

Compressão de Imagens Usando Redes Neurais Artificiais, Transformada Wavelet e Transformada Cosseno Discreta.

Sandro Reis Rocha Barros

*CEFET-CAMPOS – Centro Federal de
Educação Tecnológica de Campos
Rua Dr Siqueira 273, Bairro Dom Bosco,
cep. 28030-130 -Campos dos Goytacazes,
RJ, Brasil.*

sandro@cefetcampos.br

Gelson Vieira Mendonça

*UFRJ – Universidade Federal do Rio de
Janeiro
COPPE – Programa de Engenharia
Elétrica.*

*Caixa Postal 2324 – Ilha do Fundão, Rio de
Janeiro, RJ, Brasil
gelson@lps.ufrj.br*

Resumo

Este trabalho consiste na implementação de um algoritmo que aplica a Transformada Wavelet Multi-resolução numa imagem, com o objetivo de explorar as características de suas sub-bandas, e aplicar de forma individual, em cada sub-banda, a Transformada Cosseno Discreta. Em seguida, aplicar a Quantização Vetorial usando Redes Neurais Artificiais nessas sub-bandas e, por último, aplicar e testar a eficiência de Três métodos de codificação sem perda: Run Length Encoding (RLE), Lempel Ziv (LZ) e Huffman. Os testes realizados comprovaram a eficiência do método proposto ao serem verificadas as taxas de bit/pixel e as relações sinal/ruído das imagens padronizadas para teste.

1. Introdução

A compressão de imagens trabalha com o problema da representação eficiente de imagens digitais, e esse artigo resume um projeto que consiste na avaliação de algumas técnicas e métodos de compressão, que serão aplicadas em conjunto num sistema híbrido, para que se tenha uma imagem com alta taxa de compressão e com boa qualidade.

A quantidade desnecessária de dados para representar uma determinada informação, chama-se redundância.

Dentre os tipos de redundância temos: Redundância de Codificação, Redundância Interpixel e a Redundância Psicovisual. Os tipos Redundância de Codificação e Interpixel são usados por algoritmos de

compressão sem perdas como: RLE, LZ, HUFFMAN. Já a Redundância Psicovisual é usada em algoritmos de compressão com perdas como a Quantização Vetorial, a DCT e outras.

2. Codificação run length encoding(RLE)

A codificação RLE busca reduzir a redundância de codificação, e quando usada em conjunto com outros métodos, como a codificação de Huffman torna a redundância de codificação praticamente igual a zero. A idéia do algoritmo é substituir uma seqüência de símbolos iguais pelo número de ocorrências do símbolo, e o próprio símbolo [1].

Em imagens a aplicação da codificação RLE é de grande utilidade, devido ao grande número de redundância de pixels. A codificação RLE pode ser implementada através de três tipos de varredura: vertical, horizontal e ziguezague. Através destas, pode-se ter maiores ou menores taxas de compressão.

Neste trabalho foi implementada a varredura em ziguezague que é feita de forma diagonal, iniciando-se no canto superior esquerdo e indo até o canto inferior direito da imagem. A varredura em ziguezague tem uma ótima eficiência na compressão em combinação com a DCT que veremos posteriormente.

3. Codificação de Lempel Ziv (LZ)

O LZ é um dos algoritmos mais usados na compressão de imagens e dados. Ele visa reduzir a redundância de codificação através do uso de dicionário de dados, e não gera perdas no arquivo compactado.

O método utilizado neste trabalho foi o LZ77 com um tamanho de janela móvel de 16 caracteres.

A seqüência dos dados comprimidos pode conter valores literais ou comandos para expansão. Os comandos para expansão são pares distância/comprimento que informam a localização, na janela LZ77, do primeiro dado de uma seqüência a ser expandida e quantos dados contém esta seqüência.

No algoritmo LZ77, quanto maior o tamanho da janela LZ77 maior será a taxa de compressão obtida, mas também maior será a complexidade do algoritmo, principalmente no que diz respeito às operações de procura nas janelas. Por isso, existe um compromisso entre estes dois fatores que deve ser levado em consideração. Uma alternativa para o uso de tamanhos maiores de janelas é utilizar tabelas *hash*, que aumentam a velocidade da procura [2].

