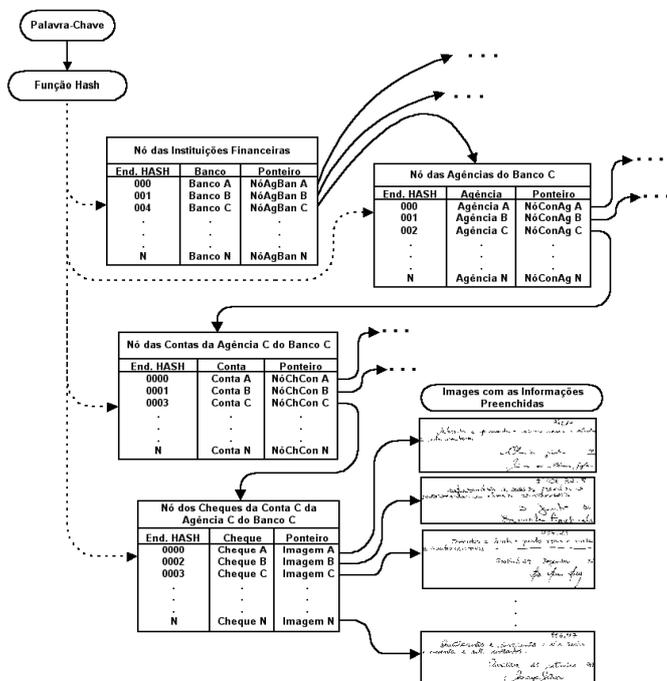


Figura 9: Buscando informações na base de dados

novo ponteiro que desta vez aponta diretamente para a imagem desejada. Na figura 9 exemplificamos uma busca ao longo dos diferentes níveis da base de dados.

Se a palavra-chave estiver armazenada nas tabelas ela será tanto encontrada pelo seu endereço hash ou ao final de sucessivas aplicações da função *rehash* (equação 2). Se o endereço hash estiver vazio, ou se a busca apontar para um endereço vazio devemos concluir que esta palavra-chave não está armazenada na tabela e, portanto, o documento a ela associado não está presente na base de dados. Se a palavra-chave estiver presente nas tabelas, o algoritmo de busca utilizará a função *rehash* para encontrá-la exatamente o mesmo número de vezes quanto as necessárias para incluí-la nas tabelas, pois, a busca utiliza os mesmos caminhos da inserção. Se a busca for por uma palavra que não está presente, a procura levará exatamente tantas provas quanto seriam necessárias para incluí-la na tabela.

## 5.2 Geração de Informações Impressas

Para reconstruirmos a imagem original de um cheque bancário, além do padrão de fundo e das informações preenchidas, devemos incluir também as informações impressas. Como estas informações não estão presentes nas imagens armazenadas na base de dados, devemos gerar uma imagem que contenha informações impressas similares às que foram eliminadas da imagem original do cheque pelo sistema de extração de informações. Para gerarmos esta imagem utilizamos a linha MICR e as informações do cliente, do banco e da agência que também estão armazenadas na base de dados em formato ASCII. A saída do módulo de geração deve ser uma imagem contendo somente as informações impressas presentes em um cheque bancário. Na prática, as informações que serão impressas no cheque já fazem parte do banco de dados das instituições financeiras, pois, no momento da confecção de um cheque bancário bem como na operação de compensação, tais informações devem estar disponíveis. Como um exemplo real podemos citar os novos ATMs, que incorporam às funções normais de um caixa eletrônico a função de confecção de folhas de cheque. Estas máquinas possuem internamente folhas somente com o padrão de fundo pré-impresso e no mo-

Informação	Nome da Fonte	Estilo	Efeito	Tamanho	Coordenada X	Coordenada Y
------------	---------------	--------	--------	---------	--------------	--------------

Figura 10: Sintaxe do arquivo de informações armazenada na base de dados

Figura 11: Exemplo de uma imagem contendo as informações impressas geradas pelo algoritmo.

