Projeções Ortogonais sobre Conjuntos Convexos para Recuperação de Imagens Comprimidas pelo JPEG

Michel Eduardo Beleza Yamagishi
Departamento de Matemática Aplicada

UNICAMP - 1997

Projeções Ortogonais sobre Conjuntos Convexos para Recuperação de Imagens Comprimidas pelo JPEG

Este exemplar corresponde a redação final da tese devidamente corrigida e defendida pelo Sr. *Michel Eduardo Beleza Yamagishi* e aprovada pela Comissão Julgadora.

Campinas, 13 de março de 1997

Prof. Dr. Alvaro Rodolfo De Pierro

Dissertação apresentada ao Instituto de Matemática, Estatística e Computação Científica, UNICAMP, como requisito parcial para obtenção do Título de MESTRE em Matemática Aplicada.

FICHA CATALOGRÁFICA ELABORADA PELA BIBLIOTECA DO IMECC DA UNICAMP

Yamagishi, Michel Eduardo Beleza

Y14p

Projeções ortogonais sobre conjuntos convexos para recuperação de imagens comprimidas pelo Jpeg / Michel Eduardo Beleza Yamagishi – Campinas, [S.P.:s.n.], 1997.

Orientador: Alvaro Rodolfo De Pierro

Dissertação (mestrado) - Universidade Estadual de Campinas, Instituto de Matemática, Estatística e Computação Científica.

1. Compressão de imagens. 2. Algoritmo. 3. Conjuntos convexos. I. De Pierro, Alvaro Rodolfo. II. Universidade Estadual de Campinas. Instituto de Matemática, Estatística e Computação Científica. III. Título

Dissertação de Mestrado defendida e aprovada em 26 de fevereiro de 1997 pela Banca Examinadora composta pelos Profs. Drs.

Prof (a). Dr (a). JOSÉ MÁRIO MARTINEZ PEREZ

Prof (a). Dr (a). LÚCIO TUNES DOS SANTOS

Prof (a). Dr (a). ALVARO RODOLFO DE PIERRO

Deus criou o Homem à sua imagem,
à imagem de Deus o criou,
macho e fêmea Ele os criou.
(Gn 1,27)

But life is short and information endless...
Abbreviation is a necessary evil and the
abbreviator's business is to make the best of a
job which, although intrinsically bad, is still
better than nothing.
(Aldous Huxley)

Aos meus pais
Marlene e Teruaki.

Agradecimentos

Em primeiro lugar gostaria de agradecer a Deus que, segundo minha fé, muito me ajudou na realização desta dissertação ;

Pelo apoio e incentivo, gostaria de agradecer aos meus pais, Marlene e Teruaki, à minha noiva Mônica,à paróquia Santa Izabel na pessoa do Pe. Paulo Crozera, aos professores e funcionários do Departamento de Matemática Aplicada, aos funcionários da secretaria de pós-graduação e a todos os amigos (são muitos para mencionar);

Pela orientação e amizade, agradeço ao Prof. Álvaro Rodolfo De Pierro;

Pela ajuda neste trabalho, agradeço aos Professores José Mário Martínez, Lúcio Tunes dos Santos e Nir Cohen;

Pela ajuda com o LaTeX, agradeço à Professora Margarida Pinheiro Mello;

Pela ajuda em obter as imagens digitais, agradeço ao amigo Carlo Giuliano e à colega Lucila do departamento de Estatística;

E agradeço, em especial, ao CNPq pelo apoio financeiro recebido.

Conteúdo

Capítulo 1

Introdução

A utilização de imagens digitais em quase todas as aplicações computacionais nos trouxe novos desafios. Quem nunca teve a experiência de tentar armazenar em disco uma imagem no formato BMP ou GIF e ficou assustado com o tamanho em Kb do arquivo? Quem ao "navegar" pela Internet não percebeu o quanto é lenta a transmissão de arquivos gráficos? Às vezes, o computador carece de memória, e o usuário de paciência...

Não é mais possível prescindir das imagens digitais. Elas são um meio eficaz e mais simples de transmitir informações . Citando o senso comum: "uma imagem fala mais que mil palavras." Sem o auxílio das imagens digitais da Tomografia Computadorizada seria difícil um médico dar um diagnóstico confiável sobre determinadas doenças; sem as imagens tridimensionais do Mathematica, muitos estudantes e até pesquisadores matemáticos experientes ficariam embaraçados ao tentar visualizar a função $f(x,y) = sen(x^2+y^2)$ (veja figura a seguir); sem as interfaces "amigáveis", baseadas em computação gráfica, muitos usuários precisariam ler manuais bíblicos para manipular determinados programas; enfim, as imagens digitais vieram para ficar.

Vieram, mas não sozinhas. Elas trouxeram muitos problemas consigo. Um dos principais é a questão do armazenamento e transmissão das mesmas. Surge do esforço de contornar tais dificuldades a idéia de comprimi-las. Este sendo um tópico importante do processamento digital de sinais, chamado compressão de imagens. Usando um codificador, comprimem-se as imagens para eventual armazenamento ou transmissão; depois, quando forem necessárias, basta um decodificador para recuperá-las.

Se você já tentou viajar e levar todas as suas roupas numa única maleta sem a ajuda da mãe, percebeu que ao invés de levar oito calças Jeans é melhor levar apenas duas; ao invés de dez camisas, apenas 3, e assim por diante. É preciso reduzir a redundância de roupas. O preço é que, na festa de aniversário de um amigo, você não poderá usar aquela calça jeans desbotada, pois teve que deixá-la; e nem poderá ir com aquele sapato

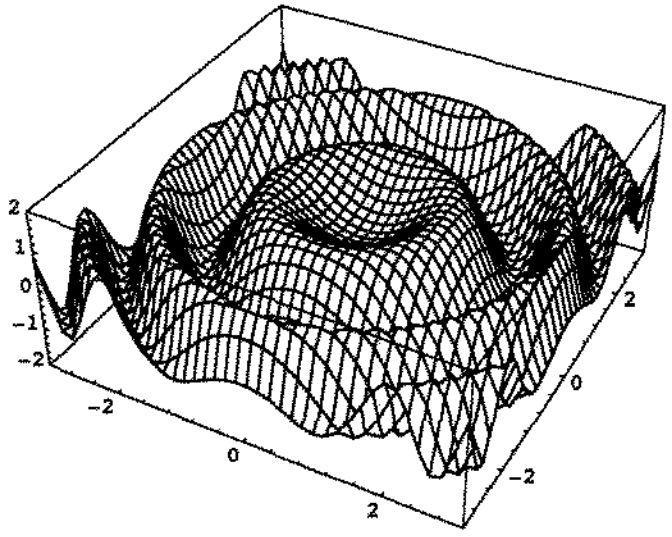


Figura 1.1: Gráfico da função $f(x,y) = sen(x^2 + y^2)$

x, porque só trouxe aquele que estava em seu pé. Só alguém muito observador notará que você está com a mesma roupa do jantar anterior. É claro que você possui roupas suficientes, mas não todas as suas roupas. Em compressão de imagens algo semelhante ocorre. Descartamos informações redundantes sem afetar a qualidade visual da imagem, porém ela pode não ser recuperada totalmente depois de comprimida, pois alguns dados foram jogados fora.

A perda de dados pode não afetar a qualidade da imagem; contudo, nos casos de alta taxa de compressão, a imagem começa a apresentar ruídos indesejados. Quando o método de compressão é o JPEG (veja [7] e [8]), o artefato mais comum é a descontinuidade entre blocos adjacentes. Surge a necessidade de métodos de recuperação de imagens "muito" comprimidas. É nesta categoria que se insere a técnica discutida nesta dissertação . Trata-se de um algoritmo iterativo de projeções ortogonais sobre conjuntos convexos fechados para recuperar imagens comprimidas com altas taxas de compressão pelo JPEG.

Nosso trabalho consistiu em primeiramente estudar os formatos em que as imagens são armazenadas e processadas pelos computadores (imagens digitais), compreender o método de compressão proposto pelo *JPEG* e realizar vários experimentos utilizando um pacote compatível com o *JPEG*. Em seguida, implementamos dois algoritmos propostos nos artigos [1] e [2]. Pesquisamos o ferramental matemático que garantia convergência dos mesmos e efetuamos testes com imagens padrões, comparando os resultados obtidos. Neste estágio do trabalho, várias idéias foram surgindo e fizemos algumas modificações

nos algoritmos para melhor compreendê-los e perceber a necessidade ou não de algumas convenções e hipóteses. Finalmente, compilamos os resultados nestas páginas.

Tentamos dividir os capítulos com a finalidade de introduzir os pré-requisitos fundamentais para compreensão dos métodos. Assim, definimos *imagem digital* e sua representação matricial; as *transformadas de imagens* têm um capítulo especial culminando com a *transformada cosseno* e suas propriedades; a *quantização* e o padrão internacional de compressão *JPEG* cada um possui um capítulo, entretanto são tratados do ponto de vista da aplicação , pois, são assuntos muito extensos e, no contexto desta dissertação , não precisaram ser aprofundados; já a *teoria de projeções sobre conjuntos convexos* foi tratada com mais detalhes, enunciamos os principais teoremas e suas respectivas demonstrações com o intuito de mostrar a ferramenta matemática que está por detrás do algoritmo; o algoritmo é apresentado em duas versões, pois a primeira é mais intuitiva e certamente ajuda na compreensão da segunda que por sua vez é mais delicada matematicamente falando; finalmente os experimentos e conclusões.

Não é fácil a tarefa de sistematizar e resumir um tema tão vasto e rico. Corre-se o risco de ser superficial demais em alguns aspectos e detalhista de mais em outros, prejudicando o conjunto do trabalho. Tentamos sublinhar a matemática do algoritmo, deixando em segundo plano os detalhes computacionais do trabalho – que, na verdade, são bem desafiadores e exigiram muitas horas de trabalho–a maior parte delas. Foi uma escolha pessoal, baseada na nossa formação .

As partes do texto menos elaboradas trazem referências bibliográficas para um eventual aprofundamento. Para aqueles que desejarem uma versão do JPEG mais recente, deixamos no Capítulo 8 o endereço na internet para obtê-lo via *ftp*. Na bibliografia há material suficiente para a compreensão do tópico aqui tratado.

Capítulo 2

Imagem Digital

A imagem digital está presente na grande parte dos procedimentos computacionais, seja como resultado visual de um determinado problema ou seja como dado a ser processado. Cada vez mais as imagens digitais são utilizadas como interface entre usuário e computador, proporcionando o chamado relacionamento "amigável" homem-máquina. Por isso, é fundamental compreendermos o conceito de imagem digital e a forma como esta é representada em nossos modelos matemáticos e no computador. Uma excelente referência sobre o tema é o livro *Computação Gráfica: Imagem*,[12].

2.1 Modelo de Imagem

Há na literatura vários modelos matemáticos para descrever uma imagem, sendo entre eles o modelo espacial o mais intuitivo e o mais utilizado, por isso nos fixaremos nele. Porém, não podemos deixar de mencionar o modelo do espaço de frequência[1], utilizado mais a frente neste trabalho.

2.1.1 Imagem Contínua

Ao abrirmos nossos olhos, recebemos de cada ponto do espaço um impulso luminoso que associa informação de cor a cada ponto. Portanto, um modelo matemático natural para descrever uma imagem é o de uma função definida em uma superfície bidimensional e tomando valores em um espaço de cor[2].

[1] Este é associado ao modelo espacial através de uma transformação matemática.

[2] Veja os capítulos 3 e 4 do livro [12] na bibliografia para maiores detalhes sobre fundamentos de cor.

Definição 1 *Uma imagem contínua é uma aplicação $i : U \to C$, onde $U \subset \Re^3$ é uma superfície e C é um espaço vetorial*[3].

Geralmente, U é subconjunto do plano, e C é um espaço de cor. A função i é chamada de *função imagem*, o conjunto U é chamado de *suporte da imagem* e o conjunto imagem (subconjunto de C) é conhecido como *valores da imagem* ou *gamute de cores da imagem*.

Se C é um espaço de dimensão 1, dizemos que a imagem é monocromática. Nesse caso particular, a imagem pode ser vista geometricamente como o gráfico $G(i)$ da função imagem i,

$$G(i) = \{(x, y, z); (x, y) \in U \text{ e } z = i(x, y)\}, \tag{2.1}$$

considerando os valores de intensidade como a altura $z = i(x, y)$ em cada ponto (x, y) do domínio.

2.1.2 Representação de uma imagem

Primeiramente, tomamos um subconjunto discreto $U' \subset U$ do domínio da imagem, depois um espaço de cor e finalmente a imagem é representada pela amostragem da função imagem i no conjunto U'. Essa é a representação mais utilizada em computação gráfica.

A imagem $i(x, y)$ será espacialmente contínua ou discreta se as coordenadas (x, y) variarem no conjunto U ou U', respectivamente. E no caso discreto, cada ponto (x_i, y_i) do subconjunto U' é denominado de *pixel*[4].

Uma vez discretizado o domínio da imagem, resta-nos, para podermos codificar a imagem no computador, trabalhar com um subconjunto discreto do espaço de cor C. Tal discretização é chamada de *quantização* .

Representação Matricial

Uma maneira muito usual de discretizar o domínio da imagem é tomar o conjunto U como sendo o retângulo

$$U = [a, b] \times [c, d] = \{(x, y) \in \Re^2;\ a \leq x \leq b \text{ e } c \leq y \leq d\}, \tag{2.2}$$

e discretizar esse retângulo usando os pontos de um retângulo bidimensional $\Delta = (\Delta_x, \Delta_y)$,

$$\Delta = \{(x_j, y_k) \in U;\ x_j = j.\Delta_x,\ y_k = k.\Delta_y,\ \ j, k \text{ inteiros},\ \ \Delta_x, \Delta_y \in \Re\}. \tag{2.3}$$

conforme a figura abaixo:

[3] O adjetivo "contínuo" nesta definição é utilizado com o sentido de não-discreto; e não com o seu significado usual em topologia de que a aplicação i é contínua.

[4] Do inglês Picture Element.

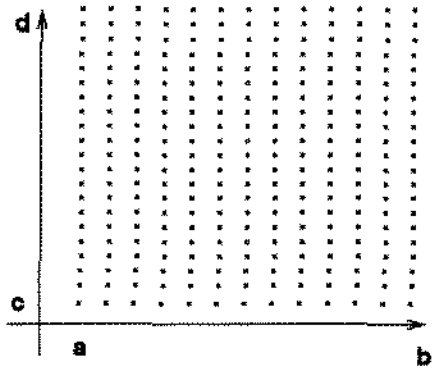


Figura 2.1: Discretização do domínio

Assim, cada pixel (x_j, y_k) da imagem pode portanto ser representado por coordenadas inteiras (j, k). Logo, a imagem pode ser modelada de forma conveniente no formato matricial. A imagem dessa forma é associada a uma matriz A de ordem $m \times n$, $A = (a_{jk}) = (i(x_j, y_k))$. Cada elemento a_{jk} da matriz A representa o valor da função imagem i no ponto de coordenadas (x_j, y_k) do reticulado, sendo pois um vetor do espaço de cor, representando a cor do pixel de coordenadas (j, k). Se a imagem for monocromática, então cada elemento a_{jk} da matriz A é um número real que representa a luminância do pixel.

Chamamos de resolução vertical da imagem ao número de linhas m da matriz A e de resolução horizontal da imagem ao número de colunas n da matriz A. Ao produto $m \times n$ damos o nome de resolução espacial ou resolução geométrica. Esta está relacionada com a frequência de amostragem final da imagem: quanto maior a frequência tanto maior será a quantidade de detalhes da imagem, isto é, apresentará elementos de alta frequência no espaço de frequências. Entretanto, a resolução da imagem dada em termos absolutos não fornece muita informação sobre a resolução real da imagem quando realizada em um dispositivo físico. Porque ficamos na dependência do tamanho físico do pixel do dispositivo[5]. Uma medida mais confiável da resolução espacial é a chamada *densidade de resolução* que fornece o número de pixels por unidade linear de medida. Geralmente, utilizamos o número de pixels por polegada, *ppi*[6], também conhecido como *dpi*[7].

2.1.3 Imagem Digital

Pelo que foi exposto acima, podemos pensar nos seguintes modelos de imagem: *contínuo–contínuo*, *contínuo–quantizada*, *discreta–contínuo* e *discreta–quantizada*. O primeiro mod-

[5] Por exemplo, o monitor é um dispositivo físico.

[6] Pixels Per Inch.

[7] Dots Per Inch.

elo sendo muito utilizado no desenvolvimento dos métodos matemáticos para o processamento de imagens. Em contra partida, o último modelo é o mais utilizado nos dispositivos gráficos e por isso, do ponto de vista prático, é o mais importante. E uma imagem representada por esse modelo é chamada de *imagem digital.*

Uma imagem digital possui basicamente duas componentes: as coordenadas do pixel e a informação de cor de cada pixel. Estes dois elementos estão diretamente relacionados com a resolução espacial e resolução de cor da imagem. O número de componentes do pixel é a dimensão do espaço de cor utilizado. Por exemplo, cada pixel de uma imagem monocromática tem uma única componente.

Em outras palavras, uma imagem digital monocromática pode ser representada por uma matriz real, possibilitando um processamento computacional e numérico mais familiar para a maioria dos programadores de computadores e matemáticos.

Entendido o conceito de imagem digital, podemos partir diretamente para as mais diversas aplicações em processamento digital de imagens. Em particular, o aprimoramento da qualidade de uma imagem com artefatos entre blocos devido a uma alta taxa de compressão, que é o tema desta dissertação .

Capítulo 3

Transformadas de Imagens

Há problemas matemáticos difíceis de serem tratados tal como são propostos. Um método muito utilizado para solucioná-los é a *Transformação Matemática*. Ou seja, transforma-se um problema difícil em outro de mais fácil abordagem. Uma vez resolvido este, efetua-se a transformação inversa para obter-se a solução do problema original.

Em processamento digital de imagens, as transformadas têm um papel fundamental, em particular, quando se trata de compressão de imagens, pois é a primeira medida a ser tomada. Nos parágrafos seguintes, apresentaremos a definição geral de transformadas de imagens e as principais propriedades que as tornam tão importantes do ponto de vista da compressão. Nosso objetivo é caminhar em direção à transformada cosseno para justificar o fato dela ser, atualmente, a preferida em compressão de imagens.

3.1 Definições e Propriedades

Utilizando a representação matricial de uma imagem, esta pode ser vista como um elemento do espaço vetorial das matrizes na base canônica[1]. Após a transformação , a imagem pode ser entendida como escrita em outra base ortonormal conveniente. A conveniência da transformação será precisada posteriormente, quando tratarmos das propriedades. Vamos à definição .

Consideremos uma imagem $u(x,y) \in \Re^{N\times N}$[2]. Definiremos o par de transformadas:

$$v(k,l) \triangleq \sum_{m=0}^{N-1}\sum_{n=0}^{N-1} u(m,n)a_{k,l}(m,n) \tag{3.1}$$

[1]As definições e propriedades das matrizes aqui tratadas podem ser encontradas em [21].