4. Codificação de Huffman

A codificação de Huffman converte o valor do nível de pixel da imagem original para um novo código de tamanho variável, baseado nas freqüências de ocorrência na imagem. Desse modo, são atribuídos níveis que freqüentemente acontecem em códigos menores, e são atribuídos níveis que acontecem sem muita freqüência em códigos mais longos. O resultado é que a imagem comprimida exigirá menos bits globais para descrever a imagem original.

Os códigos de Huffman são atribuídos criando a árvore de Huffman que emparelha os valores de nível baseado nas suas freqüências combinadas de ocorrência. A árvore de Huffman assegura que os códigos mais longos são atribuídos aos níveis menos freqüentes. [3]

5. A transformada cosseno discreta (DCT)

A transformada é a chave para o processo de compressão: ela toma um conjunto de pontos no domínio espacial e os transforma em uma representação equivalente no domínio da freqüência. Um dos objetivos da transformada é diminuir a correlação, eliminando-se, portanto as redundâncias estatísticas.

Uma das transformadas utilizadas no processo de compressão é a Transformada Cosseno Discreta (DCT), que converte um bloco de pixels em uma matriz de coeficientes, descorrelacionando a informação da imagem. A transformada que descorrelaciona a informação de uma imagem de modo ótimo é a Transformada de Karhunen-Loève (TKL), a qual, por não ter um algoritmo rápido, não permite

implementação computacional eficiente. Por outro lado, dentre os métodos eficientes de transformação de imagens, aquele que mais se aproxima da TKL é justamente o que usa a Transformada Cosseno Discreta (DCT), que por isso foi escolhida para ser usada no sistema proposto.

Os coeficientes iniciais da transformada do bloco contêm as informações mais importantes da imagem, assim, deve-se garantir o armazenamento de certo número de coeficientes com baixos índices de posição. Por outro lado, a transformada fornece uma série de cossenos que no limite são convergentes, cujos coeficientes diminuem em amplitude conforme os índices crescem. Pode-se, portanto, quantizar os coeficientes por região, diminuindo-os e também eliminando os menos significantes, obtendo um ganho relevante na taxa de compressão.

Para aplicar a DCT em uma imagem, é necessário, para efeito de velocidade nos cálculos e para uma melhor taxa de compressão, dividir a imagem original em blocos, 4 x 4, 8 x 8 ou 16 x 16, podendo usar até blocos de 32 x 32 ou 64 x 64. Em geral, usam-se blocos 8 x 8.

O tempo requerido para computar cada elemento na DCT é fortemente dependente do tamanho do bloco. O número de multiplicações, para o caso bidimensional é proporcional a N^2 , ou seja, quando N cresce, o tempo exigido para processar cada elemento na DCT aumenta consideravelmente. A implementação da DCT, em geral, deve dividir a imagem em blocos menores, mais rapidamente computáveis. O padrão JPEG, por exemplo, usa o bloco de tamanho 8 x 8. [4]

A aplicação da DCT sobre o bloco, concentra a representação mais importante da imagem no lado superior esquerdo, enquanto as informações menos relevantes, se encontram no lado direito inferior do bloco.

6. A quantização vetorial

A quantização vetorial [5], nada mais é do que uma generalização da quantização escalar estendida ao caso em que os dados quantizados são vetores multidimensionais, ou seja, um conjunto de números reais. A QV é usada normalmente em processamento de sinais digitais, onde na maioria dos casos, os sinais de entrada já têm alguma forma de representação digital e saída desejada é uma versão comprimida do sinal original.

Na QV, os vetores podem ser representações de quaisquer tipos de padrões, sejam eles formados por amostras digitalizadas de uma forma de onda de sinal de voz, ou formados por um conjunto de bits de uma imagem, etc. A QV pode ser vista como uma maneira

de reconhecer padrões, onde um padrão de entrada (vetor de entrada) é aproximado na saída, por outro padrão pertencente a um conjunto de padrões pré-determinados, supostamente bastante representativos dos possíveis padrões da fonte de dados que se deseja quantizar. Esse conjunto de padrões (vetores) fixos e pré-determinados é chamado de *codebook* e fica armazenado tanto no codificador como no decodificador.

De uma maneira genérica, um quantizador vetorial de dimensão K e tamanho N pode ser definido como o mapeamento de vetores ou pontos de um espaço Euclidiano K -dimensional R^K sobre um subconjunto finito Y , de N vetores de saída.