mento da solicitação do cliente, o ATM acessa o banco de dados da instituição financeira e faz uma busca e leitura das informações necessárias para a impressão das folhas, gerando assim, um cheque bancário normal. Estas mesmas informações já disponíveis podem ser usadas para criarmos uma imagem que contenha somente as informações impressas presentes em um cheque bancário. Simplesmente é feita a leitura destas informações no banco de dados e a partir delas geramos uma imagem. As informações impressas consistem em uma seqüência de caracteres que contém o número da conta, o número do cheque, o nome do cliente e outras informações particulares a respeito do banco, da agência e do cliente, além de linhas horizontais e verticais. Apesar de uma boa parte das informações impressas serem únicas para todos os clientes de um mesmo banco optamos por não armazená-las na base de dados na forma de uma imagem, mas sim gerá-las no momento da reconstrução da imagem do cheque. Este procedimento propicia um melhor *matching* entre os componentes impressos do cheque no que diz respeito a suas posições relativas.

### 5.2.1 Informações Armazenadas na Base de Dados

As informações necessárias para gerarmos uma imagem com as informações impressas são apenas os elementos impressos nos cheques e as posições onde eles devem aparecer. As informações podem ser divididas em três partes distintas: informações do banco, informações da agência e informações do cliente. Teremos, portanto, três arquivos distintos na base de dados.

Além das informações propriamente ditas devemos conhecer também a posição onde cada elemento deve aparecer. Assim, cada elemento deve vir acompanhado de uma coordenada  $(x, y)$ . Além disso, devemos também especificar o tamanho e o tipo do caractere que será apresentado. Portanto, nosso arquivo deverá conter informações como: o tipo, o tamanho e o efeito da fonte. O nosso arquivo de dados terá a sintaxe apresentada na figura 10.

### 5.2.2 Algoritmo para Geração das Informações Impressas

Para a geração da imagem com as informações impressas, inicialmente as informações codificadas na linha MICR são lidas. É feita uma busca na base de dados, e sendo válidas as informações fornecidas pela linha MICR, um espaço temporário na memória é alocado para a criação de uma imagem. A base de dados é acessada e todas as informações pertinentes são lidas e direcionadas para esta área de memória. Após a leitura da linha MICR e identificação de algumas variáveis, o algoritmo de

geração da imagem busca as informações na base de dados, as codifica para o tipo de fonte pertinente, ajustando os efeitos, tamanho e demais propriedades, verifica a posição onde elas aparecerão dentro da imagem e então escreve na memória alocada. Na figura 11 apresentamos um exemplo de imagem gerada pelo algoritmo.

### 5.3 Reconstrução de Imagens

Após a busca e recuperação da imagem contendo as informações preenchidas na base de dados é alocado um espaço na memória onde a imagem é temporariamente armazenada. Porém, esta imagem contém somente as informações introduzidas pelo usuário no momento do preenchimento do cheque. Como vimos anteriormente, o restante das informações, consideradas redundantes para efeitos de armazenamento, foram eliminadas. No momento da visualização do cheque desejamos não apenas mostrar as informações preenchidas, mas a imagem original do cheque com as informações impressas e o padrão de fundo. Assim, devemos realizar o procedimento oposto ao realizado pelo módulo de processamento e extração de informações, ou seja, devemos acrescentar as demais informações. As amostras destas informações estão armazenadas no banco de dados. No caso do padrão de fundo, temos uma amostra que é comum para todos os cheques de um mesmo banco. Com relação às informações impressas, como vimos no item anterior, temos a disposição no banco de dados os parâmetros e instruções para gerá-las.

Para a reconstrução da imagem do cheque um procedimento inverso ao apresentado no sistema de extração de informações (Koerich e Lee, 1997; Lee et al., 1997) é adotado, isto é, é realizada uma operação de adição morfológica entre as imagens, porém, não é necessário realizar qualquer ajuste de posição, pois, as imagens disponíveis na base de dados já foram devidamente normalizadas antes de serem armazenadas. Assumindo que  $D = \{d(x, y)\}$  é a imagem resultante da operação de adição,  $A = \{a(x, y)\}$  é a imagem com as informações preenchidas e  $C = \{c(x, y)\}$  é uma amostra do padrão de fundo, uma operação típica de adição entre estas imagens é definida por:

$$d(x, y) = c(x, y) + a(x, y) \quad (6)$$

Como estamos trabalhando com imagem em níveis de cinza, os pixels da imagem resultante devem assumir valores entre 0 e 255, onde zero representa o pixel preto e 255 o pixel branco. Assim, a operação de adição deve ser modificada para que os pixels resultantes possuam valores dentro desta faixa. A operação de adição modificada será dada por:

$$d(x, y) = \begin{cases} c(x, y) & \text{se } c(x, y) > a(x, y) \\ a(x, y) & \text{caso contrário} \end{cases} \quad (7)$$