[2]A imagem pode ser retângular, ou seja, $u(x,y) \in \Re^{N\times M}$. A generalização é imediata.

$$u(m,n) \triangleq \sum_{k=0}^{N-1}\sum_{l=0}^{N-1} v(k,l)a^*_{k,l}(m,n), \tag{3.2}$$

onde $\{a_{k,l}\}$, chamada de Transformada de Imagens, é um conjunto ortogonal completo, ou seja, a nova base; $a^*_{k,l}$ denota o conjugado de $a_{k,l}$, e em geral $a_{k,l} \in C$.

Os elementos $v(k,l)$ são os coeficientes transformados, e $V = \{v(k,l)\}$ é a imagem transformada. A propriedade de ortonormalidade assegura que qualquer soma truncada da forma

$$u_{P,Q}(m,n) \triangleq \sum_{k=0}^{P-1}\sum_{l=0}^{Q-1} v(k,l)a^*_{k,l}(m,n) \ , \ P \leq N \ , \ Q \leq N \tag{3.3}$$

minimizará a soma dos erros quadráticos, ou seja,

$$\sigma_e^2 \triangleq \sum_{m=0}^{N-1}\sum_{n=0}^{N-1} [u(m,n) - u_{P,Q}(m,n)]^2 \tag{3.4}$$

Além disso, a propriedade de completude assegura que o erro é *zero* quando $P = Q = N$.

3.2 Transformações Unitárias Separáveis

O número de multiplicações e adições para obter-se $v(k,l)$, usando a fórmula (3.1) é $\mathcal{O}(N^4)$ que é proibitivo para imagens de utilizações práticas[3]. Podemos reduzir para $\mathcal{O}(N^3)$ operações se exigirmos que a transformada seja separável, isto é,

$$a_{k,l}(m,n) = a_k(m)b_l(n) = a(k,m)b(l,n), \tag{3.5}$$

onde $\{a_k(m), \quad k = 0,...,N-1\}$, $\{b_l(n), \quad l = 0...N-1\}$ são conjuntos de vetores completos e ortonormais em $\Re^N$. Assim, $A = \{a(k,m)\}$ e $B = \{b(l,n)\}$ são matrizes unitárias, e portanto,

$$AA^{*^t} = A^tA^* = I \ , \ BB^{*^t} = B^tB^* = I. \tag{3.6}$$

Por razões práticas, escolhemos $B = A$. Com tal escolha, podemos reescrever as fórmulas (3.1) e (3.2) da seguinte maneira:

$$v(k,l) = \sum_{m=0}^{N-1}\sum_{n=0}^{N-1} a(k,m)u(m,n)a(l,n) \ \leftrightarrow \ V = AUA^t, \tag{3.7}$$

[3]Para uma imagem 512×512, teríamos a ordem de 6.87×10^{10} operações .

$$u(m,n) = \sum_{m=0}^{N-1}\sum_{n=0}^{N-1} a^*(k,m)v(k,l)a^*(l,n) \;\leftrightarrow\; U = A^{*^t}VA^*. \tag{3.8}$$

Para imagens retangulares pertencentes a $\Re^{M\times N}$, teremos:

$$V = A_M U A_N^t, \tag{3.9}$$

$$U = A_M^{*^t} V A_N^*. \tag{3.10}$$

onde A_M e A_N são matrizes unitárias de $M \times M$ e de $N \times N$[4], respectivamente. Essas são as chamadas Transformações bidimensionais separáveis. Notemos que (3.7) pode ser escrita como:

$$V^t = A[AU]^t, \tag{3.11}$$

ísto é, (3.7) é obtida primeiramente transformando cada coluna de U e em seguida transformando as linhas da primeira operação para obter-se as linhas de V.

3.3 Base de Imagens

Vamos explicitar a base de imagens. Cada transformação terá a sua própria base com as respectivas características. Os elementos da base podem ser assim obtidos.

Denotaremos por a_k^* a k-ésima coluna de A^{*^t}. Vamos definir as matrizes

$$A_{k,l}^* = a_k^* a_l^{*^t} \tag{3.12}$$

e o produto interno das matrizes F e G, ambas de $N \times N$, como

$$< F, G >= \sum_{m=0}^{N-1}\sum_{n=0}^{N-1} f(m,n)g^*(m,n) \tag{3.13}$$

Com as definições acima, é possível escrever (3.10) como

$$U = \sum_{m=0}^{N-1}\sum_{n=0}^{N-1} v(k,l)A_{k,l}^*, \tag{3.14}$$

onde $v(k,l) =< U, A_{k,l}^* >$.

A fórmula (3.14) expressa qualquer imagem U como combinação linear de N^2 matrizes $A_{k,l}^*$ $k,l = 0,...,N-1$ que são os elementos da base de imagens. Em outras palavras, U é a projeção ortogonal de V sobre o espaço gerado pela base de imagens.

[4]A partir deste momento utilizaremos a expressão A de $M \times N$ para dizer que $A \in \Re^{M\times N}$.

Se U e V são escritos como vetores (ordenados por linhas ou colunas) u e v, respectivamente, então as equações (3.9) e (3.10) podem ser expressas como:

$$v = (A \otimes A)u = \mathcal{A}u \tag{3.15}$$

$$u = (A \otimes A)^{*^t} v = \mathcal{A}^{*^t} v, \tag{3.16}$$

onde $\mathcal{A} = A \otimes A$ é o produto de *Kronecker* [5].

3.4 Propriedades das Transformadas Unitárias

Como vimos até agora, exigimos que as transformações fossem unitárias. Não sem motivo, pois tais transformações têm propriedades muito importantes para a manipulação de imagens. Algumas propriedades são comuns a todas as transformações unitárias; outras, apenas a um número restrito de transformações unitárias. São essas diferenças que tornam umas mais indicadas que outras para determinados problemas. Vejamos primeiro as propriedades comuns a todas; e quando formos falar da transformada cosseno indicaremos as características que a distinguem das demais e tornam-na melhor na compressão de imagens.

3.4.1 Conservação de Energia

Sempre que utilizarmos a norma $\| \cdot \|$ estaremos nos referindo à norma euclidiana, ou seja, $\| v \| = (\sum_{i=0}^{N-1} \mid v(k) \mid^2)^{1/2}$.

Nas transformações unitárias

$$v = Au, \tag{3.18}$$

temos que

$$\| v \|^2 = \| u \|^2. \tag{3.19}$$

Assim, utilizando a definição de norma,

[5] Definição : Se A e B são matrizes de $M_1 \times M_2$ e $N_1 \times N_2$, respectivamente, então o produto de Kronecker de A e B é definido como:

$$A \otimes B = \begin{bmatrix} a_{11}B & \dots & a_{1M_2}B \\ \vdots & & \vdots \\ a_{M_1 1}B & \dots & a_{M_1 M_2}B \end{bmatrix}, \tag{3.17}$$

onde a_{ij} é o elemento da matriz A na i-ésima linha e j-ésima coluna.

$$\| v \|^2 = \sum_{k=0}^{N-1} | v(k) |^2 = v^{*^t} v = u^{*^t} A^{*^t} A u = u^{*^t} u = \sum_{n=0}^{N-1} | u(n) |^2 = \| u \|^2, \qquad (3.20)$$

obtemos que as transformações unitárias preservam energia ou, equivalentemente, o módulo do vetor u no espaço N-dimensional. Analogamente, para transformações unitárias bidimensionais como em (3.1) e (3.2), podemos provar que

$$\sum_{m=0}^{N-1} \sum_{n=0}^{N-1} | u(m,n) |^2 = \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} | v(k,l) |^2. \qquad (3.21)$$

3.4.2 Compactação de Energia e Variância dos Coeficientes Transformados

Compactação de energia é uma das principais propriedades para a aplicação que faremos das transformadas de imagens.

Tal característica consiste em que algumas das transformações unitárias tendem a concentrar grande parte da energia de uma imagem em relativamente poucos coeficientes transformados. Como a energia total é conservada, isto implica que muitos coeficientes transformados possuirão pouca energia (próximo a zero).

Se μ_u e R_u denotarem a média e a covariância de um vetor u, então teremos que

$$\mu_v = E[v] = E[Au] = AE[u] = A\mu_u \qquad (3.22)$$

$$R_v = E[(v - \mu_v)(v - \mu_v)^{*^t}] = A(E[(u - \mu_u)(u - \mu_u)^{*^t}])A^{*^t} = AR_u A^{*^t}. \qquad (3.23)$$

As variâncias dos coeficientes transformados serão dadas pelos elementos da diagonal de R_v, isto é,

$$\sigma_v^2(k) = [R_v]_{k,k} = [AR_u A^{*^t}]_{k,k} \qquad (3.24)$$

Como A é uma matriz unitária, segue-se que

$$\sum_{k=0}^{N-1} | \mu_v(k) |^2 = \mu_v^{*^t} \mu_v = \mu_u^{*^t} A^{*^t} A \mu_u = \sum_{n=0}^{N-1} | \mu_u(n) |^2 \qquad (3.25)$$

O que implica em

$$\sum_{k=0}^{N-1} E[| v(k)) |^2] = \sum_{n=0}^{N-1} E[| u(n) |^2] \qquad (3.26)$$

A energia média $E[\mid v(k) \mid^2]$ dos coeficientes transformados $v(k)$ tende a ser distribuida de forma não uniforme, embora possa ser uniformemente distribuida para $u(n)$[6].

3.4.3 Descorrelação

Quando os elementos do vetor u são altamente correlacionados, os elementos do vetor transformado tendem a ser descorrelacionados. Isto implica que os elementos fora da diagonal principal da matriz de covariância R_v são proporcionalmente pequenos em relação aos elementos da diagonal. Em linguagem matemática, as transformações realmente interessantes são aquelas que " diagonalizam " a matriz de covariância.

3.5 Transformada de Karhunen-Loeve e a Transformada Cosseno

É muito difícil justificar a importância da Transformada Cosseno no contexto de compressão de imagens sem antes discutir a Transformada de Karhunen-Loeve (TKL). A Transformada Cosseno aproxima esta última num sentido que deixaremos claro nos próximos parágrafos.

Com o intuito de ilustrar a finalidade da Transformada de Karhunen-Loeve, consideremos o problema de transmitir via Internet um sinal senoidal. Este poderia ser discretizado e em seguida transmitido. Intuitivamente, quanto mais fina for a discretização tanto melhor será a qualidade do sinal reconstruido. Entretanto, uma forma mais eficiente de transmiti-lo é fornecer a amplitude, a fase, a frequência, o tempo inicial e a informação que o sinal é uma senoide.Ou seja, apenas 5 (cinco !) informações são suficientes para reconstruir o sinal perfeitamente ! Note a compressão presente nesse exemplo.

Do ponto de vista da Teoria de Informação , os valores discretizados do sinal são altamente correlacionados e pouca informação é transmitida por cada valor discretizado; por outro lado, a amplitude, a fase, a frequência, o tempo inicial e a informação da forma do sinal são totalmente descorrelacionados e possuem a mesma quantidade de informação dos valores discretizados.

Vamos definir a TKL (veja [6]). Para um vetor aleatório $u \in \Re^N$, a base da TKL é dada pelos autovetores ortonormalizados da matriz de autocorrelação R[7] do vetor u, isto é,

$$R\phi_k = \lambda_k \phi_k. \quad 0 \leq k \leq N-1 \tag{3.27}$$

[6] Veja [6].

[7] $R = E[uu^t]$.

A TKL de u é definida como

$$v = \Phi^{*^t} u, \tag{3.28}$$

onde $\Phi^{*^t} = \Phi^{*^t}$. A transformada inversa é

$$u = \Phi v = \sum_{k=0}^{N-1} v(k)\phi_k, \tag{3.29}$$

onde ϕ_k é a k-ésima coluna de Φ.

Pela própria definição , obtemos que a TKL diagonaliza a matriz de autocorrelação do vetor aleatório u, isto é,

$$\Phi^{*^t} R \Phi = \Lambda = Diag\{\lambda_k\}. \tag{3.30}$$

É usual trabalhar com a matriz de covariância ao invés da matriz de autocorelação . Se $\mu = E[u]$, então

$$R_0 = cov[u] = E[(u-\mu)(u-\mu)^t] = E[uu^t] - uu^t = R - \mu\mu^t. \tag{3.31}$$

Assim, se μ é conhecida, então a matriz com os autovetores normalizados de R_0 determina a TKL do vetor aleatório de média zero $u - \mu$. Em geral, a TKL de u e $u - \mu$ não necessariamente são idênticas.

3.5.1 A TKL de uma Imagem

Se uma imagem $u(m,n)$, de $N \times N$, é representada por um campo aleatório cuja função de autocorrelação é dada por

$$E[u(m,n)u(m',n')] = r(m,n;m',n'), \;\; 0 \leq m,m',n,n' \leq N-1 \tag{3.32}$$

então a base da TKL é formada pelas autofunções ortonormalizadas $\psi_{k,l}(m,n)$, obtidas resolvendo-se

$$\sum_{m'=0}^{N-1}\sum_{n'=0}^{N-1} r(m,n;m',n')\psi_{k,l}(m',n') = \lambda_{k,l}\psi_{k,l}(m,n), \tag{3.33}$$

onde $0 \leq k,l,m,n \leq N-1$.

Em notação matricial podemos reescrever (3.33)

$$\mathcal{R}\psi_i = \lambda_i \psi_i, \quad i = 0, ..., N^2 - 1 \tag{3.34}$$

onde $\psi_i \in \Re^{N^2}$ é a representação vetorial de $\psi_{k,l}$ e $\mathcal{R} \in \Re^{N^2 \times N^2}$ é a matriz de autocorrelação da imagem em sua forma vetorial $u \in \Re^{N^2}$. Então

$$\mathcal{R} = E[uu^t]. \tag{3.35}$$

Se $\mathcal{R}$ é separável, então a matriz $\Psi \in \Re^{N^2 \times N^2}$, cujas colunas são os $\{\psi_i\}$, torna-se separável. Por exemplo, seja

$$r(m,n;m',n') = r_1(m,m')r_2(n,n'). \tag{3.36}$$

Em notação matricial, isto significa que

$$\mathcal{R} = R_1 \otimes R_2, \quad \Psi = \Phi_1 \otimes \Phi_2, \tag{3.37}$$

onde

$$\Phi_j R_j \Phi^{*^t} = \Lambda_j, \quad j = 1,2 \tag{3.38}$$

e a TKL de u é

$$v = \Psi^{*^t} u = [\Phi^{*^t} \otimes \Phi^{*^t}]u. \tag{3.39}$$

Ou seja, a TKL é

$$V = \Phi_1^{*^t} U \Phi_2^*, \tag{3.40}$$

e a inversa

$$U = \Phi_1 V \Phi_2^t. \tag{3.41}$$

Exemplo

Consideremos a função de covariância separável de uma campo aleatório de média zero modelado por um processo de Marko de primeira ordem estacionário

$$r(m,n;m',n') = \rho^{|m-m'|}\rho^{|n-n'|}, \tag{3.42}$$

isto fornece $\mathcal{R} = R \otimes R$, onde R é a matriz abaixo

$$R = \begin{bmatrix} 1 & \rho & \rho^2 & \dots & \rho^{N-1} \\ \rho & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \rho^2 \\ \vdots & \ddots & \ddots & \ddots & \rho \\ \rho^{N-1} & \dots & \dots & \rho & 1 \end{bmatrix}. \tag{3.43}$$

Assim, $\mathcal{R} = R \otimes R$. Os autovalores de R são dados por

$$\lambda_k = \frac{(1-\rho^2)}{1-2\rho Cos(\omega_k)\rho^2}, \tag{3.44}$$

e os autovetores

$$\phi_k(m) = \phi(m,k) = \left(\frac{2}{N+\lambda_k}\right)^{1/2} Sen\left(\omega_k\left(m+1-\frac{N+1}{2}\right)+\frac{(k+1)\pi}{2}\right), \quad 0 \leq m,k \leq N-1 \tag{3.45}$$

onde os ω_k são as raízes positivas da equação

$$Tan(N\omega) = -\frac{(1-\rho^2)Sen(w)}{Cos(w)-2\rho+\rho^2 Cos(w)}, \tag{3.46}$$

para N par. Tem-se um resultado semelhante no caso em que N é ímpar.

Portanto, podemos obter a matriz Φ formada pelos autovetores de R. Logo, $\Psi = \Phi \otimes \Phi$ e a TKL será $\Phi^t \otimes \Phi^t$.

3.5.2 Propriedades da TKL

A TKL possui muitas propriedades importantes para o processamento de sinais. Destacaremos algumas mais importantes para a compressão de imagens. Por simplicidade, u será um vetor aleatório de média zero e sua matriz de covariância R será definida positiva.

Descorrelação

Os coeficientes transformados pela TKL $\{v(k), k = 0, ..., N-1\}$ são descorrelacionados e têm média zero, ou seja,

$$E[v(k)] = 0, \tag{3.47}$$

e

$$E[vv^{*^t}] = \Phi^{*^t} E[uu^t]\Phi = \Phi^{*^t} R\Phi = \Lambda = \text{diagonal}. \tag{3.48}$$

Vale resaltar que a matriz da TKL Φ não é a única matriz com essa propriedade. Na realidade, há muitas outras matrizes (unitárias ou não) que descorrelacionam os coeficientes transformados $v(k)$. Veja a seguinte matriz

$$L = \begin{bmatrix} 1 & 0 & \cdots & \cdots & 0 \\ -\rho & \ddots & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & -\rho & 1 \end{bmatrix}. \tag{3.49}$$

Fazendo o produto $L^t RL$ temos a matriz diagonal abaixo

$$D = \begin{bmatrix} 1-\rho^2 & 0 & \cdots & \cdots & 0 \\ 0 & 1-\rho^2 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 1-\rho^2 & 0 \\ 0 & \cdots & 0 & 0 & 1 \end{bmatrix}. \tag{3.50}$$

Portanto, a transformação $v = L^t u$, resultará em uma sequência $v(k)$ descorrelacionada. Observe que $L \neq \Phi$. Surge a seguinte pergunta: não seria melhor utilizar a matriz traingular inferior L ao invés da matriz Φ, uma vez, que esta última é de dificil obtenção ?

Para responder esta pergunta, vamos a próxima propriedade da TKL.