Isto é,

$$QV: R^K \rightarrow Y$$

onde $Y = \{ y_i \in R^K ; i = 1, 2, \dots, N \}$ é o conjunto de padrões ou vetores de reprodução, também chamados *codewords*, que formam o alfabeto de reprodução que é chamado *codebook*. A taxa de codificação de um QV é determinada por $r = \lceil \log_2 N \rceil / K$ medida em bits por amostra, assumido que cada *codeword* seja uma seqüência binária de comprimento fixo.

Um QV de N -*codewords*, quando da etapa de construção do *codebook*, subdivide o espaço R^K , através de treinamento com vetores estatisticamente representativos da fonte que se quer quantizar. Esse sub-espaços chamados de regiões, células de Voronoi, agrupamentos, ou partições, tentam representar N classes (R_1, R_2, \dots, R_N) diferentes de vetores, levando em consideração suas localizações no espaço Euclidiano R^K , e associa um vetor de reprodução ou *codeword* $y_i = (y_{i1}, y_{i2}, \dots, y_{iK})$ para cada região. Esta *codeword* deve ser o centróide da região, isto é, o vetor que representa a média dos vetores pertencentes à mesma classe. Desta forma, qualquer vetor de entrada $X \in R^K$ que esteja situado dentro de uma região de Voronoi qualquer, será reconstruído na saída do quantizador, pelo centróide dessa região.

Após o final do treinamento, onde os centróides de cada região alcançam o seu ponto definitivo, para cada vetor “ x ” de entrada, o QV descobre a qual classe ele pertence e então ele é reproduzido na saída pelo centróide “ y ” referente aquela região.

O sistema de quantização vetorial é então formado por duas partes, um codificador e um decodificador. Para um dado vetor de entrada $x = (x_1, x_2)$, o codificador identifica a região dentro da qual o vetor está situado. Após a identificação, o codificador transmite pelo canal de informação, o índice da

codeword que representa a classe identificada. O decodificador opera mapeando o índice recebido pelo canal de informação, no “*codebook*”, e fornecendo, na saída, o vetor de reprodução, isto é, o centróide da classe em questão.

7. Redes neurais artificiais (RNA)

A capacidade de aprender através de exemplos e de generalizar a informação aprendida é, sem dúvida, o atrativo principal da solução de problemas através de RNA [6]. Elas são capazes de extrair informações não-apresentadas de forma explícita através dos exemplos. Não obstante, as RNA são capazes de atuar como mapeadores universais de funções multivariáveis, com custo computacional que cresce apenas linearmente com o número de variáveis. Outra característica importante é a capacidade de auto-organização e de processamento temporal, que aliada àquelas citadas anteriormente, faz das RNA uma ferramenta computacional interessante para se implementar a etapa de quantização da imagem, através de um Quantizador Vetorial.

O sistema proposto envolve o uso de redes tipo SOM (*Self-Organizing Maps*) desenvolvida por Teuvo Kohonen, para treinamento não-supervisionado.

8. O algoritmo proposto

A aplicação de transformadas nas imagens, seguidas de alguma forma de codificação, tem sido um caminho alternativo muito usado ultimamente.

Dentro desse propósito, para se obter sucesso no processamento de imagens, existe a necessidade de se usar uma propriedade que é a representação explícita de escala [7, 8, 9], pois esta permite que as imagens sejam analisadas em diferentes escalas, uma vez que as mesmas contêm objetos e detalhes de tamanhos diferentes que podem ser vistos de várias distâncias.

Uma forma de representar uma imagem em escalas é através da pirâmide Gaussiana de Burt e Adelson [10], que se obtém por meio de filtragem passa-baixas e sub-amostragem na imagem original e sucessivamente, nas sub-imagens daí produzidas.

Nessa representação, as diferenças entre pares de sub-imagens de escalas consecutivas da pirâmide Gaussiana formam a denominada pirâmide Laplaciana.

Outra forma de representar uma imagem em escalas é através da obtenção das suas contribuições em faixas de frequências diferentes, aplicada em codificação de imagens em 1986, por Woods e O’neil [11].