A operação de adição é realizada em duas etapas. Inicialmente são adicionadas as imagens do padrão de fundo e das informações preenchidas, e em seguida a imagem resultante desta operação é novamente adicionada, agora com a imagem contendo as informações impressas geradas. Seja  $F = \{f(x, y)\}$  a imagem resultante da segunda operação de adição,  $D = \{d(x, y)\}$  a imagem resultante da primeira operação de adição e  $E = \{e(x, y)\}$  a imagem com as informações impressas geradas, a segunda

Tabela 2: Limites práticos para o sistema de armazenamento e recuperação de informações.

Elemento	Limite
Bancos	100
Agências por Banco	1.000
Contas Corrente por Agência	1.000
Cheques por Conta Corrente	10.000

operação de adição modificada será dada por:

$$f(x, y) = \begin{cases} d(x, y) & \text{se } d(x, y) > e(x, y) \\ e(x, y) & \text{caso contrário} \end{cases} \quad (8)$$

## 6 RESULTADOS EXPERIMENTAIS

O método de armazenamento, recuperação e reconstrução proposto foi testado em imagens digitalizadas de cheques bancários reais. Estes cheques foram digitalizados com uma resolução de 200 dpi e em 256 níveis de cinza. As imagens digitalizadas fazem parte da base de dados formada pelo LRPRC<sup>2</sup> e que está disponível em CD-ROM padrão ISO 9660.

Para avaliar a relação desempenho/hardware, foram desenvolvidas duas versões do sistema, uma para Windows NT (PC) e outra para Solaris (SUN). Ambas foram implementadas usando linguagem C/C++ ANSI.

Para avaliarmos o desempenho das operações de armazenamento e recuperação de informações consideramos quatro fatores: o tamanho da base de dados, o tempo de processamento, a taxa de compressão e a qualidade da imagem reconstruída. O tamanho da base de dados está relacionado com a quantidade de informações que ela pode armazenar e que influencia diretamente as operações de armazenamento e recuperação. O tempo de processamento representa o tempo que o computador leva para executar cada operação. Estas operações incluem a compressão e a indexação da imagem, a inserção da imagem na base de dados, a busca e recuperação de informações, e a reconstrução da imagem original do cheque bancário. A taxa de compressão representa a relação entre o tamanho da imagem original do cheque e o tamanho da imagem armazenada. Finalmente a qualidade da imagem reconstruída representa a qualidade visual das imagens apresentadas pelo sistema.

### 6.1 Construção da Base de Dados

Para verificarmos a eficiência dos métodos propostos para o armazenamento e recuperação de cheques bancários construímos uma base de dados utilizando os cheques bancários disponíveis bem como levamos em conta a capacidade da máquina onde o sistema foi implementado. Assim, como as informações extraídas dos cheques bancários ocupam em média 7kB de espaço em disco (figura 5), limitamos a 1.000.000 o número máximo de cheques bancários a serem armazenados. Tendo em vista este limite prático, foram criadas diferentes configurações de dados, respeitando, porém, aos limites dados pela tabela 2.

Estes números estão próximos da realidade atual do sistema bancário brasileiro. Segundo a Federação Brasileira das Associa-

<sup>2</sup>Laboratório de Reconhecimento de Padrões e Redes de Comunicação, www.lrprc.fee.unicamp.br .

Tabela 3: Tempo médio e pior tempo.

Operação	Solaris		Windows	
	TM(us)	PT(ms)	TM(us)	PT(ms)
Inserção	7,46	10,0	74,83	170,2
Busca	6,19	11,0	66,38	60,1

ções de Bancos (FEBRABAN), o país possui atualmente 194 bancos, 16.158 agências e 44.7 milhões de contas correntes ativas. Ainda segundo a FEBRABAN, são compensados mensalmente pelas Câmaras de Compensação por volta de 150 milhões de cheques bancários.