Minimização do Erro Médio Quadrático

Primeiramente, transformemos o vetor u no vetor v pela operação $v = Au$ (veja gráfico a seguir). Em seguida, formemos o vetor w tomando-se apenas os m primeiros elementos do vetor v, ou seja, $w = I_m v$, onde I_m é um operador que preserva os m primeiros elementos de v e transforma os demais elementos em zeros. Finalmente, w é transformado em z, isto é, $z = Bw$. As matrizes A e B pertencem a $\Re^{N\times N}$ e a matriz I_m é diagonal com 1 nos m primeiros elementos e zero nos demais. Consequentemente,

$$w(k) = \begin{cases} v(k), & 0 \leq k \leq m-1 \\ 0, & k \geq m \end{cases}. \tag{3.51}$$

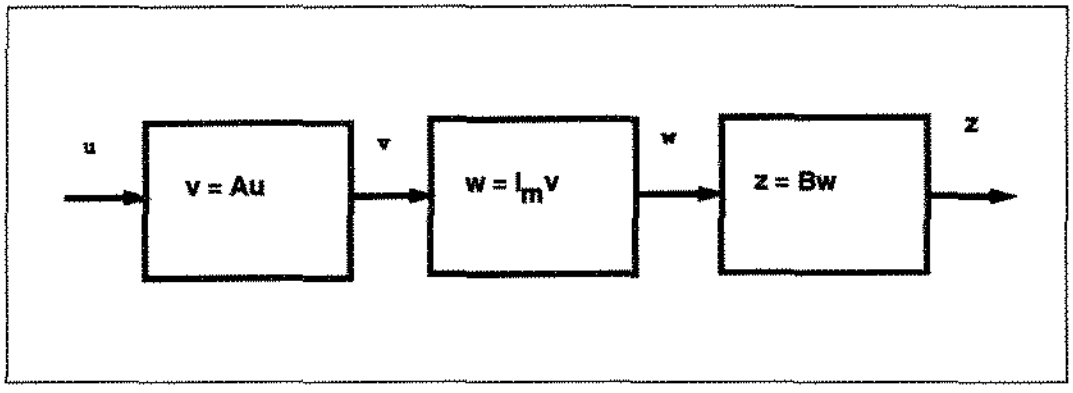


Figura 3.1: Transformações do vetor u até o vetor z

O erro médio quadrático entre as sequências $u(n)$ e $z(n)$ é definida como

$$J_m = \frac{1}{2} E\left(\sum_{n=0}^{N-1} \mid u(n) - z(n) \mid^2 \right) = \frac{1}{N} Tr[E\{(u-z)(u-z)^{*^t}\}], \tag{3.52}$$

onde $Tr(A)$ é o traço[8] da matriz A. Desejamos encontrar matrizes A e B tal que J_m seja minimizado para cada escolha de $m \in [1, N]$. Este mínimo é atingido pela TKL de u. Veja o teorema abaixo.

Teorema 1 *O erro J_m definido em (3.52) é mínimo quando*

$$A = \Phi^{*^t}, \quad B = \Phi, \quad AB = I, \tag{3.53}$$

onde as colunas de Φ são dispostas de acordo com a ordem decrescente dos autovalores de R.

Demonstração : Apesar de longa, a prova do teorema é bastante simples. Vamos fixar a notação .

$$v = Au, \quad w = I_m v, \text{ e } z = Bw. \tag{3.54}$$

Podemos portanto, escrever J_m da seguinte forma

$$J_m = \frac{1}{N} Tr\left[(I - BI_m A) R (I - BI_m A)^{*^t} \right]. \tag{3.55}$$

Para minimizar J_m, vamos, primeiramente, derivá-lo em relação aos elementos da matriz A, e igualar o resultado a zero. Isto fornece

$$I_m B^t (I - BI_m A)^* R = 0, \tag{3.56}$$

que por sua vez, simplifica J_m

$$J_m = \frac{1}{N} Tr\left[(I - BI_m A) R \right], \tag{3.57}$$

e fornece a igualdade

$$I_m B^{*^t} = I_m B^{*^t} B I_m A. \tag{3.58}$$

Quando $m = N$, o valor mínimo de J_m deve ser zero o que exige

$$I - BA = 0 \text{ ou } B = A^{-1}. \tag{3.59}$$

Usando (3.59) em (3.58) e rearranjando os termos,

$$I_m B^{*^t} B = I_m B^{*^t} B I_m. \quad 1 \leq m \leq N \tag{3.60}$$

[8] $Tr(A) = \sum_{i=0}^{N-1} a_{ii}$.

Para que a equação acima seja verdadeira para todo m, é necessário que $B^{*^t}B$ seja uma matriz diagonal. Como $B = A^{-1}$, é fácil ver que (3.58) permanece inauterada se a matriz B é substituida por DB ou BD, onde D é uma matriz diagonal. Portanto, sem perda de generalidade, podemos normalizar B de tal forma que $B^{*^t}B = I$, isto é, B é uma matriz unitária. Consequentemente, A também é uma matriz unitária e $B = A^{*^t}$. Isto fornece

$$J_m = \frac{1}{N}Tr((I - A^{*^t}I_mA)R) = \frac{1}{N}Tr(R - I_mARA^{*^t}). \tag{3.61}$$

Como R é fixa, J_m é minimizado se

$$\tilde{J}_m = Tr(I_mARA^{*^t}) = \sum_{k=0}^{m-1} a_k^t R a_k^* \tag{3.62}$$

é maximizado, onde a_k^t é a k-ésima linha de A. Como A é unitária,

$$a_k^t a_k^* = 1. \tag{3.63}$$

Para maximizar $\tilde{J}_m$ sujeito às restrições 3.63), tomamos a função Lagrangeana

$$\tilde{J}_m = \sum_{k=0}^{m-1} a_k^t R a_k^* + \sum_{k=0}^{m-1} \lambda_k (1 - a_k^t a_k^*), \tag{3.64}$$

e derivamos em relação a a_j. O resultado será a condição necessária de primeira ordem

$$Ra_j^* = \lambda_j a_j^*, \tag{3.65}$$

onde a_l^* são os autovalores ortonormalizados de R.

Assim,

$$\tilde{J}_m = \sum_{k=0}^{m-1} \lambda_k, \tag{3.66}$$

que é maximizado se $\{a_j^*,\ 0 \leq j \leq m-1\}$ corresponder aos m maiores autovalores de R. Como $\tilde{J}_m$ deve ser minimizado para todo m, é necessário que $\lambda_0 \geq \lambda_1 \geq \cdots \geq \lambda_{N-1}$. Então, a_k^t, as linhas de A, são os autovetores conjugados transpostos da matriz R, isto é, A é a TKL de u $\square$.

Concentração de Energia

Dentre todas as transformações unitárias $v = Au$, a TKL Φ^{*^t} é que concentra a maior energia média em $m \leq N$ coeficientes transformados de v. Definamos

$$\sigma_k^2 = E[\mid v(k) \mid^2], \quad \sigma_0 \geq \sigma_1 \geq \cdots \geq \sigma_{N-1} \tag{3.67}$$

e

$$S_m(A) = \sum_{k=0}^{m-1} \sigma_k^2. \tag{3.68}$$

Com tais definições , podemos provar o teorema a seguir.

Teorema 2 *Para qualquer $m \in [1, N]$ fixo, temos*

$$S_m(\Phi^{*^t}) \geq S_m(A), \tag{3.69}$$

onde A é qualquer outra transformação unitária.

Demonstração : Notemos que

$$S_m(A) = \sum_{k=0}^{m-1} (ARA^{*^t})_{k,k} = Tr(I_m A^{*^t} RA) = \tilde{J}_m, \tag{3.70}$$

que pelo Teorema 1 é maximizado quando A é a TKL. □

Feita esta pequena introdução da TKL, podemos definir a matriz da Transformada Cosseno, apresentar suas principais propriedades e relacioná-la com a TKL.

A matriz de $N \times N$ da transformada cosseno $C = \{c(k,n)\}$ é definida como:

$$c(k,n) = \begin{cases} \frac{1}{\sqrt{N}} & \text{se } k = 0 \ , 0 \leq n \leq N-1 \\ \sqrt{\frac{2}{N}} \cos[\frac{\pi(2n+1)k}{2N}] & \text{se } 1 \leq k \leq N-1 \ , 0 \leq n \leq N-1 \end{cases}$$

3.5.3 Propriedades da Transformada Cosseno

1. A transformada cosseno é real e ortogonal

$$C = C^* \Rightarrow C^{-1} = C^t;$$

2. a transformada cosseno não é a parte real da *Transformada de Fourier Discreta;*

3. a transformada cosseno é uma Transformada Rápida. Ou seja, a transformada cosseno de um vetor de N elementos pode ser calculada em $\mathcal{O}(N \log_2 N)$ operações ;

4. a transformada cosseno possui uma excelente taxa de compactação para dados altamente correlacionados.

5. as linhas da matriz da transformada cosseno, C, são os autovetores da matriz tridiagonal simétrica Q_c abaixo:

$$Q_c = \begin{bmatrix} 1-\alpha & -\alpha & 0 & \dots & 0 \\ -\alpha & 1 & -\alpha & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & -\alpha & 1 & -\alpha \\ 0 & \dots & 0 & -\alpha & 1-\alpha \end{bmatrix} \tag{3.71}$$

As últimas duas propriedades são fundamentais para a utilização da transformada cosseno na compressão de imagens.

Já mencionamos as principais propriedades da TKL que a tornam teoricamente ótima para compressão de imagens. Entretanto, do ponto de vista computacional, a TKL não é recomendada, pois depende dos dados, isto é, para cada imagem teríamos que computar uma nova matriz para a TKL. E isso é muito caro computacionalmente falando. Portanto, a transformada que seja próxima da TKL no sentido de descorrelacionar os coeficientes transformados, minimizar o erro médio quadrático para coeficientes truncados e seja computacionalmente mais barata será a indicada para utilização em compressão. Essa transformada é a transformada cosseno quando se trata de dados que podem ser modelados por um processo de Markov de primeira ordem estacionário[9]. Lembremos que para um processo de Markov $u(n)$ de primeira ordem estacionário a função de covariância é, para algum ρ,$r(n) = \rho^{|n|}$, $\mid \rho \mid < 1$, $\forall n$[10].

Para um vetor $u = \{u(n)\ ,\ n = 1, ..., N\}$ sua matriz de covariância será:

$$R = \begin{bmatrix} 1 & \rho & \rho^2 & \dots & \rho^{N-1} \\ \rho & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \rho^2 \\ \vdots & \ddots & \ddots & \ddots & \rho \\ \rho^{N-1} & \dots & \dots & \rho & 1 \end{bmatrix} \tag{3.72}$$

A relação de proximidade entre a transformada cosseno e a TKL se torna clara quando verificamos que R^{-1} é uma matriz simétrica e tridiagonal[11] e que quando tomamos os escalares $\beta^2 = \dfrac{(1-\rho^2)}{(1+\rho^2)}$ e $\alpha = \dfrac{\rho}{(1+\rho^2)}$ teremos:

[9] Um vetor aleatório $u(n)$ é modelado por um processo de Markov de ordem p se $Prob[u(n)\backslash u(n-1), u(n-2), ...] = Prob[u(n)\backslash u(n-1), ..., u(n-p)], \forall n$. Isto é, a probabilidade condicional de $u(n)$ dado todo o passado, só depende da probabilidade condicional dados somente $u(n-1), ..., u(n-p)$.

[10] Confira em [6].

[11] Veja [6] e [13].

$$\beta^2 R^{-1} = \begin{bmatrix} 1-\rho\alpha & -\alpha & 0 & \dots & 0 \\ -\alpha & 1 & -\alpha & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & -\alpha & 1 & -\alpha \\ 0 & \dots & 0 & -\alpha & 1-\rho\alpha \end{bmatrix} \tag{3.73}$$

Ou seja, $\beta^2 R^{-1} \cong Q_c$ quando $\rho \cong 1$. Isto implica que os autovetores de R e os autovetores de Q_c estarão muito próximos, isto é, C aproxima a TKL.

Talvez o seguinte comentário seja oportuno. O fato de modelar nossas imagens por processos de Markov de primeira ordem estacionário justifica-se, pois se tomarmos um pixel qualquer da imagem, provavelmente, ele pertencerá a uma mesma região, por exemplo o cabelo da Lena, dos pixels imediatamente vizinhos, e portanto terá mais ou menos o mesmo tom de cinza. Ou seja, pixels vizinhos estão altamente correlacionados ($\rho \cong 1$).

Capítulo 4

Quantização

É através da quantização que descartamos os dados redundantes ou pouco significativos da imagem e assim obtemos a compressão propriamente dita.

Resta-nos fazer uma pequena consideração sobre a distinção entre *dados* e *informação* . Utilizaremos para alcançar essa finalidade uma pequena analogia.

Quando contamos uma história a alguém, utilizamos palavras para transmiti-la. Dependendo do grau de detalhes que utilizarmos mais palavras necessitaremos para contar o mesmo fato. Por exemplo, suponhamos que o José queira dizer para o João o resultado do jogo de basquete feminino da seleção brasileira contra a seleção norte-americana nas olimpíadas de Atlanta (87×111). Uma maneira de contar seria:" os Estados Unidos venceram o jogo por 111 a 87 ". Ou alternativamente: " no primeiro tempo o placar foi 57 a 46 para os Estados Unidos. No segundo tempo, os Estados Unidos fizeram mais 54 pontos enquanto o Brasil apenas 41 ". Naturalmente, as duas formas indicam o placar do jogo, porém a última forma é mais longa,isto é, utiliza um maior número de palavras. Portanto, a primeira forma é preferível do ponto de vista da economia de palavras. Nesse exemplo, o placar do jogo é a informação e as palavras são os dados utilizados para transmiti-la.

A seguir, faremos o paralelo com a compressão de imagens. Onde a informação é a imagem em si; enquanto os dados são as cores utilizadas para formar a imagem. Mostraremos em que sentido há redundância de cores (ou detalhes) em uma imagem e como eliminar essa redundância a fim de reduzir a quantidade de dados.

4.1 Redundância Psico-visual em Imagens

A existência da redundância psico-visual se deve ao fato que a percepção humana de informação numa imagem não envolve uma análise *quantitativa* em cada *pixel* ou o valor da luminância da imagem. Geralmente, um observador procura distinguir características tais como fronteiras, ou regiões de texturas diferentes, ou seja, contrastes na imagem,

e mentalmente tenta agrupá-las em partes reconhecíveis. Depois, o cerébro relaciona esses grupos com o seu próprio conhecimento a priori para assim completar o processo de reconhecimento da imagem.

De fato, se tomarmos um pixel de uma imagem na sua forma matricial, provavelmente a sua cor será parecida com a dos pixels vizinhos, porque há uma grande chance de eles pertencerem a um mesmo objeto. Caso isso não ocorra, este pixel pode estar na fronteira de vários objetos na imagem, ou talvez ele seja parte do padrão de cor de uma textura. Em qualquer caso, a estrutura da informação implica sempre em alguma redundância dos dados. E é essa redundância que procuramos eliminar.

A eliminação da informação redundante só é possível porque a sua contribuição não é relevante para o processo de visão normal. Isto é, há uma perda de dados; entretanto, aos olhos comuns, é impossível perceber qualquer diferença na imagem. Ou seja, a perda é quantitativa e não qualitativa.[1] Tal corte na redundância psico-visual é chamado de *quantização* . Esta denominação é consistente com a utilização corriqueira da palavra, pois, na prática, consiste em uma função de um domínio contínuo em um contra-domínio com um número finito de valores (discreto). Portanto, o processo de quantização é irreversível[2], e a compressão é dita *lossy*.

4.2 Quantização de uma Imagem

Matematicamente, um quantizador é definido através de uma função que leva uma variável contínua u em uma variável discreta u', que possui um número finito de valores $\{r_1, ..., r_L\}$. Esta função é geralmente uma função escada, sendo a regra de quantização a seguinte: defina $\{t_k,\ k = 1, ..., L+1\}$ como o conjunto de nível crescente de transição com t_1 e t_{L+1} como valores mínimo e máximo de u, respectivamente. Se u pertence ao intervalo $[t_k, t_{k+1})$, então u é levado em r_k, o k-ésimo nível de reconstrução . Veja figura abaixo. E afim de ilustrar o que foi dito, passaremos a descrever um quantizador, que embora muito simples, é bastante utilizado no prátrica.

Quantizador de Lloyd-Max

Este quantizador minimiza o erro médio quadrático para um determinado número de níveis de quantização . Seja u uma variável aleatória real com função de densidade de probabilidade contínua $p_u(u)$. Desejamos encontrar os níveis de transição t_k e os níveis de reconstrução r_k para uma quantização de L níveis de tal maneira que o erro médio quadrático

[1] Veremos que se a quantização for muito exigente, a qualidade da imagem estará comprometida.

[2] Veremos, mais tarde, que esta é a razão pela qual surgem os ruídos na imagem comprimida

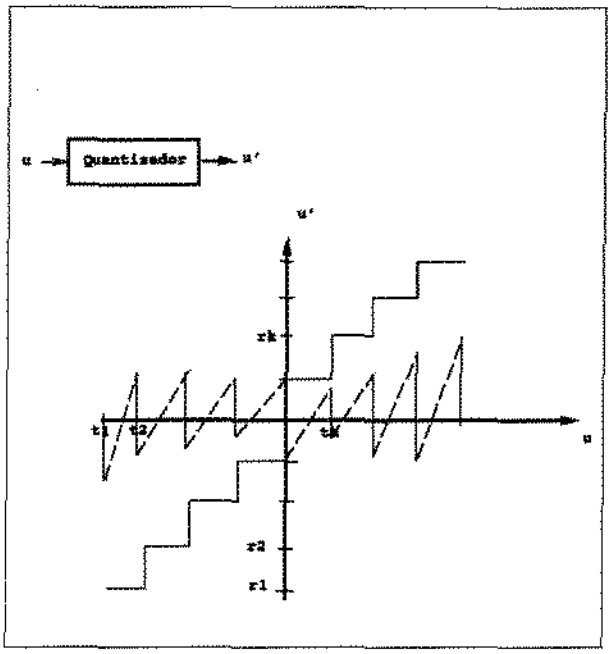


Figura 4.1: Exemplo de Quantizador

$$\varepsilon = E[(u-u')^2] = \int_{t_1}^{t_{L+1}} (u-u')^2 p_u(u)du \tag{4.1}$$

seja mínimo.

Reescrevendo-se,

$$\varepsilon = \sum_{i=1}^{L} \int_{t_i}^{t_{i+1}} (u-u')^2 p_u(u)du, \tag{4.2}$$

e para obter as condições necessárias de minimização de ε basta derivar com respeito a t_k e a r_k e igualar a zero. Ou seja,

$$\frac{\partial \varepsilon}{\partial t_k} = (t_k - r_{k-1})^2 p_u(t_k) - (t_k - r_k)^2 p_u(t_k) = 0 \tag{4.3}$$

$$\frac{\partial \varepsilon}{\partial r_k} = 2\int_{t_k}^{t_{k+1}} (u-r_k) p_u(u)du = 0, \quad 1 \leq k \leq L. \tag{4.4}$$

Usando o fato que $t_{k-1} \leq t_k$, podemos simplificar as equações acima, resultando em

$$t_k = \frac{(r_k + r_{k-1})}{2} \tag{4.5}$$

$$r_k = \frac{\int_{t_k}^{t_{k+1}} u p_u(u)du}{\int_{t_k}^{t_{k+1}} p_u(u)du} = E[u \backslash u \in \mathcal{I}_k], \tag{4.6}$$

onde, $\mathcal{I}_k$ é o k-ésimo intervalo $[t_k, t_{k+1})$.

As equações para t_k e r_k (4.5) são não-lineares e devem ser resolvidas simultaneamente tendo como dados t_1 e t_L. Para equacioná-las, utilizamos métodos numéricos como, por exemplo, o método de Newton.

Quantização Uniforme

Este é um dos quantizadores mais simples. Discutiremo-no afim de exemplicar os procedimentos gerais da quantização .

Se tomarmos uma distribuição uniforme, as equações do quantizador de Lloyd-Max se tornam lineares, e resultam em intervalos iguais entre os níveis de transição e os níveis de reconstrução .