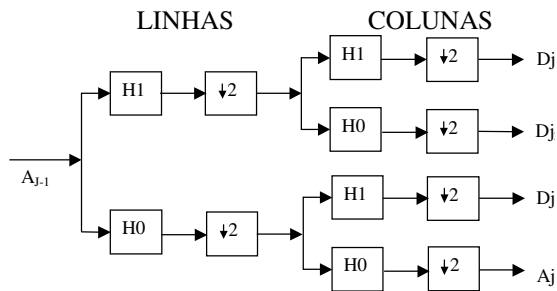
A codificação por sub-bandas divide a faixa de frequências do sinal em um determinado número de

sub-bandas, usando um banco de filtros passa-faixa. Cada sub-banda é então demodulada para a faixa de frequências base através de sub-amostragem. Esta sub-amostragem faz com que cada sub-imagem ocupe toda a faixa de frequências do sinal, com resolução mais baixa. As sub-bandas são então quantizadas e codificadas. Num primeiro estágio, a imagem é dividida em 4 sub-bandas, através da filtragem com um banco de filtros (de análise) que divide o espectro de frequências da imagem por dois, na horizontal e na vertical, seguida de sub-amostragem por dois (decimação), nas duas direções. Inversamente, a imagem original é obtida através da adição das 4 imagens geradas pela super-amostragem por dois (interpolação com zeros) de cada sub-imagem seguida de filtragem com os filtros de reconstrução (ou síntese).

O primeiro esquema de divisão das sub-bandas a ser utilizado foi a divisão uniforme [11].

O esquema utilizado no sistema proposto é o de divisão em oitavas, as quais são obtidas por uma transformada wavelet biortogonal separável.

Os algoritmos básicos de análise e síntese tomam por base os propostos por Mallat [12] conforme mostrado nas figuras 1 e 2.



H1: Filtro Passa-Altas
H0: Filtro Passa-Baixas
 $\downarrow 2$: Decimação de Linhas ou Colunas

Figura 1 - Diagrama em blocos da seqüência de análise da transformada wavelet

Os filtros utilizados foram os mesmos testados por Barros [13].

$$h_0(n) = \{-0.125, 0.25, 0.75, 0.25, -0.125\}$$

$$h_1(n) = \{-0.25, 0.5, -0.25\}$$

$$g_0(n) = \{0.25, 0.5, 0.25\}$$

$$g_1(n) = \{-0.125, -0.25, 0.75, -0.25, -0.125\}$$

Na decimação da saída de filtros passa-baixas, mantém-se as linhas ou colunas pares, descartando-se as ímpares. Na decimação da saída de filtros passa-altas, mantém-se as linhas ou colunas ímpares,

descartando-se as pares. Na interpolação na entrada filtros passa-altas, acrescentam-se linhas ou colunas pares contendo zeros, mantendo-se as ímpares.

O resultado final pode ser visto na figura 2. A imagem \$A_N\$ (\$N=3\$) corresponde às baixas frequências, \$D_{j1}\$ às frequências altas na vertical (bordas horizontais), \$D_{j2}\$ às frequências altas na horizontal (bordas verticais) e \$D_{j3}\$ às frequências altas tanto na vertical como na horizontal (cantos ou bordas diagonais), onde \$N\$ indica o número de oitavas e \$j\$ é o contador de oitavas.



Figura 2 – Disposição das sub-bandas Wavelet de uma imagem.

No sistema proposto, a Transformada Wavelet é aplicada sobre uma imagem digital com 256 tons de cinza de 256x256 pixels, em três escalas, tendo como resultado 10 sub-bandas wavelets que serão processadas individualmente. A Transformada Wavelet é uma ferramenta importante para diminuir o efeito de blocos causado pela quantização vetorial utilizando redes neurais. Outra característica é a concentração de energia em algumas sub-bandas, onde os pixels têm uma maior correlação, podendo assim diminuir o número de bits/pixel para algumas sub-bandas que possuem uma menor correlação. A Transformada Cosseno Discreta é aplicada nas sub-bandas em blocos do mesmo tamanho dos blocos da quantização vetorial diminuindo a diversidade dos blocos, e assim melhorando a qualidade da imagem final. A DCT é aplicada da seguinte forma na sub-bandas:

- Nas sub-bandas \$A_3\$ e \$D_{13}\$ não é aplicada a DCT;
- Nas sub-bandas \$D_{31}, D_{32}, D_{33}\$ a DCT é aplicada em blocos de 2x2 pixel;