Porém, como somente dispúnhamos de 200 cheques bancários, as imagens foram replicadas várias vezes, porém indexadas com palavras-chave diferentes geradas aleatoriamente, mas obedecendo os limites dados pela tabela 1. Os dados foram distribuídos de forma controlada e também aleatoriamente ao longo dos 4 níveis de informação.

Durante a inserção dos cheques na base de dados, pode ocorrer que o endereço hash calculado já esteja ocupado por outro cheque. Este evento é chamado de colisão. Para solucionar o problema de colisão, foram testadas várias técnicas (*linear probing*, *rehashing* e *open addressing*, etc), sendo escolhida a técnica *rehashing*, pois, foi a técnica que forneceu os menores tempos de resposta tanto na busca de cheques presentes, como também de cheques não presentes na base de dados.

## 6.2 Análise do Desempenho

Para analisar o desempenho do sistema de armazenamento e recuperação de cheques bancários foram realizados experimentos com diferentes configurações para a base de dados. Para cada configuração foram medidos os seguintes fatores: tempo médio para inserção/busca; pior tempo para inserção/busca; tempo de resposta para um cheque não existente na base de dados; e número máximo de colisões.

A análise do desempenho foi realizada em dois sistemas: PC Pentium II 350MHz com 128MB de memória RAM, 512Kb de cache, disco rígido Quantum Fireball Plus KA com 512Kb de buffer e tempo de busca médio de 8,5ms, e sistema operacional Windows NT 4.0; SUN UltraSparc II 296MHz com 768MB de memória RAM, 2MB de cache, disco rígido Seagate Barracuda com 1MB de buffer e tempo de busca médio de 9.7ms e sistema operacional Solaris 5.6.

Levando-se em conta a impossibilidade de realizar a etapa de extração de informações para todos os cheques e também a indisponibilidade de um grande número de amostras, replicamos os número de amostras disponíveis e associamos a cada amostra um número de identificação gerado aleatoriamente de acordo com os limites dados pela tabela 1. Estas amostras foram armazenadas em disco rígido. Assim, o sistema de informações faz a leitura dos cheques a partir do disco rígido e inicia a geração da base de dados segundo a estrutura proposta. Os tempos médios (TM) e o pior tempo (PT) de cada operação são dados na tabela 3. A figura 12 apresenta a operação de inserção de cheques na base de dados, onde o tempo médio de inserção é dado em função do número de cheques inseridos na base de dados. O sistema implementado em Solaris apresenta um tempo médio 10 vezes menor em relação ao sistema implementado em Windows.

Tabela 4: Exemplos de conjuntos de dados utilizados na avaliação de desempenho.

Elemento	Configurações				
	1	10	100	1.000	10.000
Banco	1	1	1	10	100
Agência	1	10	100	10	10
Conta Corrente	100	100	100	10	10
Cheque	10.000	1.000	100	1.000	100

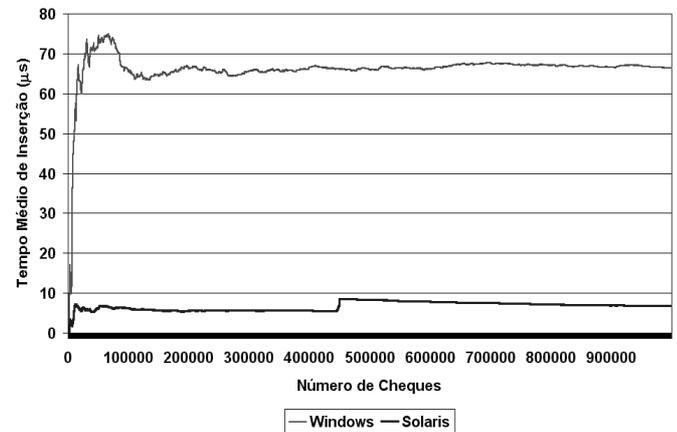


Figura 12: Tempo médio de inserção de cheques na base de dados

Devido a limitação do espaço em disco, a quantidade máxima de amostras armazenadas na base de dados foi limitada a 1.000.000 de cheques. Este conjunto de amostras foi dividido ao longo dos quatro níveis de maneira ordenada e também aleatoriamente. A tabela 4 apresenta algumas das configurações utilizadas.