Seja

$$p_u(u) = \begin{cases} \frac{1}{t_{L+1}-t_1}, & t_1 \leq u \leq t_{L+1} \\ 0, & \text{caso contrário} \end{cases} \tag{4.7}$$

Das equações (4.5), obtemos

$$r_k = \frac{(t_{k+1}^2 - t_k^2)}{2(t_{k+1} - t_k)} = \frac{t_{k+1} + t_k}{2} \tag{4.8}$$

$$t_k = \frac{t_{k+1} + t_{k-1}}{2}. \tag{4.9}$$

Assim, $t_k - t_{k-1} = t_{k+1} - t_k = \text{constante} \triangleq q$. Finalmente, obtemos

$$q = \frac{t_{L+1} - t_1}{L}, \tag{4.10}$$

$$t_k = t_{k-1} + q, \tag{4.11}$$

$$r_k = t_k + \frac{q}{2}. \tag{4.12}$$

Portanto, tanto os níveis de transição como os de reconstrução são igualmente espaçados.

Quantização dos Coeficientes Transformados

Na aplicação que faremos, a quantização será efetuada nos coeficientes transformados, ou seja, no espaço das frequências. Primeiramente, aplicamos a transformada cosseno na imagem, afim de descorrelacionar os dados e concentrar a energia da imagem nos coeficientes de baixa frequência.

Matematicamente, isto é equivalente a expandir a imagem em termos de uma base ortonormal. Se a imagem é quadrada, $N \times N$ e A é a sua representação matricial, então

$$A = \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} v(k,l) C_{k,l}^{*}, \tag{4.13}$$

onde $C_{k,l}^{*}$, $k, l = 0, ..., N-1$ é a base de imagens da transformada cosseno e $v(k,l)$ são os coeficientes transformados.

Chamemos de V a matriz formada pelos coeficientes transformados. A quantização é efetuada sobre os coeficientes de V. A vantagem desse procedimento é que geralmente os coeficientes de alta frequência têm módulo muito inferior aos coeficientes de baixa frequência. E aqueles estão relacionados a detalhes da imagem, podendo ser descartados na quantização sem afetar a qualidade da imagem. Mostraremos como esse processo é realizado no capítulo sobre o método Jpeg de compressão. Por enquanto, é importante notar que conceitualmente a quantização é como acima definida, entretanto a interpretação é ligeiramente diferente, pois a quantização espacial está relacionada com o espaço de cores e o corte na redundância da mesmas; enquanto a quantização dos coeficientes transformados está relacionada ao espaço das frequências e o corte de detalhes na imagem.

Para melhor compreensão, veja a seguinte imagem transformada

$$\begin{pmatrix} -415 & -29 & -62 & 25 & 55 & -20 & -1 & 3 \\ 7 & -21 & -62 & 9 & 11 & -7 & -6 & 6 \\ -46 & 8 & 77 & -25 & -30 & 10 & 7 & -5 \\ -50 & 13 & 35 & -15 & -9 & 6 & 0 & 3 \\ 11 & -8 & -13 & -2 & -1 & 1 & -4 & 1 \\ -10 & 1 & 3 & -3 & -1 & 0 & 2 & -1 \\ -4 & -1 & 2 & -1 & 2 & -3 & 1 & -2 \\ -1 & -1 & -1 & -2 & -1 & -1 & 0 & -1 \end{pmatrix}. \tag{4.14}$$

Notamos que os coeficientes de alta frequência têm valor em módulo menor que os de baixa frequência. Ou seja, tais elementos contribuem muito pouco para a formação da imagem, podendo ser desprezados (lembremos da representação em bases ortogonais).

Capítulo 5

Jpeg - Método Padrão de Compressão de Imagens

Dentro do espírito desta dissertação , este capítulo pretende ser um roteiro dos passos envolvidos no Jpeg sem aprofundá-los. Caso haja interesse numa discussão mais detalhada, recomendamos o paper de Gregory K. Wallace (veja [7]). Há, na Internet, implementações disponíveis do Jpeg, indicaremos um endereço no capítulo sobre os experimentos.

O Jpeg é o padrão internacional de compressão de imagens. Foi desenvolvido em conjunto pela ISO[1] e pelo CCITT[2] com a finalidade de facilitar a troca de imagens comprimidas entre o maior número de aplicativos. Pois, sem um padrão de compressão, havia um problema sério de compatibilidade entre os diversos pacotes de compressão no mercado. Cada aplicativo possuia sua própria rotina de compressão, e era praticamente impossível o intercâmbio de imagens entre softwares de fabricantes diferentes.

As duas seções anteriores, na verdade, eram pré-requisitos para compreender os dois principais procedimentos do Jpeg[3] que agora iremos apresentar. O Jpeg possui três sistemas diferentes de codificação :

1. um **Sistema Básico de Codificação** que permite perda de dados e é baseado na Transformada Cosseno Discreta (TCD), a qual é a mais indicado para a maioria das aplicações que envolvem compressão;

2. um **Sistema de Codificação Extendido** para alta taxa de compressão, alta precisão ou para aplicações que necessitam de reconstrução progressiva;

3. um **Sistema Independente de Codificação** que não permite perda de dados, possibilitando desta forma a recuperação exata da imagem.

[1] International Standardization Organization.
[2] Consultative Committee of International Telephone and Telegraph.
[3] Joint Photographics Experts Group.

Nenhum formato de arquivo é especificado, tampouco resolução espacial ou modelo de espaço de cor. Para um produto ser compatível com o padrão Jpeg deve possuir pelo menos o sistema básico de codificação .

No sistema básico, também conhecido como sistema sequêncial básico, os dados de entrada e saída são restringidos a uma precisão de 8 bits, enquanto os coeficientes transformados pela TCD são restringidos a 11 bits. A compressão é efetuada em três passos sequênciais:

- cálculo da Transformada Cosseno Discreta;
- quantização ;
- codificação simbólica.

Sem perda de generalidade, vamos supor que a imagem a ser comprimida seja monocromática. Pois, no caso policromático, podemos imaginar cada componente de cor sendo processada separadamente, recaindo num processo análogo ao monocromático para cada componente de cor.

No codificador, primeiramente, a imagem é dividida em blocos de dimensão 8×8 que são processados da esquerda para a direita, de cima para baixo. Cada bloco possui 64 elementos (elementos da matriz 8×8). Subtraímos de cada elemento o valor 2^{n-1}, onde 2^n é o número máximo de níveis de cinza da imagem. Após essa operação , realizamos a TDC do bloco seguida da quantização e finalmente os dados são reordenados em zig-zag como na figura a seguir

0	1	5	6	14	15	27	28
2	4	7	13	16	26	29	42
3	8	12	17	25	30	41	43
9	11	18	24	31	40	44	53
10	19	23	32	39	45	52	54
20	22	33	38	46	51	55	60
21	34	37	47	50	56	59	61
35	36	48	49	57	58	62	63

Essa forma de armazenar os dados é conveniente porque o vetor unidimensional assim obtido contém os coeficientes em ordem crescente de frequência. E o Jpeg aproveita o fato que os coeficientes de alta frequência, depois de quantizados, são na maioria nulos para depois, os chamados coeficientes AC[4], serem codificados eficientemente utilizando a *variable-length code*[5] que define o valor do coeficiente e o número de zeros que o precedem.

[4]São todos os coeficientes, exceto o primeiro que é conhecido como coeficiente DC.

[5]Maiores detalhes veja o Apêndice A, os livros [9], [16] ou o artigo [7].

Somente o primeiro coeficiente, isto é o coeficiente DC, é codificado diferentemente. Ele é codificado subtraindo-se dele o coeficiente DC do bloco anterior.

No decodificador, realizamos as operações em ordem reversa, tendo que fazer uma aproximação bruta para a "quantização inversa", pois esta é irreversível. Veja no gráfico abaixo os esquemas do codificador e do descodificador.

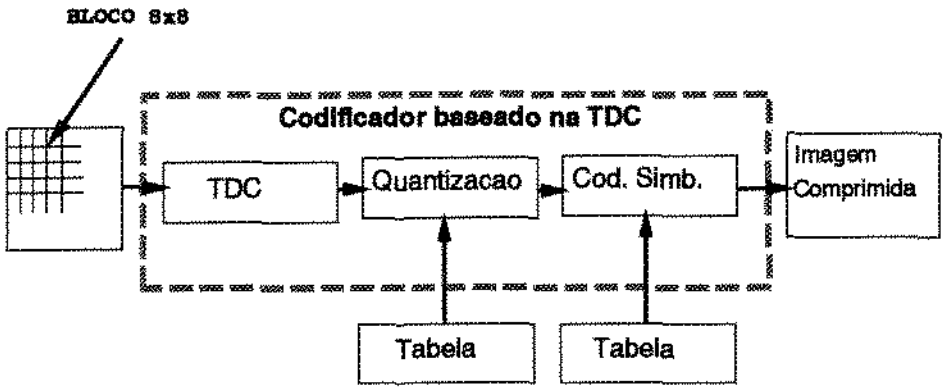


Figura 5.1: Codificador JPEG

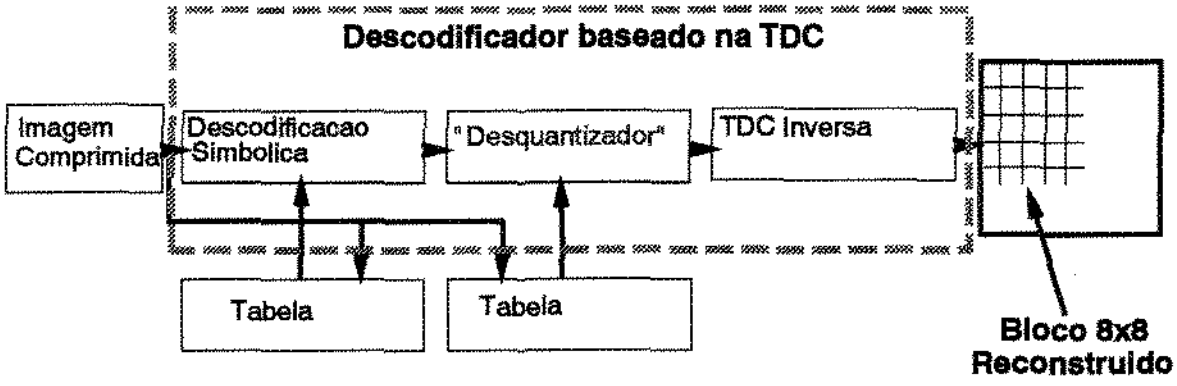


Figura 5.2: Descodificador JPEG

Para melhor compreensão, vamos dar um exemplo. Considere o seguinte bloco 8×8 como sendo parte de uma imagem com 256, isto é, 2^8 níveis de cinza.

$$\begin{pmatrix} 52 & 55 & 61 & 66 & 70 & 61 & 64 & 73 \\ 63 & 59 & 66 & 90 & 109 & 85 & 69 & 72 \\ 62 & 59 & 68 & 113 & 144 & 104 & 66 & 73 \\ 63 & 58 & 71 & 122 & 154 & 106 & 70 & 69 \\ 67 & 61 & 68 & 104 & 126 & 88 & 68 & 70 \\ 79 & 65 & 60 & 70 & 77 & 68 & 58 & 75 \\ 85 & 71 & 64 & 59 & 55 & 61 & 65 & 83 \\ 87 & 79 & 69 & 68 & 65 & 76 & 78 & 94 \end{pmatrix} \tag{5.1}$$

Devemos subtrair 2^7 de cada elemento da matriz, ou seja

$$\begin{pmatrix} -76 & -73 & -67 & -62 & -58 & -67 & -64 & -55 \\ -65 & -69 & -62 & -38 & -19 & -43 & -59 & -56 \\ -66 & -69 & -60 & -15 & 16 & -24 & -62 & -55 \\ -65 & -70 & -57 & -6 & 26 & -22 & -58 & -59 \\ -61 & -67 & -60 & -24 & -2 & -40 & -60 & -58 \\ -49 & -63 & -68 & -58 & -51 & -65 & -70 & -53 \\ -43 & -57 & -64 & -69 & -73 & -67 & -63 & -45 \\ -41 & -49 & -59 & -60 & -63 & -52 & -50 & -34 \end{pmatrix} \tag{5.2}$$

Fazendo-se a transformada cosseno, obtemos os coeficientes transformados. Veja matriz abaixo.

$$T = \begin{pmatrix} -415 & -29 & -62 & 25 & 55 & -20 & -1 & 3 \\ 7 & -21 & -62 & 9 & 11 & -7 & -6 & 6 \\ -46 & 8 & 77 & -25 & -30 & 10 & 7 & -5 \\ -50 & 13 & 35 & -15 & -9 & 6 & 0 & 3 \\ 11 & -8 & -13 & -2 & -1 & 1 & -4 & 1 \\ -10 & 1 & 3 & -3 & -1 & 0 & 2 & -1 \\ -4 & -1 & 2 & -1 & 2 & -3 & 1 & -2 \\ -1 & -1 & -1 & -2 & -1 & -1 & 0 & -1 \end{pmatrix}. \tag{5.3}$$

Algumas observações podem ser feitas neste ponto. Primeiro, os coeficientes de baixa frequência são em módulo maiores que os de alta frequência; segundo, o primeiro coeficiente é uma "média", a menos de uma constante multiplicativa, dos coeficientes do bloco.

Depois, fazemos a quantização dos coeficientes transformados. Para isso, utilizamos uma tabela de quantização que deve ser fornecida pelo o usuário. Por exemplo, a tabela

$$Q = \begin{pmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{pmatrix}. \tag{5.4}$$

A quantização é obtida realizando a operação

$$\hat{T}(i,j) = \text{Inteiro mais próximo} \left[\frac{T(i,j)}{Q(i,j)} \right], \tag{5.5}$$

onde $\hat{T}(i,j)$ é o coeficiente transformado e quantizado da i-ésima linha e j-ésima coluna,$T(i,j)$ e $Q(i,j)$ são os coeficientes transformados e coeficientes da tabela de quantização , respectivamente. Efetuando-se a quantização , obtemos

$$\hat{T} = \begin{pmatrix} -26 & -3 & -6 & 2 & 2 & 0 & 0 & 0 \\ 1 & -2 & -4 & 0 & 0 & 0 & 0 & 0 \\ -3 & 1 & 5 & -1 & -1 & 0 & 0 & 0 \\ -4 & 1 & 2 & -1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}. \tag{5.6}$$

Notamos que apenas 17 elementos são diferentes de zero. Agora, ordenando-se os elementos em zig-zag, obtemos o seguinte vetor

$$-26 \quad -3 \quad 1 \quad -3 \quad -2 \quad -6 \quad 2 \quad -4 \quad 1 \quad -4 \quad 1 \quad 1 \quad 5 \quad 0 \quad 2 \quad 0 \quad 0 - 1 \quad 2 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad -1 \quad -1$$

onde □ indica que é fim do bloco.

Segue-se a codificação simbólica *Huffman coding*[6] ou *arithmetic coding*.

No decodificador, após recriarmos a matriz $\hat{T}$, fazemos uma aproximação para a "quantização inversa" da seguinte forma:

$$\dot{T}(i,j) = \hat{T}(i,j)Q(i,j). \tag{5.7}$$

Resultando em

[6] Veja Apêndice A.

$$T = \begin{pmatrix} -416 & -33 & -60 & 32 & 48 & 0 & 0 & 0 \\ 12 & -24 & -56 & 0 & 0 & 0 & 0 & 0 \\ -42 & 13 & 80 & -24 & -40 & 0 & 0 & 0 \\ -56 & 17 & 44 & -29 & 0 & 0 & 0 & 0 \\ 18 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}. \tag{5.8}$$

Realizando-se a transformada cosseno inversa,

$$\begin{pmatrix} -70 & -64 & -61 & -64 & -69 & -66 & -58 & -50 \\ -72 & -73 & -61 & -39 & -30 & -40 & -54 & -59 \\ -68 & -78 & -58 & -9 & 13 & -12 & -48 & -64 \\ -59 & -77 & -57 & 0 & 22 & -13 & -51 & -60 \\ -54 & -75 & -64 & -23 & -13 & -44 & -63 & -56 \\ -52 & -71 & -72 & -54 & -54 & -71 & -71 & -54 \\ -45 & -59 & -70 & -68 & -67 & -67 & -61 & -50 \\ -35 & -47 & -61 & -66 & -60 & -48 & -44 & -44 \end{pmatrix} \tag{5.9}$$

E notamos que há diferenças entre os elementos da matriz reconstruída (5.9) e os elementos da matriz original (5.2). Entretanto, geralmente, isso é imperceptível na imagem. E esse é o caso ótimo, pois apenas as redundâncias foram descartadas. Contudo, se a quantização dos elementos de baixa frequência for alta, aparecerão na imagem os artefatos entre blocos[7].

[7]Veja capítulo 7.

Capítulo 6

Projeções sobre Conjuntos Convexos para Recuperação de Imagens

Em 1978, Dante C. Youla (veja [3]) foi um dos primeiros a perceber que muitos problemas de recuperação de imagens eram essencialmente de natureza geométrica, e sugeriu a seguinte formulação matemática: a imagem original f é um vetor que, a priori, sabemos pertencer a um subespaço linear $\mathcal{P}_b$ contido em um espaço de Hilbert,$\mathcal{H}$; entretanto tudo que temos é a sua imagem $P_a f$ que é a projeção de f sobre um subespaço linear $\mathcal{P}_a$ também contido em $\mathcal{H}$.

1. Encontre condições necessárias e suficientes para que f seja unicamente determinada a partir de $P_a f$;

2. encontre condições necessárias e suficientes para que a reconstrução linear de f a partir de $P_a f$ seja estável na presença de ruídos.

As respostas são as seguintes:

- f é unicamente determinada a partir de $P_a f$ se, e somente se, $\mathcal{P}_b$ e o complemento ortogonal de $\mathcal{P}_a$ só tiverem o vetor zero em comum;

- o problema de reconstrução é bem posto se, e somente se, o ângulo entre $\mathcal{P}_b$ e o complemento ortogonal de $\mathcal{P}_a$ é maior que zero. (Todos os ângulos estão no intervalo $[0, \pi/2]$);

- sem ruído, existe nos casos acima um algoritmo recursivo (veja gráfico) para a recuperação de f utilizando-se somente operações de projeções sobre $\mathcal{P}_b$ e sobre o complemento ortogonal de $\mathcal{P}_a$

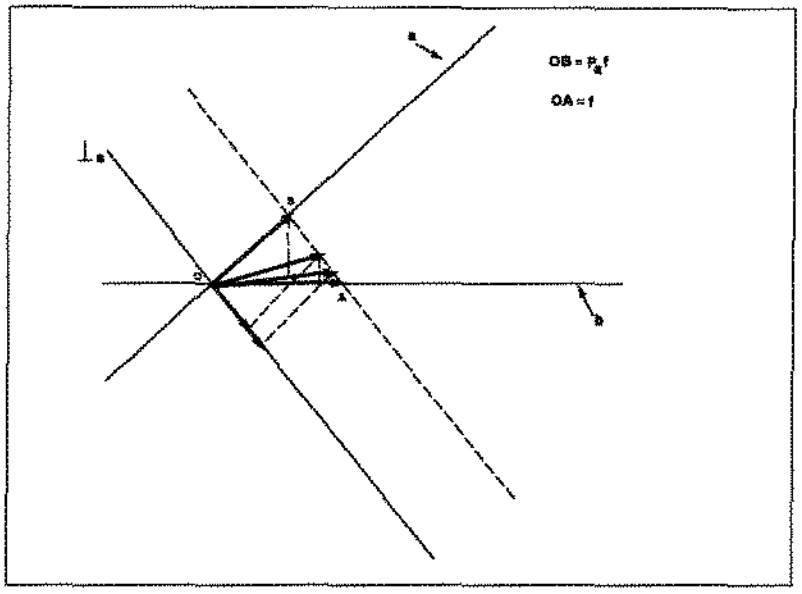


Figura 6.1: Algoritmo Iterativo: $f_{k+1} = P_a f + P_b P_{\perp a} f_k$.

