1. Publicação nº	2. Versão	3. Data	5. Distribuição
INPE-2847-PRE/389		*Agosto, 1983*	☐ Interna ☒ Externa ☐ Restrita

4. Origem	Programa
DIN/DPD	*DESOFT/INFORMÁTICA*

6. Palavras chaves - selecionadas pelo(s) autor(es)

COMPRESSÃO DE DADOS
TEORIA DA INFORMAÇÃO
BANCO DE DADOS

7. C.D.U.: *681.3.016*

8. Título *INPE-2847-PRE/389*

UM SISTEMA DE COMPRESSÃO DE DADOS

10. Páginas: *14*

11. Última página: *12*

12. Revisada por

Luiz Alberto Vieira Dias

9. Autoria *Didier Jean Charpentier*
Orion de Oliveira Silva

13. Autorizada por

Nelson de Jesus Parada
Diretor Geral

Assinatura responsável Orion de Oliveira Silva

14. Resumo/Notas

Com a difusão dos Computadores, grandes programas, grandes blocos de textos, imagens digitalizadas de satélites e bancos de dados dos mais diversos tipos passaram a ser armazenados e acessados por computador. Embora o custo de armazenagem venha caindo constantemente, continua a haver necessidade de técnicas de compactação que economizem espaço de memória (primária, secundária e terciária). Neste trabalho, apresentam-se três métodos de compactação aperfeiçoados pelos autores, sem perda de informação, isto é, aqueles nos quais os textos descompactados são idênticos aos textos originais. Fornecem-se detalhes sobre a implementação dos algoritmos e apresentam-se resultados comparativos.

15. Observações

Este trabalho será apresentado no 6º CNMAC, que realizar-se-á no ITA, São José dos Campos - SP.

UM SISTEMA DE COMPRESSÃO DE DADOS

Didier Jean Charpentier
Orion de Oliveira Silva

Instituto de Pesquisas Espaciais - INPE
Conselho Nacional de Desenvolvimento Científico e Tecnológico-CNPq
Caixa Postal 515 - 12200 - São José dos Campos - SP

RESUMO

Com a difusão dos Computadores, grandes programas, grandes blocos de textos, imagens digitalizadas de satélites e bancos de dados dos mais diversos tipos passaram a ser armazenados e acessados por computador. Embora o custo de armazenagem venha caindo constantemente, continua a haver necessidade de técnicas de compactação que economizem espaço de memória (primária, secundária e terciária). Neste trabalho, apresentam-se três métodos de compactação aperfeiçoados pelos autores, sem perda de informação, isto é, aqueles nos quais os textos descompactados são idênticos aos textos originais. Fornecem-se detalhes sobre a implementação dos algoritmos e apresentam-se resultados comparativos.

ABSTRACT

With the spreading of computer technology, large programs, large blocks of text, digital satellite imagery and a wide variety of data bases have been stored and retrieved by computers. While the storage cost has been dropping continuously, the need for compression techniques for saving memory space (primary, secondary and terciary) still exists. Three compression methods improved by the authors, without information loss, are presented, i.e., those in which the decompressed texts are identical to the original ones. Details about the algorithm, implementations will be given, as well as comparative results.

I - INTRODUÇÃO

Com a difusão dos computadores, grandes programas, grandes blocos de textos, imagens digitalizadas de satélites e banco de dados dos mais diversos tipos passaram a ser armazenados e acessados por computador. Embora o custo de armazenagem venha caindo constantemente, continua a haver necessidade de técnicas de compactação que economizem espaço de memória (primária, secundária e terciária). Neste trabalho, apresenta-se um sistema de compactação de uso geral que foi implementado no Burroughs B-6800 do Instituto de Pesquisas Espaciais (INPE), com as seguintes opções: a) compactação normal na qual se usa o método de Bruce Hans (Hans, 1974), que foi melhorado pelos autores; b) compactação que exclui os brancos, sendo o restante compactado pelo método de Bruce Hans melhorado; c) compactação das diferenças (Gonzalez, 1977), onde se mapeia o conjunto de "n" números inteiros $x_1, x_2, \ldots, x_n$ em um novo conjunto $x_1, x_2-x_1, \ldots, x_n-x_{n-1}$, e, em seguida, estas diferenças são compactadas pelo método de Bruce Hans. Foram feitas comparações entre os três métodos, a), b) e c), para textos em ALGOL, Francês e Imagens do satélite LANDSAT. Foram introduzidos por simulação, erros aleatórios em textos compactados, verificando que eles não se propagavam no texto, apenas deslocavam as posições dos elementos do texto a partir da ocorrência do erro. Para resolver esse problema de deslocamento, foram introduzidos elementos delimitadores para que um erro, ao ocorrer em um bloco, deslocasse apenas os elementos daquele bloco, definido pelos delimitadores. São dados a seguir detalhes sobre a implementação dos algoritmos, resultados comparativos e instruções sobre a utilização do sistema.