Apesar dos limites impostos pela capacidade do hardware, é possível avaliarmos mais profundamente o desempenho do sistema para quantidades maiores de informação. Para tal, foram introduzidas algumas pequenas modificações no sistema de modo que todos os cheques inseridos na base de dados apontassem para uma mesma imagem de cheque. Desta maneira conseguimos eliminar o limite de espaço em disco e assim pudemos avaliar o desempenho do sistema para 10.000.000 de cheques. O tempo de resposta em função do número de cheques é apresentado na figura 13. Note que esta modificação não influi significativamente nos tempos de resposta do sistema para valores menores que 1.000.000 (figura 12).

O tempo de resposta do sistema para a busca de um cheque não presente na base de dados é em média similar ao tempo de busca de um cheque presente na base de dados (70µs), porém com uma variância um pouco maior ( $\pm 20\mu s$ ).

## 6.3 Desempenho dos Algoritmos de Extração, Recuperação e Reconstrução de Imagens

Para termos uma visão global do desempenho do sistema realizamos alguns experimentos incluindo o estágio de extração de informações. Para o conjunto de dados testado avaliamos também as operações de extração de informações (realizada pelo primeiro estágio do sistema), recuperação e reconstrução da imagem original do cheque. O gasto computacional para cada operação

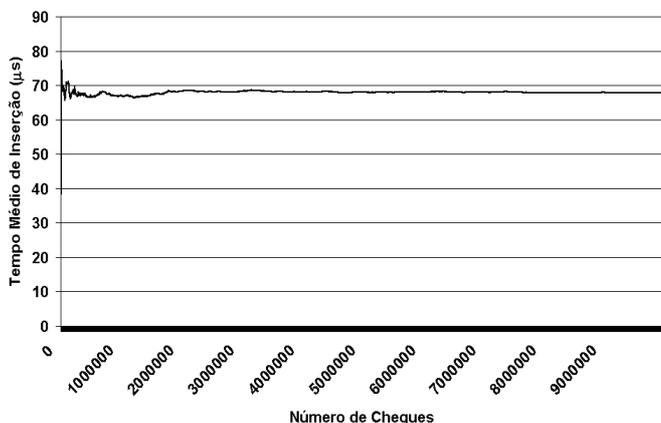


Figura 13: Tempo médio de inserção para 10 milhões de cheques (Windows)

Tabela 5: Tempo médio de cada operação.

Operação	Tempo Médio (s)
Extração de Informações	10,4
Recuperação	0,81
Reconstrução	1,21

é apresentado na tabela 5<sup>3</sup>.

#### 6.4 Avaliação da Imagem Reconstruída

A qualidade visual das imagens reconstruídas depende em grande parte das informações que foram utilizadas para formá-las, isto é, as imagens e informações disponíveis na base de dados. Para o conjunto de 200 imagens testadas obtivemos imagens reconstruídas similares às dos cheques bancários originais. O único detalhe que se torna evidente é o aumento da intensidade das partes preenchidas, que nas imagens originais geralmente não possuem um nível uniforme. Nas imagens reconstruídas, as imagens com as informações preenchidas, que sofreram um processo de binarização em alguma parte do processamento para a extração de informações, passam a ter um nível de cinza igual a 0 (preto). Este detalhe, porém, dá um maior contraste às imagens reconstruídas em relação às originais. Além disso, notamos ausência da linha MICR que, devido à indisponibilidade da fonte correspondente, não pôde ser incluída nas imagens reconstruídas. Na figura 14 apresentamos uma imagem de um cheque bancário original e a correspondente imagem reconstruída a partir das informações recuperadas na base de dados.

#### 6.5 Avaliação do Método de Compressão

Para avaliarmos a eficiência do processo de armazenamento em relação ao espaço ocupado, comparamos a imagem original digitalizada do cheque bancário com a imagem final que sofreu uma redução de tamanho. Para avaliarmos o resultado do método de compressão, utilizamos o fator de desempenho, que representa a eficiência da compressão ( $CE$ ), conforme definido por Ranganathan et al. (1994) e que é dado por:

$$CE = \frac{ois - cis}{ois} \cdot 100 \quad (9)$$

<sup>3</sup>Dados para o sistema em Windows.