Com o objetivo de apresentar os principais resultados da teoria matemática envolvida, seguiremos a sequência do livro editado por Stark (veja [5]). Antes porém, devemos definir dois tipos de convergência: a convergência fraca e a convergência forte.

Definição 2 *Convergência Fraca. Uma sequência $\{x_n\}$ em um espaço normado X converge fracamente se existir um elemento $x \in X$ tal que para todo elemento $f \in X'$*[1]*,*

$$\lim_{n\to\infty} f(x_n) = f(x).$$

Denotamos a convergência fraca por

$$x_n \rightharpoonup x.$$

O elemento x é chamado de limite fraco de $\{x_n\}$.

Definição 3 *Convergência Forte. Uma sequência $\{x_n\}$ em um espaço normado X converge fortemente se existir um elemento $x \in X$ tal que*

$$\lim_{n\to\infty} \| x_n - x \| = 0.$$

[1] X' é o espaço dual de X, isto é, o espaço formado por todos os funcionais lineares limitados definidos sobre X.

6.1 Conjuntos Convexos em Espaços de Hilbert

Os conjuntos convexos em espaços de Hilbert, por possuirem propriedades importantes, desempenham um papel relevante nos algoritmos para recuperação de imagens propostos nos artigos [1] e [2]. Apresentaremos a definição de conjuntos convexos, e demonstraremos as propriedades em espaços de Hilbert (veja [5]) que faremos uso no decorrer deste capítulo.

Definição 4 *Um subconjunto $\mathcal{C}$ de um espaço de Hilbert*[2] $\mathcal{H}$ *é dito convexo se dados x_1 e x_2 elementos contidos em $\mathcal{C}$ para todo λ, $0 \leq \lambda \leq 1$, o elemento $\lambda x_1 + (1-\lambda)x_2$ também está contido em $\mathcal{C}$.*

O teorema a seguir é importante porque (i) sugere a definição de *Operador de Projeção* , ou seja, para qualquer $x \in \mathcal{H}$ a projeção $P_{\mathcal{C}}x$ de x em $\mathcal{C}$ é o elemento de $\mathcal{C}$ mais próximo de x; (ii) garante que se $\mathcal{C}$ é convexo e fechado, então $P_{\mathcal{C}}x$ existe e é único e satisfaz ao critério de minimização :

$$\| x - P_{\mathcal{C}}x \| = \min_{y \in \mathcal{C}} \| x - y \| . \tag{6.1}$$

Teorema 3 *Seja C um subconjunto convexo fechado de um espaço de Hilbert $\mathcal{H}$. Seja f um elemento qualquer de $\mathcal{H}$. Então existe um único elemento $x_0 \in \mathcal{C}$ tal que:*

$$\inf_{x \in \mathcal{C}} \| f - x \| = \| f - x_0 \| . \tag{6.2}$$

Isto é x_0 é o elemento em $\mathcal{C}$ mais próximo de f no sentido da norma.

Demonstração : Para demonstrar a unicidade, suponhamos que

$$\inf_{x \in \mathcal{C}} \| f - x \| = \delta = \| f - x_0 \| = \| f - y_0 \|,$$

onde $x_0 \in \mathcal{C}$ e $y_0 \in \mathcal{C}$. Substituindo-se x e y na identidade

$$\| \frac{x+y}{2} \|^2 + \| \frac{x-y}{2} \|^2 = \frac{\| x \|^2 + \| y \|^2}{2} \tag{6.3}$$

por $f - x_0$ e $f - y_0$, respectivamente, obtemos

$$\| f - \frac{x_0 + y_0}{2} \|^2 = \delta^2 - \| \frac{x_0 - y_0}{2} \|^2 \leq \delta^2.$$

Mas, como $\mathcal{C}$ é convexo, $\frac{x_0+y_0}{2} \in \mathcal{C}$, portanto

$$\| f - \frac{x_0 + y_0}{2} \|^2 \geq \delta^2; \tag{6.4}$$

[2] Daqui em diante utilizaremos a letra $\mathcal{H}$ para denotar um espaço de Hilbert.

assim, $\| x_0 - y_0 \| = 0$ e $x_0 = y_0$.

Utilizando-se a definição de ínfimo, existe uma sequência $\{x_n\}$ de elementos contidos em $\mathcal{C}$ tal que

$$lim \| f - x_n \| = \delta. \tag{6.5}$$

Substituindo-se x e y na equação (6.3) por $f - x_n$ e $f - x_m$, respectivamente, obtemos:

$$\| x_n - x_m \|^2 = 2(\| f - x_n \|^2 + \| f - x_m \|^2) - 4 \| f - \frac{x_n + x_m}{2} \|^2 . \tag{6.6}$$

Logo, como $\frac{x_n + x_m}{2} \in \mathcal{C}$ para $n, m = 1 \to \infty$. As equações (6.4) e (6.6) implicam em

$$0 \leq \lim_{n,m \to \infty} \| x_n - x_m \|^2 \leq 4\delta^2 - 4\delta^2 = 0, \tag{6.7}$$

conclui-se que

$$\lim_{n,m \to \infty} \| x_n - x_m \| = 0. \tag{6.8}$$

A sequência $\{x_n\}$ é de Cauchy e converge para $x_0 \in \mathcal{C}$, pois $\mathcal{C}$ é fechado. Segue o resultado desejado pela equação (6.5):

$$\| f - x_0 \| = \delta. \quad \square \tag{6.9}$$

Um subespaço vetorial fechado é automaticamente[3] convexo e fechado. Essa observação será bastante útil nos colorários a seguir.

Corolário 1 *Seja $\mathcal{G}$ um subespaço vetorial fechado contido em $\mathcal{H}$. Então cada $x \in \mathcal{H}$ possui uma única decomposição na forma $x = x_1 + x_2$, onde $x_1 = P_{\mathcal{G}} x \in \mathcal{G}$ e $x_2 \in \perp\mathcal{G}$, que é o complemento ortogonal de $\mathcal{G}$.*

Demonstração : Vamos demonstrar, primeiramente, a unicidade da decomposição . Para isso, suponhamos por contradição que $x = x_1 + x_2 = x_3 + x_4$, onde x_1 e x_3 pertencem a $\mathcal{G}$, e x_2 e x_4 pertencem a $\perp\mathcal{G}$. Assim, $x_1 - x_3 = x_4 - x_2$ é um elemento da interseção de $\mathcal{G}$ e $\perp\mathcal{G}$. Mas, pela própria definição de complemento ortogonal, temos que $x_1 = x_3$ e $x_2 = x_4$. Agora, para provar a existência da decomposição , faremos a seguinte escolha: $x_1 = P_{\mathcal{G}} x$ e $x_2 = x - P_{\mathcal{G}} x$. É imadiato que $x = x_1 + x_2$. Falta apenas verificar que $x_2 \in \perp\mathcal{G}$. Isto é, $< x_2, y >= 0 \; \forall \, y \in \mathcal{G}$. Pela definição de $P_{\mathcal{G}}$ e utilizando a convexidade de $\mathcal{G}$, para qualquer λ, $0 < \lambda < 1$, $\lambda y + (1 - \lambda) P_{\mathcal{G}} x \in \mathcal{G}$, para todo $y \in \mathcal{G}$ tal que

$$\| x - P_{\mathcal{G}} x \|^2 \leq \| x - \lambda y - (1 - \lambda) P_{\mathcal{G}} x \|^2 . \tag{6.10}$$

Expandindo-se o lado direito da equação ,

[3] Veja definição de Conjuntos Convexos.

$$\| x - P_{\mathcal{G}}x \|^2 \leq \| x - P_{\mathcal{G}}x \|^2 + \lambda^2 \| y - P_{\mathcal{G}}x \|^2 - 2\lambda Re < x - P_{\mathcal{G}}x, y - P_{\mathcal{G}}x > . \quad (6.11)$$

Logo,

$$2Re < x - P_{\mathcal{G}}x, y - P_{\mathcal{G}}x > \leq \lambda \| y - P_{\mathcal{G}}x \|^2, \quad (6.12)$$

tomando-se o limite $\lambda \to 0$, encontramos a seguinte desigualdade

$$Re < x - P_{\mathcal{G}}x, y - P_{\mathcal{G}}x > \leq 0, \quad (6.13)$$

para todo $y \in \mathcal{G}$.

Como $\mathcal{G}$ é um subespaço vetorial, $\mu y \in \mathcal{G}$ para todo $\mu \in \Re$ e a equação (6.13), para μ real,

$$\mu Re < x - P_{\mathcal{G}}x, y > \leq Re < x - P_{\mathcal{G}}x, P_{\mathcal{G}}x > . \quad (6.14)$$

Essa equação só é verdadeira somente se

$$Re < x - P_{\mathcal{G}}x, y > = 0 \quad (6.15)$$

para todo $y \in \mathcal{G}$. Se, agora, substituirmos y em (6.15) por iy, obtemos

$$Im < x - P_{\mathcal{G}}x, y > = 0 \quad (6.16)$$

de tal maneira que

$$< x - P_{\mathcal{G}}x, y > = 0, \ \forall y \in \mathcal{G}. \ \square \quad (6.17)$$

Corolário 2 *Seja $\mathcal{G}$ um conjunto convexo fechado. Então, para qualquer $x \in \mathcal{H}$,*

$$Re < x - P_{\mathcal{G}}x, y - P_{\mathcal{G}}x > \leq 0, \quad (6.18)$$

para todo $y \in \mathcal{G}$. No sentido contrário, se algum $z \in \mathcal{G}$ tem a propriedade

$$Re < x - z, y - z > \leq 0 \quad (6.19)$$

para todo $y \in G$, então $z = P_{\mathcal{G}}x$.

Demonstração : A equação (6.13) é válida para qualquer conjunto convexo fechado $\mathcal{G}$. Portanto a primeira parte da demonstração é imediata. Quanto a outra parte, a equação (6.19) implica que

$$\| x - y \|^2 = \| x - z + z - y \|^2 = \| x - z \|^2 - 2Re < x - z, y - z > + \| z - y \|^2 \geq \| x - z \|^2 \quad (6.20)$$

para todo $y \in \mathcal{G}$. Assim,

$$\| x - y \|^2 \geq \| x - z \|^2$$

para todo $y \in \mathcal{G}$. Logo, $z = P_{\mathcal{G}} x$ por definição . □

Teorema 4 *Seja $\mathcal{G}$ um conjunto convexo qualquer. Então para todo par de elementos x e y em $\mathcal{H}$,*

$$\| P_{\mathcal{G}} x - P_{\mathcal{G}} y \|^2 \leq Re < x - y, P_{\mathcal{G}} x - P_{\mathcal{G}} y > . \tag{6.21}$$

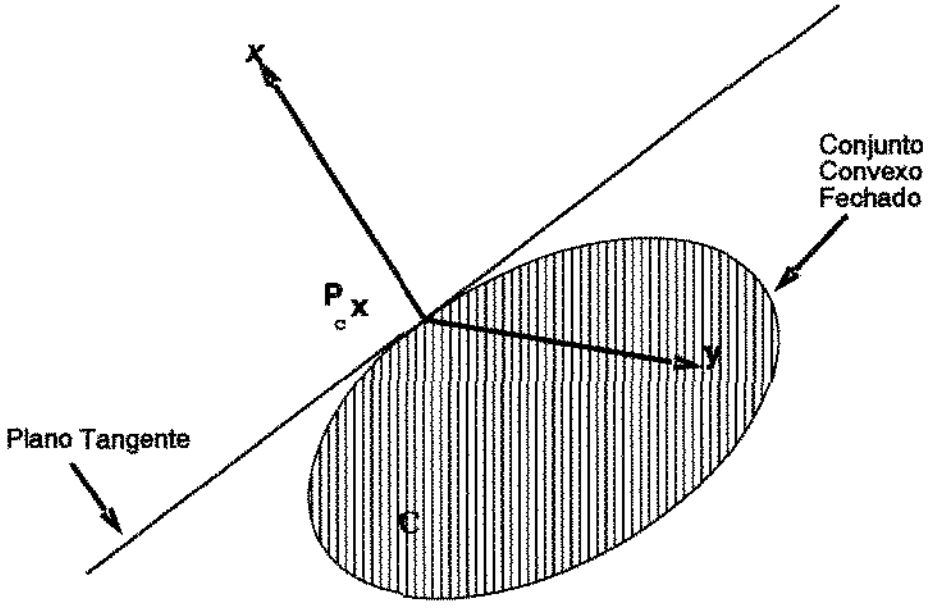


Figura 6.2: $\mathcal{C}$ é um conjunto convexo e fechado

Demonstração : Como $P_{\mathcal{G}} x$ e $P_{\mathcal{G}} y$ pertencem a $\mathcal{G}$, segue-se da equação (6.18) que

$$Re < x - P_{\mathcal{G}} x, P_{\mathcal{G}} y - P_{\mathcal{G}} x > \leq 0 \tag{6.22}$$

e

$$Re < y - P_{\mathcal{G}} y, P_{\mathcal{G}} x - P_{\mathcal{G}} y > \leq 0. \tag{6.23}$$

O resultado é obtido somando-se as equações acima. □

O próximo corolário será fundamental, pois possibilitará a utilização da teoria do ponto fixo para provar a convergência do algoritmo proposto. É introduzida também a noção

de operador não-expansivo que é um caso particular dos operadores de Lipschitiz[4]com constante de Lipschitiz igual a 1. Mais tarde, necessitaremos do conceito de operadores de contração [5], isto é, operadores de Lipschitiz com constante menor que 1.

Corolário 3 *Operadores de projeção sobre conjuntos convexos fechados $\mathcal{G}$ são não-expansíveis e, portanto, contínuos.*

Demonstração : A desigualdade de Schwartz aplicada a equação (6.21) fornece, para todo x e y em $\mathcal{H}$,

$$\| P_{\mathcal{G}}x - P_{\mathcal{G}}y \| \leq \| x - y \| . \tag{6.24}$$

Portanto, sob o operador $P_{\mathcal{G}}$ a distância entre duas imagens nunca excede a distância entre os pontos originais. Que é precisamente a definição de operador não-expansivo. Além disso, $\| x - y \|$ "pequeno" implica em $\| P_{\mathcal{G}}x - P_{\mathcal{G}}y \|$ "pequeno", logo $P_{\mathcal{G}}$ é contínuo. □

6.2 Operadores de Contração e Teorema do Ponto Fixo

Nesta seção , abordaremos os principais resultados sobre operadores de contração (definição logo em seguida) e provaremos o teorema do ponto fixo correspondente. A relevância deste teorema reside no fato de assegurarmos a convergência de muitos algoritmos iterativos que utilizam operadores de contração .

Definição 5 *Um operador $T : \mathcal{G} \to \mathcal{H}$ é um operador de contração se existe uma constante positiva θ,$0 < \theta < 1$, tal que*

$$\| Tx - Ty \| \leq \theta \| x - y \| \tag{6.25}$$

para todo $x, y \in \mathcal{G}$.

Definição 6 *Um ponto fixo de um operador $T : \mathcal{G} \to \mathcal{G}$ é um ponto $x \in \mathcal{G}$ tal que $Tx = x$.*

[4]Um operador P é chamado Lipschitiziano se dado $P : M \to N$, exista uma constante $c > 0$, chamada constante de Lipschitiz, tal que

$$\| Px - Py \| \leq c \| x - y \|$$

quaisquer que sejam $x, y \in M$.

[5]Daremos a definição rigorosa na próxima seção .

Qualquer operador de contração tem no máximo um ponto fixo. Pois se $Tx_1 = x_1$ e $Tx_2 = x_2$, então

$$\| x_1 - x_2 \| = \| Tx_1 - Tx_2 \| \leq \theta \| x_1 - x_2 \| \tag{6.26}$$

o que implica $\| x_1 - x_2 \| = 0$ e portanto $x_1 = x_2$.

O teorema seguinte é o mais conhecido na literatura. Ele nos fornece condições para que uma contração possua um ponto fixo e uma maneira iterativa de obtê-lo.

Teorema 5 *Se $\mathcal{G}$ é um subconjunto não vazio e fechado de $\mathcal{H}$, então qualquer operador de contração $T : \mathcal{G} \to \mathcal{G}$ possui um único ponto fixo $x_\infty \in \mathcal{G}$. Além disso, partindo-se de qualquer ponto $x_0 \in \mathcal{G}$, temos $T^n x_0 \to x_\infty$ quando $n \to \infty$.*

Demonstração : A unicidade já foi discutida. Resta-nos provar a existência e como obtê-lo através de iterações . Seja x_0 qualquer elemento de $\mathcal{G}$ e faça

$$x_n = Tx_{n-1}, \quad n = 1 \to \infty. \tag{6.27}$$

Então,

$$\| x_{n+1} - x_n \| \leq \theta \| x_n - x_{n-1} \| \leq \cdots \leq \theta^n \| x_1 - x_0 \| . \tag{6.28}$$

Assim, para qualquer $m > n$,

$$\| x_m - x_n \| \leq \| x_m - x_{m-1} \| + \| x_{m-1} - x_{m-2} \| + \cdots + \| x_{n+1} - x_n \| \tag{6.29}$$

$$\leq \theta^n (1 + \theta + \cdots + \theta^{m-n-1}) \| x_1 - x_0 \| \tag{6.30}$$

$$\leq \frac{\theta^n}{1-\theta} \| x_1 - x_0 \| . \tag{6.31}$$

Como $0 < \theta < 1$, segue-se que $\{x_n\}$ é uma sequência de Cauchy. Assim, como $\mathcal{G}$ é fechado, converge para um elemento $x_\infty \in \mathcal{G}$. Além disso,

$$\| Tx_\infty - x_\infty \| \leq \| Tx_\infty - x_{n+1} \| + \| x_{n+1} - x_\infty \| \leq \theta \| x_\infty - x_n \| + \| x_\infty - x_{n+1} \| \tag{6.32}$$

o lado direito da equação acima tende a zero quando $n \to \infty$. Logo, $\| Tx_\infty - x_\infty \| = 0$, isto é, $Tx_\infty = x_\infty$.