- Nas sub-bandas D21, D22, D23, D11, D12 a DCT é aplicada em blocos de 4x4 pixel;

Os blocos da DCT são pequenos por motivo de tempo de processamento, pois o tempo da transformada cresce exponencialmente em relação ao tamanho do bloco. A Quantização Vetorial utilizando Rede Neural faz um mapeamento dos blocos trocando-os pelo identificador da codeword que melhor representa cada bloco, isso causa perda no processo de descompressão. As codewords pertencem a um codebook, e cada sub-banda tem o seu próprio codebook, menos a sub-banda A3, por ter uma alta correlação entre os pixels e não poder apresentar muita perda, e a sub-banda D13 que é descartada por não causar muita perda na descompressão por ter uma correlação baixa entre os pixels. O codebook é gerado através de um treinamento com os coeficientes da DCT aplicado a cada sub-banda da Wavelet. Para o treinamento da rede, foi utilizado um conjunto de 13 imagens digitais com 256 tons de cinza, de 256x256 pixels, que serão utilizadas como vetor de entrada da rede neural.

O vetor de entrada da rede será constituído de blocos de dimensões variáveis como pode ser visto na figura 3.

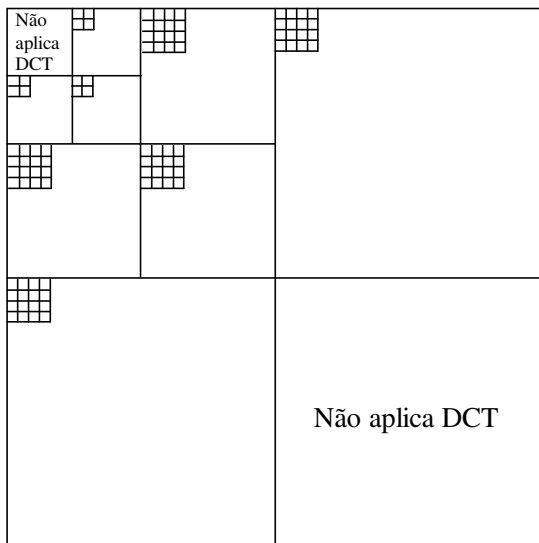


Figura 3 – Esquema dos blocos da DCT

A tabela 1 mostra o esquema de quantização.

9. Resultados

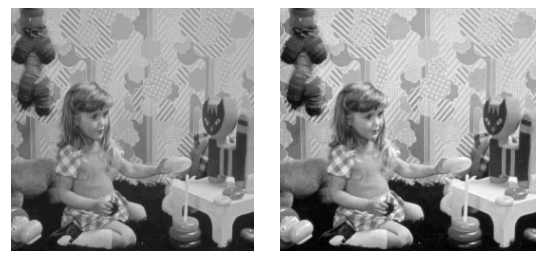
Os resultados obtidos podem ser considerados satisfatórios, uma vez que outros métodos tradicionais alcançam resultados semelhantes.

Tabela 1 – Alocação de bits dos codebooks para cada sub-banda.

Sub-banda	Nº de Bits	Nº de Codewords
A3	8	0
D11	9	512
D12	9	512
D13	9	512
D21	8	256
D22	8	256
D23	8	256
D31	8	256
D32	8	256
D33	0	0

A partir desses resultados, novos métodos podem ser propostos, principalmente no tocante ao uso das características de detalhes das sub-bandas da transformada wavelet, inclusive a variação da formatação das codewords da quantização vetorial levando-se em consideração a correlação dos coeficientes da wavelet codificados nas formas verticais e horizontais.

A seguir, apresenta-se os resultados obtidos com algumas imagens de teste:



(a) Girl.256 original (b) Girl.256 recuperada

Wavelet + DCT + QV	Bits/pixel
Huffman	3,9854
LZ	0,5073
RLE	1,7492

PSNR de Girl recuperada = 37,27dB

Figura 4 – Imagens Girl.256 original e recuperada.