Figura 14: Amostra de um cheque bancário original e de um cheque bancário reconstruído.

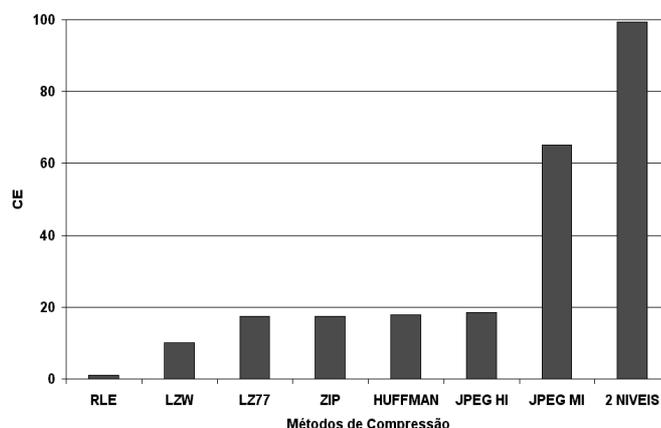


Figura 15: Comparação da eficiência de alguns métodos de compressão

onde  $ois$  é o tamanho da imagem original e  $cis$  é o tamanho da imagem comprimida.

Para o total de 200 imagens avaliadas, obtivemos para  $CE$  um valor médio de 99,32. Para o cálculo deste  $CE$ , consideramos apenas o tamanho das imagens contendo as informações preenchidas propriamente ditas. Se acrescentarmos a cada uma destas imagens as informações e instruções necessárias para sua reconstrução, teremos um acréscimo da ordem de 500 bytes no tamanho de cada imagem. Ainda assim, a eficiência da compressão assumirá valores altos, com a  $CE$  passando a ter um valor médio de 99,27.

#### 6.6 Comparação com Outros Métodos de Compressão

O método de compressão apresentado difere um pouco dos métodos tradicionais, pois, ele não reduz somente o tamanho do arquivo, mas divide a imagem original em três (padrão de fundo, informações impressas e informações manuscritas), assim, a comparação direta com outros métodos de compressão torna-se um pouco difícil. Porém, para justificarmos a utilização de tal método, comparamos a eficiência do método proposto com outros algoritmos típicos de compressão. Para o propósito de comparação consideramos os métodos RLE, LZW, LZ77, ZIP, Huffman e JPEG com perdas. Na figura 15 são apresentados os valores de  $CE$  fornecidos por cada algoritmo bem como pelo método de compressão proposto.

### 7 CONCLUSÃO E DISCUSSÃO

Uma atividade importante em todo o sistema bancário é o armazenamento de cheques bancários. Entretanto esta tarefa ainda é feita de uma forma predominantemente manual e os métodos utilizados para o armazenamento, indexação, manipulação, re-

cuperação e visualização dos documentos são ineficientes e dispendiosos. Este trabalho apresenta uma solução para o armazenamento e recuperação de cheques bancários através de computadores em contraste com os métodos atualmente utilizados pelas instituições financeiras que se baseiam em técnicas de microfilmagem. Dos cheques bancários são extraídas somente as informações relevantes sob o ponto de vista do armazenamento, que posteriormente são comprimidas, indexadas e armazenadas utilizando-se uma base de dados especialmente projetada para este fim. As informações podem ser recuperadas da base de dados e imagens similares às imagens originais dos cheques bancários podem ser obtidas.

O sistema proposto apresenta como principais vantagens a transformação de um processo eminentemente manual em um processo predominantemente automático, com a utilização de computadores e dispositivos de aquisição de imagens e dados. A padronização proporcionada pela digitalização propicia uma excelente maneira de racionalização do trabalho, podendo eliminar etapas como transporte, circulação e manuseio de documentos em papel ou em filme, reduzindo o fluxo de papel circulante, pois, as imagens dos cheques podem ser transportadas através de redes de comunicação com grande eficiência, rapidez e segurança. Além disso podemos ainda fornecer na saída do sistema uma imagem mais próxima da imagem real do cheque bancário, ou seja, podemos utilizar amostradas coloridas dos padrões de fundo dos cheques. Não é necessária nenhuma mudança adicional para a utilização de padrões de fundo coloridos, o funcionamento do sistema também não é alterado a não ser pelo espaço maior que os mesmos ocuparão na base de dados. A inexistência de outros trabalhos que utilizem métodos computacionais para o armazenamento deste tipo de documento inviabiliza uma avaliação mais rigorosa dos resultados obtidos pelo método proposto através da comparação com os obtidos por métodos similares.