Devemos enfatizar ainda que a equação (6.28) nos fornece uma estimativa para o erro na n-ésima iteração , isto é,

$$\| x_\infty - x_n \| \leq \frac{\theta^n}{1-\theta} \| x_1 - x_0 \| . \tag{6.33}$$

onde percebemos que quanto menor for θ mais rápida será a convergência. □

Corolário 4 *Seja T^l, l um inteiro ≥ 1, um operador de contração de um conjunto fechado $\mathcal{G}$ em si mesmo. Então T possui um único ponto fixo x_∞ que é obtido como o limite de qualquer sequência $\{T^n x_0\}$, $x_0 \in \mathcal{G}$.*

Demonstração : De acordo com o teorema 5, existe um único $x_\infty \in \mathcal{G}$ de tal maneira que $T^l x_\infty = x_\infty$. Portanto,

$$Tx_\infty = T(T^l x_\infty) = T^l(Tx_\infty) \tag{6.34}$$

e $Tx_\infty \in \mathcal{G}$ é também um ponto fixo de T^l. Porém, pela unicidade, $Tx_\infty = x_\infty$ e, portanto x_∞ é ponto fixo de T. Consideremos, agora, a sequência $\{T^n x_0\}$ quando $n \to \infty$, e escrevamos $n = ql + r$ onde q é um inteiro e $0 \leq r < l$. Logo, quando $n \to \infty$, $q \to \infty$ e

$$\| T^n x_0 - x_\infty \| = \| T^r T^{ql} x_0 - T^r x_\infty \| \leq \| T^{ql} x_0 - x_\infty \| \to 0. \tag{6.35}$$

Consequentemente, $T^n x_0 \to x_\infty$ quando $n \to \infty$. □

A partir desse momento vamos tentar enfraquecer nossas hipóteses e tentar estender os resultados dos teoremas anteriores para esses novos casos.

Necessitamos de uma hipótese mais fraca. O conceito de operador não-expansivo é mais razoável do ponto de vista prático. Veja a definicão abaixo.

Definição 7 *Um operador $T : \mathcal{G} \to \mathcal{H}$ é chamado não-expansivo se*

$$\| Tx - Ty \| \leq \| x - y \| \tag{6.36}$$

para todos $x, y \in \mathcal{G}$.

O proóximo teorema foi proposto por Browder (veja [10]) mas a demonstração que apresentaremos é devida a Halpern (veja [11]).

Teorema 6 *Seja $T : \mathcal{G} \to \mathcal{G}$ um operador não-expansivo cujo domínio $\mathcal{G}$ é um conjunto limitado, convexo, fechado e não vazio. Então T tem pelo menos um ponto fixo.*

Pelo enunciado do Teorema 6, já percebemos o preço de substituir o operador de contração por operador não-expansivo: perdemos a unicidade do ponto fixo. Vamos à demonstração .

Demonstração : Seja y_0 um elemento de $\mathcal{G}$ pré-selecionado, e seja $\mathcal{G}_0 = \{x : x = y - y_0,\ y \in \mathcal{G}\}$. O conjunto $\mathcal{G}_0$ é limitado, fechado e convexo, além disso, contém o vetor zero ϕ[6]. Todo ponto $x \in \mathcal{G}_0$ possui uma decomposição única $x = y - y_0,\ \ y \in \mathcal{G}_0$. Seja $F : \mathcal{G}_0 \to \mathcal{G}_0$ definida como

$$Fx = Ty - y_0. \tag{6.37}$$

[6]Vamos denotar o vetor zero por ϕ.

Esse operador F é não-expansivo porque $x_1 = y_1 - y_0$ e $x_2 = y_2 - y_0$ implicam em

$$\| Fx_1 - Fx_2 \| = \| Ty_1 - Ty_2 \| \leq \| y_1 - y_2 \| \tag{6.38}$$

$$= \| (y_1 - y_0) - (y_2 - y_0) \| = \| x_1 - x_2 \| . \tag{6.39}$$

Para qualquer k fixo, $0 < k < 1$, o operador $G = kF$ é uma contração de $\mathcal{G}_0$ em si mesmo. Na verdade, para qualquer $x \in \mathcal{G}_0$, $kFx = k(Fx) + (1-k)\phi \in \mathcal{G}_0$ e para todos $x_1, x_2 \in \mathcal{G}_0$,

$$\| Gx_1 - Gx_2 \| = k \| Fx_1 - Fx_2 \| \leq k \| x_1 - x_2 \| . \tag{6.40}$$

Assim, utilizando-se o teorema 6 , existe para toda constante, k, $0 < k < 1$, um único $x_k \in \mathcal{G}_0$ tal que

$$x_k = kFx_k. \tag{6.41}$$

Agora, se pudermos mostrar que $x_k \to g$ quando $k \to 1$ por baixo, então pela continuidade de F , $g \in \mathcal{G}_0$ e pela equação (6.41) teremos que $g = Fg$. Ou, como $g = f - y_0$, $f \in \mathcal{G}$, obtemos que $f = Tf$, assim f é ponto fixo de T.

Portanto, devemos provar que

$$\lim_{k \to 1; 0<k<1} x_k = g \tag{6.42}$$

onde g é o (único) ponto fixo de F em $\mathcal{G}_0$ de norma mínima.

Suponhamos que $0 < k < l \leq 1$, $x_k = kFx_k$, $x_l = lFx_l$ e seja $h = x_l - x_k$. Portanto, como $\| Fx_l - Fx_k \| \leq \| x_l - x_k \|$, chegamos a

$$< \frac{x_k + h}{l} - \frac{x_k}{k}, \frac{x_k + h}{l} - \frac{x_k}{k} > \leq \| h \|^2 \tag{6.43}$$

ou

$$(\frac{1}{l} - \frac{1}{k})^2 \| x_k \|^2 + (\frac{1}{l^2} - 1) \| h \|^2 \leq 2l(\frac{1}{k} - \frac{1}{l}) Re < x_k, h > . \tag{6.44}$$

Então,

$$Re < x_k, h > \geq 0 \tag{6.45}$$

que juntamente com a identidade

$$\| x_l \|^2 = \| x_k + h \|^2 = \| x_k \|^2 + \| h \|^2 + 2Re < x_k, h >, \tag{6.46}$$

resulta na desigualdade abaixo

$$\| x_l \|^2 \geq \| x_k \|^2 + \| x_l - x_k \|^2 . \tag{6.47}$$

Para qualquer escolha da sequência $0 < k_1 < k_2 < \cdots$ tal que $k_i \to 1$, a sequência $\{\| x_k \|\}$ é monótona não-decrescente e limitada (porque $\mathcal{G}_0$ é limitado). Logo, converge e , em particular, da equação (6.47),

$$\| x_l - x_k \|^2 \leq \| x_l \|^2 - \| x_k \|^2 \to 0 \tag{6.48}$$

quando $k, l \to \infty$. Pela completude de $\mathcal{H}$, $x_k \to g \in \mathcal{G}_0$ porque $\mathcal{G}_0$ é fechado. (claro que o limite g é independente da escolha particular da sequência $k_i \to 1$.)

Finalmente, seja e um ponto fixo qualquer de F em $\mathcal{G}_0$. Então, $e = 1.Fe$ e podemos aplicar a equação (6.47) com $x_l = e$, $l = 1$, $x_k = x_k$, e $k = k_i$ para qualquer $i = 1 \to \infty$. Como $i \to \infty$, $x_k \to g$, então

$$\| e \|^2 \geq \| g \|^2 + \| e - g \|^2 \geq \| g \|^2 . \tag{6.49}$$

Ou seja, $\| g \| = \inf \| e \|$, onde e está sendo tomado entre os pontos fixos de F em $\mathcal{G}_0$ □

Vamos provar alguns lemas que serão utilizados na demonstração do próximo teorema.

Lema 1 *O conjunto dos pontos fixos $\mathcal{T}$ de um operador não-expansivo T com domínio convexo fechado $\mathcal{G}$ e imagem $\mathcal{H}$ é um conjunto convexo fechado.*

Demonstração : Seja $x_i = Tx_i$, $i = 1, 2, ...$, e suponha que $x_i \to x$. Como $\{x_i\} \subset \mathcal{G}$, que é, por sua vez, fechado, então $x \in \mathcal{G}$, e Tx está bem definido. Assim, utilizando o fato que T é um operador não-expansivo,

$$\| Tx - x \| = \| Tx - Tx_i + x_i - x \| \leq 2 \| x - x_i \| \to 0, \tag{6.50}$$

logo $Tx = x$ e $\mathcal{T}$ é fechado. Falta provar a convexidade.

Para quaisquer $x, y \in \mathcal{G}$, temos a seguinde igualdade

$$\| x - y \|^2 - \| Tx - Ty \|^2 = 4Re < Px - Py, (I - P)x - (I - P)y >, \tag{6.51}$$

onde I é o operador identidade e

$$P = \frac{(I + T)}{2}. \tag{6.52}$$

Porém, T é não-expansivo, portanto

$$Re < Px - Py, (I - P)x - (I - P)y > \geq 0 \tag{6.53}$$

para todos $x, y \in \mathcal{G}$. Como P e T têm precisamente os mesmos pontos fixos, é suficiente mostrar que o conjunto dos pontos fixos de P é convexo.

Seja y qualquer ponto fixo de P. Então, a equação (6.53) reduz-se a

$$Re < Px - y, x - Px > \geq 0 \tag{6.54}$$

para todo $x \in \mathcal{G}$. Reciprocamente, se algum $y \in$ satisfaz a equação (6.54) para todo $x \in \mathcal{G}$, em particular é satisfeita para $x = y$, que implica em $\| y - Py \| \leq 0$ ou $y = Py$. Resumindo, o conjunto dos pontos fixos de T é o conjunto de todos os $y \in \mathcal{G}$ que satisfazem a equação (6.54) para todo $x \in \mathcal{G}$. Entretanto, esse conjunto é convexo. □

Corolário 5 *Um operador T com domínio convexo fechado $\mathcal{G}$ é não-expansivo se, e somente se, $P = \dfrac{(I+T)}{2}$ satisfaz a equação (6.53) para todos $x, y \in \mathcal{G}$.*

Definição 8 *Um operador $T : \mathcal{G} \to \mathcal{H}$ é chamado de demi-fechado[7] se*

$$\{x_n\} \subset \mathcal{G}, \quad x_n \rightharpoonup x_0, \quad x_0 \in \mathcal{G}, \quad Tx_n \to y_0 \tag{6.55}$$

implicar em $Tx_0 = y_0$[8].

Definição 9 *Um operador $T : \mathcal{G} \to \mathcal{G}$ é dito assintoticamente regular se para todo $x \in \mathcal{G}$, $T^n x - T^{n+1} x \to \phi$.*

Lema 2 *Em um espaço de Hilbert $\mathcal{H}$ seja $\{x_n\}$ uma sequência que converge fracamente a x_0. Então, para qualquer $x \neq x_0$,*

$$\liminf_{n\to\infty} \| x_n - x \| > \liminf_{n\to\infty} \| x_n - x_0 \| . \tag{6.56}$$

Demonstração : Como uma sequência fracamente convergente é limitada, ambos os limites são finitos. Então, para provar a desigualdade basta utilizar a igualdade abaixo

$$\| x_n - x \|^2 = \| x_n - x_0 + x_0 - x \|^2 = \| x_n - x_0 \|^2 + \| x_0 - x \|^2 + 2Re < x_n - x_0, x_0 - x > \tag{6.57}$$

e notar que o último termo tende a zero quando $n \to \infty$. □

Lema 3 *Seja T qualquer operador não-expansivo com domínio convexo fechado $\mathcal{G} \subset \mathcal{H}$. Então, $I - T$ é demi-fechado.*

Demonstração : Seja $\{x_n\} \subset \mathcal{G}$ uma sequência que converge fracamente a x_0, e seja $\{x_n - Tx_n\}$ uma sequência que convirja fortemente a y_0. Então, como T é não-expansivo,

$$\liminf_{n\to\infty} \| x_n - x_0 \| \geq \liminf_{n\to\infty} \| Tx_n - Tx_0 \| = \liminf_{n\to\infty} \| Tx_n - x_n + x_n - Tx_0 \| \tag{6.58}$$

$$= \liminf_{n\to\infty} \| x_n - y_0 - Tx_0 \| \geq \liminf_{n\to\infty} \| x_n - x_0 \| \tag{6.59}$$

pelo lema 2. Ainda pelo mesmo lema, $x_0 = y_0 + Tx_0$ ou $(I - T)x_0 = y_0$ assim $I - T$ é demi-fechado. □

[7] Em inglês demiclosed. Outra tradução semi-fechado.

[8] T é demi-fechado se para qualquer sequência $\{x_n\} \subset \mathcal{G}$ que convergir fracamente para $x_0 \in \mathcal{G}$, a convergência forte da sequência $\{Tx_n\}$ a y_0 implicar em $Tx_0 = y_0$.

Teorema 7 *Seja $T : \mathcal{G} \to \mathcal{G}$ um operador assintoticamente regular e não-expansivo com domínio convexo fechado $\mathcal{G} \subset \mathcal{H}$ cujo conjunto de pontos fixos $\mathcal{T} \subset \mathcal{G}$ é não-vazio. Então, para cada $x \in \mathcal{G}$, a sequência $\{T^n x\}$ é fracamente convergente para um elemento de $\mathcal{T}$.*

Demonstração : Para todo $y \in \mathcal{T}$, a sequência $\{d_n\} = \{\| T^n x - y \|\}$ é não-crescente porque

$$d_{n+1} = \| T^{n+1}x - y \| = \| T(T^n x) - Ty \| \leq \| T^n x - y \| = d_n. \tag{6.60}$$

Portanto, o limite não-negativo

$$d(y) = \lim_{n \to \infty} \| T^n x - y \| \tag{6.61}$$

existe e é um número finito para todo $y \in \mathcal{T}$.

De acordo com o lema 1, $\mathcal{T}$ é um subconjunto convexo fechado de $\mathcal{G}$ e segue-se que para qualquer $\delta \geq 0$ fixo, o conjunto

$$\mathcal{T}_\delta = \{y \in \mathcal{T} : d(y) \leq \delta\} \tag{6.62}$$

é um subconjunto convexo fechado e limitado de $\mathcal{T}$ que é não-vazio para δ suficientemente grande. O conjunto $\mathcal{T}_\delta$ é fechado convexo e limitado como pode ser visto pelas desigualdades

$$\| y \| = \| y - T^n x + T^n x \| \leq \| T^n - y \| + \| T^n x \| \tag{6.63}$$

e

$$\| T^n x \| = \| T^n x - y_0 + y_0 \| \leq \| T^n x - y_0 \| + \| y_0 \|, \tag{6.64}$$

onde $y_0 \in \mathcal{T}$ é um elemento pré-selecionado. Para qualquer $y \in \mathcal{T}_\delta$

$$\| y \| \leq \delta + d(y_0) + \| y_0 \| . \tag{6.65}$$

A intersecção de todos os $\mathcal{T}_\delta$ é um conjunto convexo fechado e limitado $\mathcal{T}_{\delta_0}$. Onde δ_0 é o menor valor de δ para o qual $\mathcal{T}_\delta$ é não-vazio. O conjunto $\mathcal{T}_{\delta_0}$ só pode conter um único elemento, digamos y_0, pois suponhamos por contradição que haja também um elemento $y_1 \neq y_0$ assim a identidade

$$\| T^n x - \frac{y_0 + y_1}{2} \|^2 = \frac{1}{2}(\| T^n x - y_0 \|^2 + \| T^n x - y_1 \|^2) - \| \frac{y_0 - y_1}{2} \|^2 \tag{6.66}$$

resulta em $d[(\frac{y_0+y_1}{2})] < \delta_0$, o que contradiz a definição de δ_0.

A sequência $\{T^n x\}$ converge fracamente a y_0. De fato, como a sequência é limitada, é suficiente provar que todos os possíveis limites fracos de todas as suas subsequências é igual a y_0. Seja $T^{n'} x \rightharpoonup y_1 \neq y_0$. Então, pela regularidade assintótica de T,

$$T^{n'}x - T^{n'+1}x = (I-T)T^{n'}x \to \phi, \tag{6.67}$$

e relembrando que $I-T$ é demi-fechado, $(I-T)y_1 = \phi$;isto é, y_1 é um ponto fixo de T. Pelo lema 2,

$$\delta_0 = \lim \| T^{n'}x - y_0 \| > \lim \| T^{n'}x - y_1 \| = d(y_1), \tag{6.68}$$

que é uma contradição , pois vai contra a definição de δ_0. Assim, para todo $x \in \mathcal{G}$, a sequência $\{T^n x\}$ é fracamente convergente a um ponto fixo de T. □

Corolário 6 *A sequência $\{T^n x\}$ converge fortemente a y_0 se, e somente se, pelo menos uma de suas subsequências convergir fortemente.*

Demonstração : Sabemos que $T^n x \rightharpoonup y_0$, um ponto fixo de T, para toda escolha de $x \in \mathcal{G}$. Como os limites fraco e forte de uma sequência devem coincidir, o único limite forte possível da sequência $\{T^n x\}$ é justamente y_0. Mas pela igualdade

$$\| T^n x - y_0 \|^2 = \| T^n x \|^2 - 2Re < T^n x, y_0 > + \| y_0 \|^2, \tag{6.69}$$

obtemos que

$$\lim \| T^n x \|^2 = d^2(y_0) + \| y_0 \|^2 \tag{6.70}$$

quando $n \to \infty$. Em particular, para qualquer subsequência $\{T^{n'}x\}$ de $\{T^n x\}$,

$$\lim \| T^{n'}x \|^2 = d^2(y_0) + \| y_0 \|^2 . \tag{6.71}$$

Mas como já observamos, se qualquer subsequência $\{T^{n'}x\}$ convergir fortemente, então converge a y_0 e portanto $\lim \| T^{n'} \|^2 = \| y_0 \|^2$ o que exige que $d^2(y_0) = 0$ e

$$T^n x \to y_0. \quad \square \tag{6.72}$$

Definição 10 *Um operador $T : \mathcal{G} \to \mathcal{G}$ é chamado* reasonable wanderer [9] *se para todo $x \in \mathcal{G}$*

$$\sum_{n=0}^{\infty} \| T^n x - T^{n+1} x \|^2 < \infty. \tag{6.73}$$

é importante notar que *reasonable wanderer* implica em assintoticamente regular (veja definição 9).

Talvez seja ilustrativo dar um exemplo de operador não-expansivo, mas não assintoticamente regular. Consideremos o conjunto $\mathcal{G} = [-1, 1]$ contido na reta real, e definamos o operador T como $Tx = -x$. Tal operador é sem dúvidas não-expansivo, leva $\mathcal{G}$ em $\mathcal{G}$ e seu único ponto fixo é $x = 0$. Entretanto, $T^n x - T^{n+1}x = 2(-1)^n x$ que evidentemente não

[9] Optamos por não traduzir este termo, e escrevê- lo em itálico.

converge a zero quando $n \to \infty$, exceto se $x = 0$. Porém, Browder e Petryshyn mostraram que a combinação convexa linear

$$T_\alpha = \alpha I + (1-\alpha)T \tag{6.74}$$

de T e o operador identidade I é um operador *reasonable wanderer* para todo α no intervalo $0 < \alpha < 1$ e qualquer operador não-expansivo T.