2 - MÉTODO DE COMPRESSÃO DE UMA MANEIRA GERAL

Os métodos de compressão podem ser divididos em duas categorias:

2.1 - Primeira Categoria: Métodos Dependentes do Conteúdo

Nesta categoria, a compressão é direcionada para uma da da aplicação, dependendo do conteúdo dos dados, por exemplo: supres são de caracteres, supressão de itens repetidos em bancos de dados, conversão da notação humana para uma mais compactada, substituição de itens de dados usados comumente, evitação de espaços vazios nos arqui vos, substituição de textos em uma linguagem natural, e compressão de dados ordenados. Todos estes itens são tratados em Silva (1980).

2.2 - Segunda Categoria: Métodos Independentes do Conteúdo

Os métodos desta categoria podem ser usados para muitas aplicações e implementados em nível de "software", "hardware" ou em microprogramação.

A eficiência das técnicas definidas na primeira catego ria é altamente dependente da natureza dos dados. Algumas das técni cas são inteiramente dependentes da aplicação, tais como a técnica que será apresentada mais adiante para compressão de imagens (distância entre os "Pixels").

Um caminho, geralmente aplicável à compressão de quais quer dados armazenados ou à transmissão desses dados, é o uso de for mas mais eficientes de codificação de caracteres.

2.2.1 - Método de Bruce-Hans

Este método de compressão de dados foi publicado por Hans (1974), tendo sido testado e aperfeiçoado pelos autores deste tra balho.

O método consiste no seguinte:

i) Frequentemente, a gravação de textos, contendo muitas listas de símbolos repetidos, ocupa muito espaço na memória. O resultado é que se desperdiça muito espaço por métodos comuns de armazenamento. A técnica aqui utilizada, além de armazenar os símbolos não repetidos, reconhece listas de símbolos repetidos e as trata de modo especial.

Exemplo 1. Usam-se dois "bytes" [x|y], onde "x" corresponde ao código do símbolo, e "y", o número de vezes que o símbolo é repetido.

ii) Os símbolos não repetidos são codificados em grupos de tamanho fixo, como o número inteiro correspondente à maior palavra da máquina.

A técnica de compressão utilizada consiste em:

a) ler a lista de símbolos de entrada (cartão, fita, disco);

b) construir o dicionário de símbolos (DS);

c) codificar os símbolos usando o DS;

d) comprimir os dados de entrada.

A construção do dicionário de símbolos é particular a cada computador.

Seja B o comprimento do dicionário primário e Pi a posição do i-ésimo símbolo de um dado grupo, onde ($1 \leq i \leq N$) e ($1 \leq Pi \leq B-1$) no DS. A B-ésima posição é reservada para um caractere que se denominou de "caractere de fuga", que permite a extensão do dicionário. O grupo de posições P_1, P_2, ..., P_n é codificado como um número de ponto fixo: $P_1*B^{n-1} + P_2*B^{n-2} + Pn-1* B^1 + Pn*B^0$.

Se a B-ésima posição não for reservada, o valor $B*B^i$ pode aparecer causando ambiguidade na codificação.

O caractere de fuga significa o símbolo seguinte à posição na faixa B+1 até 2B-1. Mais do que um caractere de fuga pode ser usado para a extensão do comprimento do dicionário.

Seja "p" a posição de um dado símbolo no DS; então, o símbolo é codificado com $\lfloor p/B \rfloor$ caracteres de fuga, seguidos de mod (p,B), onde $\lfloor y \rfloor$ significa a parte inteira de y. O caractere de fuga é codificado com o zero.

Exemplo 2. Seja B=8 o comprimento do dicionário primário e N=4 (os símbolos são codificados em grupos de 4).