Para testarmos o sistema proposto utilizamos um número limitado de amostras. Com o aumento no número de cheques armazenados na base de dados, e o conseqüente crescimento do tamanho das tabelas hash, espera-se um aumento nos tempos envolvidos nos processos de armazenamento, busca e recuperação de informações. Ainda assim, o desempenho, segurança e eficiência do sistema deverão ser bem superiores aos oferecidos pelos métodos e sistemas atualmente utilizados. No que se refere ao tamanho ocupado pela imagem final, fizemos uso das características particulares das imagens de cheques bancários eliminando a maior parte das informações redundantes, conseguindo, assim, um método de compressão com eficiência superior a 99%. Se comparado com a eficiência dos métodos microfotográficos atualmente utilizados, a eficiência de compressão apenas foi aumentada em 1%, porém, a transformação no processo de armazenamento e recuperação, passando de uma forma predominantemente manual para uma forma totalmente automática, deve ser levada em conta. Quando comparado com outros métodos computacionais de compressão, os resultados obtidos pelos diferentes algoritmos indicam a eficiência do método proposto para a compressão de imagens de cheques bancários, conseguindo um ganho superior a 30% em relação ao método JPEG, que possui a melhor eficiência dentre os métodos comparados.

## REFERÊNCIAS

Akiyama, T. e N. Hagita (1990). Automated Entry for Printed Documents, *Pattern Recognition*. Vol.23, No.11, pp.1141-1154.

Arps, R. B. e T. K. Truong (1994). Comparison of International Standards for Lossless Still Image Compression. *Proceedings of IEEE*, Vol.82, No.6, 889-899.

Brasil, B. C. (1983). Resolução número 885, Manual de Normas e Instruções, Banco Central do Brasil, Brasília.

Casey, R., D. Ferguson, K. Mohiuddin e E. Walach (1992). Intelligent Forms Processing System. *Machine Vision and Applications*, Vol.5, pp.143-155.

Cormen, T. H., C. E. Leiserson e R. Rivest (1990). *Introduction to Algorithms*. The Massachusetts Institute of Technology, Massachusetts, USA

Csabharwal, C. L. e S. K. Bhatia (1995). Perfect Hash Table Algorithm for Image Databases Using Negative Associated Values. *Pattern Recognition*, Vol.28, No.7, pp.1091-1101.

Frakes, W. B. e R. B. Yates (1992). *Information Retrieval: Data Structures & Algorithms*. PTR Prentice-Hall Inc.

Howard, P. G. e J. S. Vitter (1994). Arithmetic Coding for Data Compression. *Proceedings of IEEE*, Vol.82, No.6, pp.857-865.

Koerich, A. L. e L. L. Lee (1997). Compression of Bank Check Images Based on Layout Knowledge. *Int. Conf. on Acoustics, Speech and Signal Processing*, Vol. 4, pp.3185-3188.

Lee, L. L., M. E. Lizarraga, N. R. Gomes e A. L. Koerich (1997). A Prototype for Brazilian Bankcheck Recognition. *Int. Journal of Pattern Recognition and Artificial Intelligence*, Vol.11, No.4, pp.549-569.

Ranganathan, N., S. G. Romaniuk e K. R. Namuduri (1994). A Lossless Image Compression Algorithm Using Variable Block Size Segmentation. *Int. Conference on Pattern Recognition*. Jerusalém, pp.40-44.

Satoh, S., A. Takasu e E. Katsura (1995). An Automated Generation of Electronic Library Based on Document Image Understanding. *3rd Int. Conference on Document Analysis and Recognition*, Montréal, pp.163-166.

Yasuda, Y., M. Dubois e T. S. Huang (1980). Data Compression for Check Processing Machines. *Proceedings of the IEEE*, Vol.68, No.7, pp. 874-885.