Teorema 8 *Seja $T : \mathcal{G} \to \mathcal{G}$ um operador não-expansivo com domínio convexo fechado $\mathcal{G}$ e cujo conjunto de pontos fixos é não-vazio. Então para qualquer α, $0 < \alpha < 1$ fixo, $T_\alpha : \mathcal{G} \to \mathcal{G}$ é um operador* reasonable wanderer, *e a sequência $\{T_\alpha^n x\}$ converge fracamente para um ponto fixo de T para todo $x \in \mathcal{G}$.*

Demonstração : T_α leva $\mathcal{G}$ em $\mathcal{G}$, desse modo a iteração $x_n = T_\alpha^n x$ está bem definida para todo $x \in \mathcal{G}$. Além disso, para $\alpha \neq 1$ os pontos fixos de T e T_α coincidem e resta-nos provar que T_α é um operador *reasonable wanderer*; uma vez que é imediato o fato de ser não-expansivo. Seja $y \in \mathcal{G}$ um ponto fixo de T. Então, $y = Ty = T_\alpha y$ e

$$\| x_{n+1} - y \|^2 = \| \alpha x_n + (1-\alpha)Tx_n - y \|^2 \tag{6.75}$$
$$= \| \alpha(x_n - y) + (1-\alpha)(Tx_n - y) \|^2 \tag{6.76}$$
$$= \alpha^2 \| x_n - y \|^2 + 2\alpha(1-\alpha) Re < x_n - y, Tx_n - y > + (1-\alpha)^2 \| Tx_n - y \|^2 . \tag{6.77}$$

Analogamente,

$$\| x_n - Tx_n \|^2 = \| x_n - y - (Tx_n - y) \|^2 \tag{6.78}$$
$$= \| x_n - y \|^2 - 2Re < x_n - y, Tx_n - y > + \| Tx_n - y \|^2, \tag{6.79}$$

que após multiplicar-se por $(1-\alpha)$ e somar-se com a equação (6.78), resulta em

$$\| x_{n+1} - y \|^2 + \alpha(1-\alpha) \| x_n - Tx_n \|^2 \tag{6.80}$$
$$= \alpha \| x_n - y \|^2 + (1-\alpha) \| Tx_n - y \|^2 \leq \| x_n - y \|^2 . \tag{6.81}$$

Logo,

$$\alpha(1-\alpha) \| x_n - Tx_n \|^2 \leq \| x_n - y \|^2 - \| x_{n+1} - y \|^2 \tag{6.82}$$

de tal forma que

$$\alpha(1-\alpha) \sum_{n=0}^{N} \| x_n - Tx_n \|^2 = \frac{\alpha}{1-\alpha} \sum_{n=0}^{N} \| T_\alpha^n x - T_\alpha^{n+1} x \|^2 \tag{6.83}$$
$$\| x - y \|^2 - \| x_{N+1} - y \|^2 \leq \| x - y \|^2 . \tag{6.84}$$

Consequentemente, como $\alpha \neq 0$,

$$\sum_{n=0}^{\infty} \| T_\alpha^n x - T_\alpha^{n+1} x \|^2 < \infty. \quad \square \tag{6.85}$$

6.3 Projeções sobre Conjuntos Convexos para Reconstrução de Imagens Comprimidas

A técnica de projeções sobre conjuntos convexos no contexto de processamento de imagens foi introduzido, como já mencionamos, por Youla, confira na bibliografia o paper [3]. Descreveremos a seguir os passos gerais para solucionar um problema de reconstrução de imagens comprimidas, utilizando algoritmos iterativos.

Quando comprimimos uma imagem, há perda de dados na fase de quantização . E não há como recuperar fielmente os dados perdidos, pois lembremos que a quantização é uma função que leva muitos valores em um só. Quando exigimos uma alta taxa de compressão, mais dados são descartados, e em contrapartida a qualidade da imagem vai piorando cada vez mais. Surgem ruídos na imagem, os chamados artefatos. Assim temos uma relação de proporção inversa entre compressão versus qualidade da imagem.

Nos métodos de compressão baseados na transformada cosseno por bloco, quando a taxa de compressão é alta, o principal artefato é sem dúvida a descontinuidade entre blocos adjacentes. Visualmente, temos a sensação de ver uma imagem quadriculada. Este artefato é devido ao processamento separado de cada bloco da imagem, não levando-se em conta a relação entre os blocos entre si.

Para aplicar a técnica de projeções sobre conjuntos convexos, o primeiro passo é identificar todas as características disponíveis da imagem, tanto aquelas mais imediatas, isto é, as obtidas diretamente dos dados, quanto aquelas que conhecemos a priori (por exemplo, sabemos que a imagem original não apresentava os artefatos entre os blocos adjacentes).

Depois de coletadas as informações sobre a imagem, é necessário modelá-las em termos de conjuntos convexos fechados. Essa é talvez a parte mais trabalhosa da técnica, pois, em seguida, é preciso derivar os respectivos operadores de projeção .

Após todos esses preparativos, finalmente, podemos aplicar o algoritmo iterativo que consiste em efetuar, alternadamente, sucessivas projeções sobre os conjuntos convexos fechados previamente definidos (veja gráfico abaixo). A convergência é assegurada pelo teorema que vamos demonstrar a seguir.

Antes porém, vamos fixar a notação . Suponhamos que a imagem original f esteja na intersecção

$$C_0 = \bigcap_{i=1}^{m} C_i \tag{6.86}$$

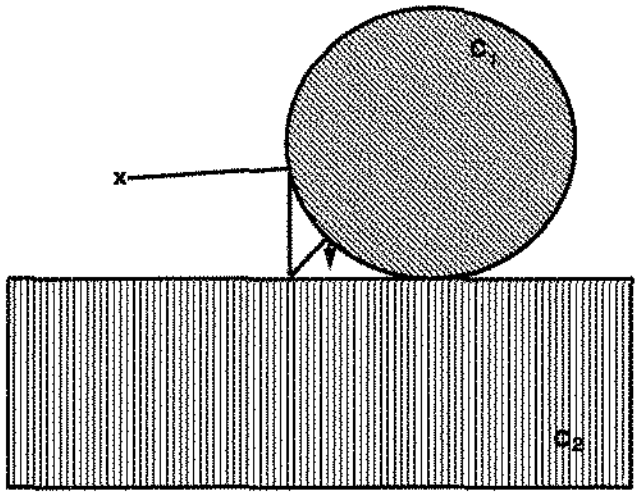


Figura 6.3: Projeções alternadas sobre conjuntos convexos

de m conjuntos convexos fechados $\mathcal{C}_i$, $i = 1, ..., m$. Evidentemente, $\mathcal{C}_0$ é um conjunto não-vazio, convexo e fechado. Denotaremos os projetores sobre $\mathcal{C}_0$ e $\mathcal{C}_i$ por P_0 e P_i, respectivamente. f é um ponto fixo de P_0 e de todos os P_i, $i = 1, ..., m$. Isto é, f é ponto fixo de P_0 e de todos

$$T_i = I + \lambda_i(P_i - I) \tag{6.87}$$

para qualquer escolha das constantes de relaxação $\lambda_1, \lambda_2, ..., \lambda_m$. Então sobre as mesmas condições , f é um ponto fixo de P_0 e da composição de operadores

$$T = T_m T_{m-1} \cdots T_1. \tag{6.88}$$

Alternativamente, podemos dizer que qualquer ponto fixo f de T pertence a $\mathcal{C}_0$? A sequência $\{T^n x\}$ converge (fracamente ou fortemente) a um ponto de $\mathcal{C}_0$ quando $n \to \infty$ independentemente da escolha de $x \in \mathcal{H}$? As respostas estão em parte contidas no teorema abaixo.

Teorema 9 *Seja $\mathcal{C}_0$ um conjunto não-vazio. Então, para todo $x \in \mathcal{H}$ e qualquer escolha das constantes $\lambda_1, \lambda_2, ..., \lambda_m$ no intervalo $0 < \lambda < 2$, a sequência $\{T^n x\}$ converge fracamente a um ponto de $\mathcal{C}_0$.*

Demonstração : Para $\lambda \geq 0$, todo operador

$$T_i = I + \lambda_i(P_i - I) = (1 - \lambda_i)I + \lambda_i P_i \tag{6.89}$$

é não-expansivo. A afirmação é verdadeira se $0 \leq \lambda_i \leq 1$, mas se $1 < \lambda_i$ temos que $1 - \lambda_i < 0$, por isso precisaremos ter um pouco mais de cuidado. Com a ajuda da equação (6.22) provamos que o operador é não-expansivo, ou seja,

$$\| T_i x - T_i y \|^2 = \| (1-\lambda_i)(x-y) + \lambda_i(P_i x - P_i y) \|^2 \quad (6.90)$$

$$= (1-\lambda_i)^2 \| x-y \|^2 + 2\lambda_i(1-\lambda_i) Re < x-y, P_i x - P_i y > + \lambda_i^2 \| P_i x - P_i y \|^2 \quad (6.91)$$

$$\leq (1-\lambda_i)^2 \| x-y \|^2 + (\lambda_i^2 + 2\lambda_i(1-\lambda_i)) \| P_i x - P_i y \|^2 \quad (6.92)$$

$$= (1-\lambda_i)^2 \| x-y \|^2 + \lambda_i(2-\lambda_i) \| P_i x - P_i y \|^2 \quad (6.93)$$

$$\leq (\lambda_i(2-\lambda_i) + (1-\lambda_i)^2) \| x-y \|^2 = \| x-y \|^2 . \quad (6.94)$$

Vamos demonstrar que T é um operador *reasonable wanderer*.
Para $m = 1$, temos $T = T_1$, $\mathcal{C}_0 = \mathcal{C}_1$, e

$$\| x - Tx \|^2 = \lambda_i^2 \| x - P_1 x \|^2 . \quad (6.95)$$

Além disso, para qualquer $y \in \mathcal{C}_0$, $Ty = P_1 y = y$ e

$$\| Tx - y \|^2 = \| x - y + \lambda_1(P_1 x - x) \|^2 \quad (6.96)$$

$$= \| x-y \|^2 + 2\lambda_i Re < x-y, P_1 x - x > + \lambda_1^2 \| x - P_1 x \|^2 \quad (6.97)$$

$$= \| x-y \|^2 - \lambda_1(2-\lambda_1) \| x - P_1 x \|^2 + 2\lambda_1 Re < x - P_1 x, y - P_1 y > \quad (6.98)$$

$$\leq \| x-y \|^2 - \lambda_1(2-\lambda_1) \| x - P_1 x \|^2 . \quad (6.99)$$

Assim, combinando a equação (6.95) e última equação chegamos ao seguinte resultado

$$\| x - Tx \|^2 \leq \frac{\lambda_1}{2-\lambda_1} (\| x-y \|^2 - \| Tx - y \|^2), \quad (6.100)$$

para $0 < \lambda_1 < 2$.

Para $m \geq 1$, provamos por indução em m, resultando na desigualdade

$$\| x - Tx \|^2 \leq b_m 2^{m-1} (\| x-y \|^2 + \| Tx - y \|^2), \quad (6.101)$$

onde

$$b_m = \sup_{1 \leq m \leq m} \{ \frac{\lambda_i}{2-\lambda_i} \}. \quad (6.102)$$

De fato, seja $T = T_m K$, onde

$$K = T_{m-1} T_{m-2} \cdots T_1, \quad (6.103)$$

e observemos que para $m \geq 2$,

$$\| x - Tx \|^2 = \| x - Kx + Kx - Tx \|^2 \quad (6.104)$$

$$\leq (\| x - Kx \| + \| Kx - Tx \|)^2 \quad (6.105)$$

$$\leq 2(\| x - Kx \|^2 + \| Kx - Tx \|^2) \quad (6.106)$$

$$\leq 2(\| x - Kx \|^2 + 2^{m-2} \| Kx - T_m K x \|^2). \quad (6.107)$$

Então pela hipótese de indução ,

$$\| x - Tx \|^2 \leq b_m 2(2^{m-2} \| x - y \|^2 - 2^{m-2} \| Kx - y \|^2 \tag{6.108}$$
$$+2^{m-2} \| Kx - y \|^2 - 2^{m-2} \| Tx - y \|^2) \tag{6.109}$$
$$= b_m 2^{m-1}(\| x - y \|^2 - \| Tx - y \|^2), \tag{6.110}$$

que é a desigualdade desejada.

Logo,

$$\sum_{n=0}^{\infty} \| T^n x - T^{n+1} \|^2 \leq b_m 2^{m-1} \| x - y \|^2 < \infty \tag{6.111}$$

e T portanto é um operador *reasonable wanderer* e, consequentemente, assintoticamente regular. Pelo teorema 7, a sequência $\{T^n x\}$ converge fracamente para um ponto fixo de T.

Porém, os pontos fixos de T coincidem com os pontos fixos de $\mathcal{C}_0$, isto é, a intersecção dos conjuntos $\mathcal{C}_i$, $i = 1, ..., m$. Assim, se $x \in \mathcal{C}_0$ implica $x = Tx$, pois $x \in \mathcal{C}_i$, $i = 1, ..., m$. Ou alternativamente, se $x = Tx$ e $y \in \mathcal{C}_0$,

$$\| x - y \| = \| Tx - Ty \| \leq \| T_1 x - T_1 y \| = \| T_1 x - y \| \leq \| x - y \|; \tag{6.112}$$

logo, $\| x - y \| = \| T_1 x - y \|$. Mas utilizando-se a inequação (6.100), isso só é possível se $x = T_1 x$ de tal forma que $x = T_m T_{m-1} \cdots T_2 x$, e repetindo-se o mesmo argumento provamos que $x \in \mathcal{C}_i$, $i = 1, ..., m$, isto é, $x \in \mathcal{C}_0$ e desta forma a demonstração está completa. □

Falta-nos ainda discutir a questão da convergência forte. Relembremos alguns resultados importantes. Seja $T : \mathcal{C} \to \mathcal{C}$ um operador não-expansivo, assintoticamente regular com domínio $\mathcal{C} \subset \mathcal{H}$ convexo fechado e cujo conjunto de pontos fixos $\mathcal{T}$ seja não-vazio. Então pelo corolário do Teorema 7, a sequência $\{T^n x\}$ converge fortemente para um elemento de $\mathcal{T}$ se, e somente se, pelo menos uma de suas subsequências convergir fortemente. Obviamente, isto pode ser assegurado se todas as iterações $T^n x$ estiverem contidas em um compacto ou um subconjunto de dimensão finita de $\mathcal{H}$.

Capítulo 7

Algoritmos de Recuperação de Imagens

Os métodos baseados na transformada cosseno por bloco são os mais utilizados para compressão de imagens. Prova disso é que o padrão internacional de compressão de imagens utiliza essa técnica. Entretanto, se muitas informações são descartadas durante a quantização para se obter altas taxas de compressão, então surgem na imagem ruídos, sendo os mais frequentes os artefatos entre blocos adjacentes. Ou seja, aparecem "descontinuidades" entre os blocos vizinhos, causando a sensação visual de se ver uma imagem toda quadriculada.

Utilizaremos o método de projeções ortogonais sobre conjuntos convexos, apresentado no Capítulo 6, para aprimorar a qualidade de imagens que apresentem artefatos entre blocos adjacentes devido à alta taxa de compressão.

A partir de propriedades que desejaríamos que a imagem tivesse (informação a priori), definiremos conjuntos convexos fechados e derivaremos os seus respectivos operadores de projeção . Aplicando em seguida o algoritmo iterativo de projeções sucessivas para obter um elemento da interseção dos conjuntos previamente estabelecidos. A convergência do método é justificada pelo Teorema 7 demonstrado no capítulo precedente.

Toda imagem A, de $N \times N$, dada em sua forma matricial será vista como um vetor real $f \in \Re^{N^2 \times 1}$, obtido ordenando-se as colunas (ou linhas) da matriz A em um vetor. A transformada cosseno por blocos será vista como uma transformação linear $B : \Re^{N^2} \rightarrow \Re^{N^2}$. Portanto, para uma imagem qualquer f, sua transformada cosseno por blocos será denotada por

$$F = Bf, \tag{7.1}$$

e a transformada inversa, como B é real e unitária, será

$$f = B^t F. \tag{7.2}$$

A matriz B, para uma imagem de $N \times N$ que é dividida em blocos de $M \times M$, é uma matriz de $N^2 \times N^2$ bloco-diagonal com $\left(\frac{N}{M}\right)^2$ matrizes de dimensão $M^2 \times M^2$ ao longo da diagonal. Estas matrizes são identicas e sua expressão é bem conhecida (veja [13]).

Os elementos de F, em (7.1), são chamados de coeficientes transformados ou coeficientes do espaço de frequência. E podem ser vistos como as coordenadas de f na nova base formada pelas colunas de B. Após obtidos, os coeficientes transformados são quantizados afim de eliminar-se as informações redundantes e conseguir-se compressão. A quantização pode ser vista como um operador $\mathcal{Q} : \Re^{N^2} \rightarrow \Re^{N^2}$. Então, podemos escrever

$$F' = \mathcal{Q}F, \tag{7.3}$$

onde F' denota os coeficientes transformados e quantizados.

O operador de quantização é não-linear e possui a seguinte propriedade

$$\mathcal{Q}^2 = \mathcal{Q}. \tag{7.4}$$

Todo o processo de codificação /decodificação pode ser modelado da seguinte forma. No codificador obteremos F' como segue

$$F' = \mathcal{Q}Bf. \tag{7.5}$$

E no decodificador a imagem é recuperada parcialmente, pois a operação de quantização é essencialmente irreversível. Isto quer dizer que no decodificador não recuperamos a imagem original f, mas uma "aproximação " f' dada pela transformada inversa de F', ou seja,

$$f' = B^t F'. \tag{7.6}$$

7.1 Propriedades da Imagem em Termos de Conjuntos Convexos Fechados

Como o nosso objetivo é encontrar uma melhor aproximação para f, utilizando a técnica discutida no Capítulo 5, precisamos escolher algumas características importantes da imagem f e modelá-las em termos de conjuntos convexos fechados.

Uma primeira propriedade é que a imagem que procuramos depois de transformada e quantizada seja compatível com os dados que possuímos, isto é,

$$C_T' \triangleq \{f : \mathcal{Q}Bf = F'\}. \tag{7.7}$$

O conjunto acima é não-vazio, pois a imagem original $f \in C'_T$. A imagem f' também pertence a C'_T devido a propriedade da equação (7.4) do operador Q. Entretanto, o conjunto C'_T não é fechado, porque geralmente os intervalos de quantização não são fechados. Por isso, utilizamos o conjunto C_T, seu fecho.

$$C_T = \{f : F_n^{min} \leq (Bf)_n \leq F_n^{max}\}, \tag{7.8}$$

onde o subíndice n indica a n-ésima componente do vetor e F_n^{min} e F_n^{max} são determinados pelo quantizador utilizado. É fácil provar que C_T é convexo e fechado. O seu operador de projeção P_T é dado por:

$$P_T f = B^t F, \tag{7.9}$$

onde o vetor F é determinado por

$$F_n = \begin{cases} F_n^{min} & \text{se } (Bf)_n < F_n^{min} \\ F_n^{max} & \text{se } (Bf)_n > F_n^{max} \\ (Bf)_n & \text{se } F_n^{min} \leq (Bf)_n \leq F_n^{max} \end{cases} \quad 1 \leq n \leq N^2 \tag{7.10}$$

Uma segunda característica importante é baseada numa informação que conhecemos a priori: a imagem original, em princípio, não apresenta descontinuidades entre os blocos. Devemos, portanto, exigir uma certa suavidade entre os blocos adjacentes. Para isso, precisaremos definir mais dois conjuntos C_c e C_l, o primeiro para forçar suavidade entre colunas de blocos adjacentes, e o último conjunto para forçar suavidade entre as linhas de blocos vizinhos.