Considerem-se os símbolos a serem codificados como tendo as posições 5, 7, 10, 15, 25 e 9 no DS. Estes símbolos são codificados em 3 números de ponto fixo:

- o primeiro tem o valor $5*8^3+7*8^2+0*8+2*8^0=3010$;
- o segundo tem o valor $0*8^3+7*8^2+0*8+0*8^0=448$;
- o terceiro tem o valor $0*8^3+1*8^2+0*8+1*8^0=65$.

Os símbolos 10, 15, 25 e 9 são codificados como (0,2), (0,7), (0,0,0,1) e (0,1), respectivamente.

Neste método, o usuário do sistema tem controle sobre os valores de B e N. Estes valores são selecionados para minimizar a quantidade média do número de bits / informação, necessários à codificação do texto de entrada.

Para cada valor de N, existe um valor ótimo para B. Este valor ótimo é dado pelo maior inteiro positivo, tal que $B^n-1 \leq L$, onde L é o maior inteiro que pode ser armazenado numa palavra de computador; isto porque o maior inteiro a ser produzido é $= (B-1)*B^{n-1} + \ldots + (B-1) = B^{n-1}$.

O valor de B é obtido pela equação $B=\lfloor(L+1)^{1/n}\rfloor$.

A Tabela 1 mostra os valores de B e N para computadores com palavra de 16 bits, 32 bits e 48 bits.

Pode-se otimizar B e N com relação a um determinado texto, usando apenas estatísticas do texto, e tomando o cuidado de contar os caracteres repetidos como uma unidade.

Sejam K_1, K_2, ..., K_n os números de caracteres do texto codificados com 1, 2, ..., n dígitos, respectivamente. Então, o problema é encontrar com B e um N para um determinado texto, de modo que minimizem T, onde $T=K_1+2*K_2+...+n*K_n$.

TABELA I

VALORES ÓTIMOS DE B E N

L = 16 BITS		L = 32 BITS		L = 48 BITS	
VALOR DE N	VALOR DE B	VALOR DE N	VALOR DE B	VALOR DE N	VALOR DE B
5	8	5	74	5	776
6	6	6	36	6	256
7	4	7	22	7	116
8	4	8	15	8	64
9	3	9	11	9	40
10	3	10	9	10	28
11	3	11	7	11	21
12	2	12	6	12	16
13	2	13	5	13	13
14	2	14	5	14	11
15	2	15	4	15	9
16	2	16	4	16	8
17	2	17	4	17	7
18	2	18	3	18	6
19	2	19	3	19	6
20	2	20	3	20	5

Não foi encontrada, ainda, uma maneira eficiente de achar o valor de T mínimo. O que se fez, aqui, foi postular um valor de B no meio da tabela e aumentar ou diminuir este valor, até encontrar o valor ótimo.

2.2.2 - Método de Distância Entre os "Pixels"

Considerando agora o problema de compactação de imagens, sejam $x_1, x_2, \ldots, x_n$ "pixels" consecutivos, cujos valores pertencem ao conjunto A = {0, 1, ..., 255}; então, mapeia-se o conjunto de n números inteiros em um novo conjunto: $x_1, x_2-x_1, \ldots, x_n-x_n-1$ (Gonzalez, 1977). Como nas imagens a diferença de um "pixel" para o seu sucessor ou o seu antecessor se repete bastante, pode-se mapear a sequência $x_1x_2\ldots x_n$ na sequência $d_1d_2\ldots d_m$ (m>n). Embora o dicionário de símbolos para codificar os d_i ($1 \leq i \leq m$) seja maior do que o dicionário de símbolos para codificar os x_j ($1 \leq j \leq n$), existe uma grande quantidade de repetições em alguns d_i's, o que possibilita assim uma média menor de "pixel" por caractere, do que se codificasse os x_j (Exemplo 3).

Exemplo 3. Seja a seguinte sequência:

1, 2, 4, 5, 6, 7, 8, 10, 9 → x_i (não existe repetição);

1, -2, -1, -1, -1, -1, -2, 1 → d_i (-1 se repete 4 vezes).

Caractere		Código
1	→	01
2	→	001
3	→	011
4	→	0001
5	→	0011
6	→	0111
7	→	00001
8	→	00011
9	→	00111

bits por caractere = 4,33

Caractere		Código
-1	→	01
-2	→	001
1	→	011

bits por caractere = 2,22

3 - DEFINIÇÃO DO SISTEMA

Apresenta-se a seguir o sistema de compactação que foi implementado no computador Burroughs B-6800 do Instituto de Pesquisas Espaciais (INPE). O Sistema fornece três opções para o usuário: a compactação normal, na qual se usa o método de Bruce Hans como foi basicamente apresentado em 2.2.1 (opção C); a compactação que exclui os brancos sendo o restante compactado pelo método de Bruce Hans (opção CB), e a compactação das diferenças, onde se faz as diferenças como explicado em 2.2.2, estas diferenças sendo compactadas pelo método de Bruce Hans (opção CD).