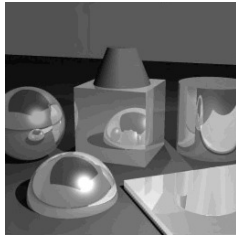


(a) Lady.256 original (b) Lady.256 recuperada

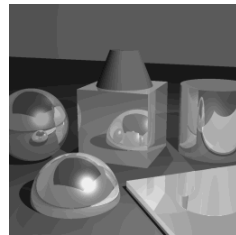
Wavelet + DCT + QV	Bits/pixel
Huffman	1,8907
LZ	0,5520
RLE	1,7453

PSNR de Lady.256 recuperada = 42,42 dB

Figura 5 – Imagens Lady.256 original e recuperada



(a) Solids.256 original



(b) Solids.256 recuperada

Wavelet + DCT + QV	Bits/pixel
Huffman	8,2662
LZ	0,4841
RLE	1,1555

PSNR de Solids.256 recuperada = 44,11dB

Figura 6 – Imagens Solids.256 original e recuperada

10. Conclusões

O algoritmo apresentado não se propõe a oferecer a melhor solução para o problema da compressão de imagem usando Wavelet e Redes Neurais. A sua contribuição está no fato de verificar a possibilidade de se unir alguns métodos de compressão com perdas e sem perdas, buscando usufruir das vantagens particulares de cada um. A partir desse algoritmo, pode-se então, buscar métodos de refinamento para se melhorar a qualidade e as taxas de compressão alcançadas. Inclusive, pode-se explorar mais a fundo o uso do SOM e a aplicação de outros métodos de preservação de bordas [14].

O algoritmo proposto apresenta uma complexidade computacional elevada para a qualidade dos resultados obtidos, mas considerando-se que há muitas técnicas que podem ser agregadas a este método, acredita-se na contribuição deste algoritmo como uma direção para se chegar a algum método mais eficiente.

11- Referências Bibliográficas

- [1] Egger, O., Fleury, P., Ebrahimi, T., Kunt, M. "High-Performance Compression of Visual Information – A Tutorial Review – Part I: Still Pictures". *Proceedings off the IEEE*, v.87, n.6, p.976-1011, Jun.1999.
- [2] Miano, John. , *Compressed Image File Formats – JPEG, PNG, GIF, XBM, BMP*. Addison Wesley Longman Inc, USA, 1999.
- [3] Baxes, Gregory A. , *Digital image processing: principles and applications*. John Wiley & Sons, Inc., 1994.
- [4] Nelson, Mark. *The Data Compression Book: Featuring fast, efficient data compression techniques in C*. M&T Books, 1991.
- [5] Linde, Y., Buzzo, A., Gray, R. M., "An Algorithm for Vector Quantizer Design", *IEEE Transaction on Communication*, v. Com-28, n. 1, pp. 84-95, jan. 1980.
- [6] DeSieno, D., "Adding a conscience to competitive learning", *In IEEE International Conference on Neural Networks*, pp. 117-124. IEEE Press, 1988.
- [7] Marr, D., *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information.*, W. H. Freeman and Company, USA, 1982.
- [8] Adelson, E. H., Anderson, C. H., Bergen, J.R., *et al.*, "Pyramid Methods in Image Processing.", *RCA Engineer*, v.29, pp. 33-41, nov./dec 1984.
- [9] Simoncelli, E. P., *Orthogonal Subband Image Transforms*. M.Sc. dissertation, Massachusetts Institute of Technology, may 1988.
- [10] Burt, P. J., Aelson, E. H., "The Laplacian Pyramid as a Compact Image Code", *IEEE Trans. On Commun.*, v.31, pp. 532-540, abril 1983.
- [11] Woods, J. W., O' NEIL, S. D., "Subband Coding of Images", *IEEE Trans. On Acoustics, Speech and Signal Processing*, v. 34, pp. 1278-1288, oct. 1986.
- [12] Mallat, S. G., "A Theory for Multiresolution Signal Decomposition: The Wavelet Representation", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 11, n. 7, pp. 674-693, july 1989.
- [13] Barros, S. R. R., 1998, *Compressão de Imagem Usando Quantização Vetorial Nas Subbandas da Transformada Wavelet*. Dissertação de MSc., COPPE/UFRJ, Rio de Janeiro, RJ, Brasil.
- [14] Ye, Xujun; Li, Zhineng; "Edge-preserving vector quantization using a neural network". *Electronic Imaging and Multimedia Systems*, Proc. SPIE Vol. 2898, p. 210-216, september, 1996.