3.1 - Algoritmo de Compressão

O algoritmo de compressão segue os seguintes passos:

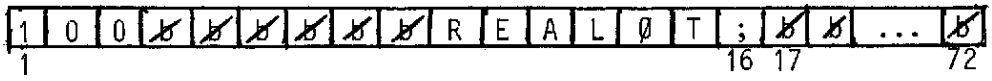
i) Leitura da opção escolhida, do nome do arquivo a ser compactado e dos parâmetros DELI.

ii) Leitura do arquivo, e cálculo das probabilidades de ocorrência de cada símbolo.

OPÇÃO CB: Além dos símbolos do arquivo, é introduzido um caractere chamado "CODE" para indicar uma sequência de "K" brancos ($3 \leq K \leq 255$). O valor de "CODE" será um caractere que não existe no arquivo a ser compactado.

Exemplo 4. O código @ não existe no arquivo.

a) Antes da compactação



b) Depois da compactação

Nessa parte, serão computadas as probabilidades de "CODE" e as dos valores de K.

OPÇÃO CD: Computação das probabilidades das diferenças dos símbolos.

iii) Escolher o caractere "CODE".
Escolher o caractere "SINCRO" que tem como objetivo delimitar a linha do arquivo, para que, numa eventual ocorrência de erros, estes erros não se propaguem, permitindo encontrar a sincronização do texto original.

iv) Ordenar o vetor das probabilidades segundo ordem decrescente.

v) Calcular os valores B e N ótimos para minimizar o volume do arquivo compactado.

vi) Calcular as tabelas: POSIÇÃO (SÍMBOLO);
SÍMBOLO (POSIÇÃO).

vii) Escrever no início do arquivo compactado o seguinte:

a) Parâmetros do arquivo original.

b) Tabela de SÍMBOLO (POSIÇÃO).

c) As opções C, CB e CD.

d) Valores de B e N.

e) Caracteres espaciais "SINCRO" e "CODE".

viii) Ler o arquivo-fonte, linha por linha. Concatenar o símbolo "SINCRO" ao fim de cada linha. Codificar, segundo a opção escolhida, e escrever o resultado no arquivo compactado.

3.2 - Algoritmo de descompressão

O algoritmo de descompressão segue os seguintes passos:

i) Ler o nome do arquivo a ser descompactado.

ii) Ler nesse arquivo o seguinte:

a) Parâmetros do arquivo original.

b) Tabela SÍMBOLO (POSIÇÃO).

c) Opção C, CB e CD.

d) Valores de B e N.

e) Caracteres especiais "SINCRO" e "CODE".

iii) Ler o arquivo compactado, linha por linha. Decodificar segundo a opção escolhida. Se houver símbolos que não existem na tabela SÍMBOLO (POSIÇÃO), então uma mensagem de erro é emitida.

Remover o caractere "SINCRO" e escrever o arquivo descompactado.

3.3 - Otimização de B e N

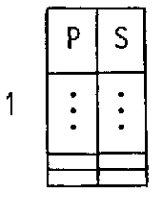
Seja um arquivo com "s" símbolos diferentes $\{S_1, S_2, \ldots, S_s\}$, com probabilidades $\{P(S_1), P(S_2), \ldots, P(S_s)\}$;

$P(S_k) \geq P(S_i)$; $K > i$;

$F(S_k) = \sum_{i=1}^{k} P(S_i)$;

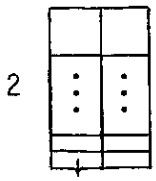
$F(S_s) = 1$.

O dicionário será dividido em "K" blocos. $\lceil R \rceil$ é o menor número inteiro, maior do que ou igual a R.



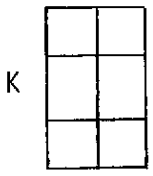
$\lfloor R \rfloor = \lceil R \rceil - 1$ se R não for inteiro.

$= R$ se R for inteiro



$K = \lceil S / B-1 \rceil$.

Seja L o mínimo de bits de uma palavra. Para codificar um símbolo do dicionário I, serão necessários $\frac{I \cdot L}{N}$ bits.



Seja $P(D_i)$ a probabilidade de um símbolo R do dicionário R do dicionário de símbolos D_i. O número médio de bits por símbolo será:

$$T = \sum_{i=1}^{k} P(P_i) \cdot \frac{i.L}{N}$$

$$= \frac{L}{N} \sum_{i=1}^{k} i\ P\ (D_i).$$

$$P(D_i) = \sum_{j=(i-1)(B-1)+1}^{i(B-1)} P(S_i) = F(i(B-1))-F((i-1)(B-1)+1),$$

$$T = \frac{L}{N} \sum_{i=1}^{k} i[F(i(B-1))-F((i-1)(B-1)+1)]$$ a ser minimizado em relação às variáveis B e N; com $B^n-1 \leq L$.

Resultados podem ser vistos no Apêndice A.

4 - CONCLUSÕES

Dada a necessidade de eficientes métodos de compressão no INPE, sugere-se a implantação definitiva do sistema aqui apresentado no computador Burroughs B6800 do INPE, uma vez que os métodos utilizados forneceram resultados satisfatórios, como mostrados no Apêndice A.

Fica aberta, ainda, a determinação de uma maneira eficiente de calcular os valores ótimos de B e N.

REFERÊNCIAS BIBLIOGRÁFICAS

GONZALEZ, R.C. *Digital image processing*. Massachusetts, Addison Wesley, 1977

HANS, B. A new technique for compression and storage of data. *Communications of the ACM*, 17(8):434-436, Aug. 1974.

SILVA, O.O. *Algoritmo de compressão de dados*. São José dos Campos, INPE, ago. 1980 (INPE-1978-RPE/223).

APÊNDICE A

TABELA COMPARATIVA DOS MÉTODOS

		nº de informações	Média	Variância	Entropia	B	N	Bits por caractere	Percentagem da redução	Compactação	Descompactação
PROGRAMA ALGOL (77220b)	C	57	104.37	4622.85	2.38	8	16	4.415	44.06	29.98	19.35
	Cb	127	61.17	8993.14	2.58	40	9	2.329	69.23	28.10	9.81
	Cd	316	255.71	2285.71	2.89	40	9	6.58	14.69	30.06	10.32
TEXTO EM FRANCÊS (57960b)	C	64	104.48	1946.51	3.24	8	16	4.38	44.10	21.19	14.64
	Cb	107	76.93	5174.54	3.24	20	11	3.11	59.94	23.41	9.16
	Cd	218	255.76	1732.64	4.67	64	8	6.71	12.42	20.74	9.84
IMAGEM LANDSAT 1 (262144b)	C	106	28.14	52.95	4.50	20	11	4.83	39.23	82.71	53.91
	Cd	158	255.05	39.74	4.00	9	15	4.69	40.67	78.21	43.28
2	C	136	26.42	103.22	4.99	27	10	5.36	32.43	87.55	55.97
	Cd	158	255.04	57.86	4.44	20	11	5.05	36.35	78.06	43.12
3	C	147	72.32	278.19	5.69	40	9	6.07	23.58	86.72	60.18
	Cd	235	255.12	305.6	5.81	64	8	6.42	18.84	79.07	39.72
4	C	166	79.50	420.82	6.00	64	8	6.38	19.66	80.15	54.13
	Cd	235	255.13	471.92	5.96	64	8	6.66	15.95	85.91	41.27
SUB IMAGEM LANDSAT (4) TRATADA											
(x 0,65)	C	103	80.46	299.71	5.90	64	8	6.32	17.60	8.56	4.31
	Cd	162	255.69	310.17	5.87	64	8	6.43	11.01	7.35	3.06
(x 0,37)	C	63	80.47	100.13	5.25	40	9	5.64	24.19	8.24	4.15
	Cd	110	255.67	137.19	5.27	40	9	5.64	24.19	7.20	3.47
(x 0,24)	C	41	80.48	40.56	4.71	20	11	4.97	34.08	8.93	4.23
	Cd	73	255.65	85.56	4.65	20	11	4.93	34.08	6.45	3.42
(x 0,14)	C	26	80.47	15.18	4.01	12	13	4.31	43.97	8.39	4.38
	Cd	45	255.64	63.53	3.94	9	15	4.20	43.97	6.64	3.39