Para definirmos o conjunto C_c, a imagem f, $N \times N$ será representada como segue

$$f = \{f_1, f_2, f_3, ..., f_N\}, \tag{7.11}$$

onde f_i denota a i-ésima coluna da imagem. Seja D um operador linear tal que Df represente a diferença entre colunas de blocos adjacentes da imagem f. Por exemplo, para o caso em que $N = 512$[1] e para blocos de 8×8,

$$Df = \begin{bmatrix} f_8 - f_9 \\ f_{16} - f_{17} \\ \vdots \\ f_{504} - f_{505} \end{bmatrix}. \tag{7.12}$$

A norma de Df

$$\| Df \| = \left[\sum_{i=1}^{63} \| f_{8.i} - f_{8.i+1} \|^2 \right]^{1/2} \tag{7.13}$$

[1] Usaremos muitas vezes $N = 512$ para ilustrar nossos exemplos.

dá uma idéia da variação total entre as colunas de blocos adjacentes[2].

Podemos, portanto, definir C_c como

$$C_c \triangleq \{f : \| Df \| \leq E_c\}, \tag{7.14}$$

onde E_c é um limitante superior para o diâmetro do conjunto. A escolha de E_c será discutida mais a frente.

O conjunto C_c é na verdade um elipsóide e, consequentemente, fechado e convexo.

Resta-nos derivar o operador de projeção sobre C_c. Para isso, consideremos um vetor no espaço $\Re^n \times \Re^n$ como uma dupla de vetores em $\Re^n$, isto é, (x, y), onde $x, y \in \Re^n$. Assim, se (x, y) , $(u, v) \in \Re^n \times \Re^n$, então podemos definir o produto interno entre eles como

$$< (x, y), (u, v) > = < x, u > + < y, v >, \tag{7.15}$$

onde $< \cdot, \cdot >$ denota o produto interno usual em $\Re^n$. O produto interno (7.15) induz a seguinte norma

$$\| (x, y) \| = \left[\| x \|^2 + \| y \|^2 \right]^{1/2}, \tag{7.16}$$

onde $\| \cdot \|$ é a norma euclidiana em $\Re^n$.

Seja

$$C = \{(x, y) : \| x - y \| \leq E_c, \quad x, y \in \Re^n\} \tag{7.17}$$

um subconjunto fechado e convexo de $\Re^n \times \Re^n$.

Teorema 10 *Para um vetor arbitrário $(x, y) \in \Re^n \times \Re^n$, a projeção , $P(x, y)$, sobre o conjunto C é dada por*

$$P(x, y) = \begin{cases} (x, y) & \text{se } \| x - y \| \leq E_c \\ (\alpha x + (1 - \alpha) y, (1 - \alpha) x + \alpha y) & \text{caso contrário} \end{cases} \tag{7.18}$$

onde $\alpha = \left[\dfrac{E_c}{\| x - y \|} + 1 \right]$.

Demonstração : Por definição , $P(x, y)$ é o vetor em C que satisfaz

$$\| (x, y) - P(x, y) \| = \min_{(u,v) \in \Re^n \times \Re^n} \| (x, y) - (u, v) \|, \tag{7.19}$$

onde $(x, y) - (u, v) = (x - u, y - v)$, e $(u, v) \in C$.

Se $(x, y) \in C$, isto é, $\| x - y \| \leq E_c$, então, é trivial, $P(x, y) = (x, y)$. No caso em que $(x, y) \notin C$, precisaremos resolver o seguinte problema de otimização :

[2]A norma interna na equação (7.11) é a norma euclidiana.

$$\min_{s.a\ \|x-y\|^2 \leq E_c^2} \| (x,y) - (u,v) \|^2 \tag{7.20}$$

Que é equivalente a minimizar a função de Lagrange:

$$J_\lambda = \| (x,y) - (u,v) \|^2 + \lambda \left[\| u - v \|^2 - E_c^2 \right], \tag{7.21}$$

onde λ é o multiplicador correspondente (veja [22]). A condição de primeira ordem diz que o gradiente da função J_λ com respeito a u e v deve ser zero, isto é,

$$u - x + \lambda(u - v) = 0 \tag{7.22}$$

e

$$v - y + \lambda(y - u) = 0, \tag{7.23}$$

portanto

$$u = \alpha x + (1 - \alpha) y \tag{7.24}$$

e

$$v = (1 - \alpha) x + \alpha y, \tag{7.25}$$

onde $\alpha = \dfrac{1 + \lambda}{1 + 2\lambda}$. Além disso, se substituirmos em $\| u - v \| = E_c$ o valor de u da equação (7.24) e o valor de v da equação (7.25), obtemos, resolvendo em λ, que $\lambda = \left[\frac{\|x-y\|}{E_c} - 1 \right] / 2$. □

Aproveitando-se o teorema acima, podemos derivar o projetor sobre C_c escrevendo a imagem f, $N \times N$, na sua forma vetorial dada em (7.11), e supondo que $\tilde{f}$ seja a projeção de f sobre C_c, temos que $\tilde{f}$ minimiza a função distância

$$\| f - \tilde{f} \|^2 = \sum_{i=1}^{N} \| f_i - \tilde{f}_i \|^2, \tag{7.26}$$

onde $\tilde{f} = \{\tilde{f}_1, ..., \tilde{f}_N\} \in C_c$. No caso $N = 512$ e blocos 8×8, $\| D\tilde{f} \| \leq E_c$ só contém as colunas de blocos vizinhos.

Definimos

$$x = \begin{bmatrix} f_8 \\ f_{16} \\ \vdots \\ f_{504} \end{bmatrix} \tag{7.27}$$

$$y = \begin{bmatrix} f_9 \\ f_{17} \\ \vdots \\ f_{505} \end{bmatrix}. \tag{7.28}$$

Logo, $D\tilde{f} = x - y$. Segue-se do teorema acima que a função distância definida em (7.26) é minimizada quando

$$\tilde{f}_i = \alpha f_i + (1-\alpha) f_{i+1} \tag{7.29}$$

e

$$\tilde{f}_{i+1} = (1-\alpha) f_i + \alpha f_{i+1}, \tag{7.30}$$

para $i = 8k$ e $k = 1, 2, ..., 63$, onde $\alpha = \frac{1}{2}\left[\frac{E_c}{\| Df \|} + 1\right]$ e $\tilde{f}_i = f_i$ caso contrário.

Os resultados acima podem ser generalizados para qualquer imagem e para qualquer dimensão dos blocos. O conjunto C_l e o seu projetor P_l são obtidos de forma análoga. Além disso, podemos definir conjuntos semelhantes para colunas (ou linhas) no interior de cada bloco.

7.2 Algoritmo 1

Utilizando-se os conjuntos C_T, C_c e C_l e seus respectivos projetores P_T, P_c e P_l podemos sugerir o seguinte algoritmo para recuperar a imagem

1. faça $f_0 = f'$;
2. para $k = 1, 2, ...$ calcule iterativamente f_k da seguinte maneira

$$f_k = P_l P_c P_T f_{k-1}$$

3. continue até que $\| f_k - f_{k-1} \| \leq TOL$, onde TOL é uma tolerância pré-estabelecida.

A convergência do algoritmo é assegurada pelo Teorema 9 do capítulo 6, lembrando que temos convergência forte por trabalharmos em espaço de dimensão finita[3].

Na definição do conjunto C_c há uma indeterminação a ser discutida: a escolha de E_c, que é um limitante para a largura do conjunto. Faz algum sentido escolher E_c da ordem de 10^{-1} ? Ou quem sabe da ordem de 10^3 ? Há alguma estimativa para E_c ?

Para imagens em geral, E_c da ordem de 10^{-1} não faz sentido, pois estaríamos exigindo que $\| Df \| \leq 10^{-1}$ o que é bem pouco provável de acontecer. A não ser que estivessemos trabalhando com uma imagem que tivesse a característica de possuir colunas de blocos adjacentes muito – extremamente – parecidas, quase idênticas. Isso é muito raro (raríssimo) de acontecer. Por outro lado, uma estimativa da ordem de 10^3 é mais plausível; entretanto depende de fatores como dimensão da imagem, riqueza de detalhes, etc. Enfim, depende da imagem. Portanto, a saída é procurar uma estimativa que esteja relacionada com a

[3]Lembremos que convergência fraca implica convergência forte em espaços de dimensão finita. Veja teorema 4.8-4 do livro do Kreyszig [14].

imagem em questão. Um ponto importante é perceber que não podemos exigir muita suavidade entre as colunas de blocos adjacentes, porque a imagem original pode não possuir tal propriedade. Aliás, podemos afirmar que a maioria das imagens não a possuem. Pois, pode acontecer que entre dois blocos vizinhos haja uma região de fronteira entre dois objetos na imagem que em termos visuais se traduz em descontinuidade de cores (um pode ser preto e outro branco). Portanto, o ideal é propor uma estimativa não muito exigente no sentido exposto.

Em [1], E_c é estimado da seguinte forma. Seja f' a imagem com os artefatos entre blocos. Reordenamos f' na sua forma vetorial por colunas

$$f' = \{f'_1, ..., f'_{512}\}, \tag{7.31}$$

onde f'_i denota a i-ésima coluna da imagem f'. E_c é estimado a partir de f' como segue

$$S_k \triangleq \left[\sum_{i=1}^{63} \| f'_{8i+k} - f'_{8i+k+1} \|^2\right]^{1/2}, \tag{7.32}$$

e

$$E_c = \frac{1}{7}\sum_{k=1}^{7} S_k, \tag{7.33}$$

onde $k = 1, ..., 7$.

7.3 Algoritmo Espaço-Adaptativo

Em 1995, foi publicado um outro artigo pelos mesmos autores de [1] (veja [2]) onde fazia-se uma alteração na definição dos conjuntos C_c e C_l. A motivação para essa mudança está baseada no fato que, anteriormente, não se respeitavam as características espaciais da imagem, tratando as colunas (ou linhas) entre blocos de forma uniforme.

Olhando-se, por exemplo, a imagem da *Lena*[4], notamos que os artefatos são mais visíveis no rosto e "regiões suaves", isto é, regiões que não sejam cheias de bordas ou de texturas diferentes. Já no cabelo e enfeites do chapéu os artefatos são menos significativos.

Essa observação nos sugere um tratamento diferenciado para regiões suaves e regiões cheia de fronteiras. No primeiro caso, podemos exigir maior suavidade; no segundo, podemos ser menos exigentes. Assim, nosso modelo dos conjuntos C_c e C_l deve conter, de alguma forma, pesos de suavização para regiões determinadas, resultando-se em um algoritmo espaço-adaptativo.

Df definido em (7.12) para uma imagem de 512×512 e blocos de 8×8 é um vetor de (512×63) componentes. Seja

[4]Esta imagem é largamente utilizada em Processamento Digital de Sinais. Há várias imagens "Lena" no Capítulo 8 dessa dissertação .

$$W = \begin{bmatrix} w_1 & 0 & 0 & \cdots & 0 \\ 0 & w_2 & 0 & \cdots & \vdots \\ 0 & 0 & \ddots & 0 & \vdots \\ \vdots & \vdots & 0 & \ddots & 0 \\ 0 & \cdots & \cdots & 0 & w_{512\times 63} \end{bmatrix}, \tag{7.34}$$

onde os w_i, $i = 1, 2, ..., (512 \times 63)$ são os pesos de suavização que dependem de característica da imagem.

Portanto, o vetor WDf é uma versão ponderada da diferença de colunas de blocos adjacentes. E podemos definir o conjunto

$$C_w \triangleq \{f : \| WDf \| \leq E\}. \tag{7.35}$$

O conjunto C_w é convexo e fechado. Vamos supor que conhecemos os w_i, e derivemos o operador de projeção sobre C_w. Para isso, precisaremos do seguinte teorema.

Teorema 11 *Sejam x_0, y_0, u e v vetores em $\Re^n$ com a norma Euclidiana $\| \cdot \|$. Seja W uma matriz $n \times n$ arbitrária. Então, sujeito a restrição $\| W(u - v) \| = E$, o funcional*

$$\psi(u, v) = \| u - x_0 \|^2 + \| v - y_0 \|^2 \tag{7.36}$$

é minimizado quando

$$u = \frac{1}{2}\left[I + (I + 2\lambda W^t W)^{-1}\right] x_0 + \left[I - (I + 2\lambda W^t W)^{-1}\right] y_0 \tag{7.37}$$

e

$$v = \frac{1}{2}\left[I - (I + 2\lambda W^t W)^{-1}\right] x_0 + \left[I + (I + 2\lambda W^t W)^{-1}\right] y_0, \tag{7.38}$$

onde λ é a única raiz positiva encontrada resolvendo-se a equação $\| W(u - v) \| = E$.

Demonstração : Tomemos o Lagrangeano

$$J_\lambda(u, v) = \psi(u, v) + \lambda(\| W(u - v) \|^2 - E^2), \tag{7.39}$$

calculando-lhe o gradiente com respeito a u e v, e igualando-o a zero, obtemos

$$(u - x_0) + \lambda W^t W(u - v) = 0 \tag{7.40}$$

e

$$(v - y_0) + \lambda W^t W(v - u) = 0. \tag{7.41}$$

Portanto, $u + v = x_0 + y_0$ e $(I + 2\lambda W^t W)(u - v) = x_0 - y_0$. O resultado segue-se imediatamente. □

Escrevendo-se a imagem f na sua forma vetorial ordenada por colunas , chamando-se a projeção de f sobre C_w de $\tilde{f}$ e sabendo-se que $\tilde{f}$ minimiza a distância

$$\| f - \tilde{f} \|^2 = \sum_{i=1}^{N} \| f_i - \tilde{f}_i \|^2, \tag{7.42}$$

onde $\tilde{f} = \{\tilde{f}_1, ..., \tilde{f}_N\} \in C_w$, então podemos obter o operador de projeção utilizando o Teorema 11, isto é, para uma imagem $f \notin C_w$, sendo P_w o operador de projeção sobre C_w, podemos escrever

$$\tilde{f} = P_w f = \{\tilde{f}_1, ..., \tilde{f}_{512}\}. \tag{7.43}$$

Definimos

$$x_0 = \begin{bmatrix} f_8 \\ f_{16} \\ \vdots \\ f_{504} \end{bmatrix}, \tag{7.44}$$

$$y_0 = \begin{bmatrix} f_9 \\ f_{17} \\ \vdots \\ f_{505} \end{bmatrix} \tag{7.45}$$

e

$$u = \begin{bmatrix} \tilde{f}_8 \\ \tilde{f}_{16} \\ \vdots \\ \tilde{f}_{504} \end{bmatrix}, \tag{7.46}$$

$$v = \begin{bmatrix} \tilde{f}_9 \\ \tilde{f}_{17} \\ \vdots \\ \tilde{f}_{505} \end{bmatrix}. \tag{7.47}$$

Com essa notação , utilizando-se o Teorema 11,

$$u = \frac{1}{2}\left[I + (I + 2\lambda W^t W)^{-1}\right] x_0 + \frac{1}{2}\left[I - (I + 2\lambda W^t W)^{-1}\right] y_0 \tag{7.48}$$

e

$$v = \frac{1}{2}\left[I - (I + 2\lambda W^t W)^{-1}\right] x_0 + \frac{1}{2}\left[I + (I + 2\lambda W^t W)^{-1}\right] y_0, \tag{7.49}$$

$\tilde{f}_i = f_i$ para $i \neq 8.k$ ou $i \neq 8.k+1$, $k = 1, 2, ..., 63$, onde o escalar λ é a raiz positiva da equação não-linear

$$\| WQ\tilde{f} \| = E. \tag{7.50}$$

A matriz W é diagonal o que simplifica bastante as contas. Por exemplo, a matriz W^tW é diagonal e positiva definida. Isto é importante porque nos garante que a matriz $(I + 2\lambda W^tW)^{-1}$ existe, pois $\lambda > 0$[5], e é facilmente obtida. Assim o operador P_w está bem definido.

Para compreender melhor o operador P_w, vamos reescrever a matriz W como segue

$$W = \begin{bmatrix} \mathbf{w_1} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{w_2} & \mathbf{0} & \cdots & \vdots \\ \mathbf{0} & \mathbf{0} & \ddots & \mathbf{0} & \vdots \\ \vdots & \vdots & \mathbf{0} & \ddots & \mathbf{0} \\ \mathbf{0} & \cdots & \cdots & \mathbf{0} & \mathbf{w_{63}} \end{bmatrix}, \tag{7.51}$$

onde $\mathbf{w_k}$ é uma matriz 512×512 diagonal correspondente à k-ésima fronteira entre blocos adjacentes. Para $k = 1, 2, ..., 63$, $\mathbf{w_k}$ é escrita da seguinte forma

$$\mathbf{w_k} = \begin{bmatrix} w_1^k & 0 & 0 & \cdots & 0 \\ 0 & w_2^k & 0 & \cdots & \vdots \\ 0 & 0 & \ddots & 0 & \vdots \\ \vdots & \vdots & 0 & \ddots & 0 \\ 0 & \cdots & \cdots & 0 & w_{512}^k \end{bmatrix}, \tag{7.52}$$

onde a relação dos elementos de $\mathbf{w_k}$ e da matriz W é

$$w_j^k = w_{512.k-1+j}, \quad \text{para } j = 1, 2, ..., 512. \tag{7.53}$$

Usando a notação introduzida em (7.51) e (7.52) podemos rescrever (7.47) e (7.49) como segue

$$\tilde{f}_i = \frac{1}{2}\left[I + (I + 2\lambda \mathbf{w_k}^t \mathbf{w_k})^{-1}\right] f_i + \frac{1}{2}\left[I - (I + 2\lambda \mathbf{w_k}^t \mathbf{w_k})^{-1}\right] f_{i+1}, \tag{7.54}$$

$$\tilde{f}_{i+1} = \frac{1}{2}\left[I - (I + 2\lambda \mathbf{w_k}^t \mathbf{w_k})^{-1}\right] f_i + \frac{1}{2}\left[I + (I + 2\lambda \mathbf{w_k}^t \mathbf{w_k})^{-1}\right] f_{i+1}, \tag{7.55}$$

para $i = 8.k$ e $k = 1, 2, ..., 63$; caso contrário $\tilde{f}_i = f_i$.

[5] Isso será demonstrado no teorema 12.

Podemos reescrever as equações acima de forma a ficar explícita a natureza espaço-adaptativa do projetor P_w

$$\tilde{f}_i = \frac{f_i + f_{i+1}}{2} + (I + 2\lambda \mathbf{w_k}^t \mathbf{w_k})^{-1} \frac{f_i - f_{i+1}}{2}, \tag{7.56}$$

$$\tilde{f}_{i+1} = \frac{f_i + f_{i+1}}{2} - (I + 2\lambda \mathbf{w_k}^t \mathbf{w_k})^{-1} \frac{f_i - f_{i+1}}{2}. \tag{7.57}$$

A matriz $(I + 2\lambda \mathbf{w_k}^t \mathbf{w_k})^{-1}$ pode ser assim escrita

$$\begin{bmatrix} \frac{1}{1+2\lambda(w_1^k)^2} & 0 & 0 & \cdots & 0 \\ 0 & \frac{1}{1+2\lambda(w_2^k)^2} & 0 & \cdots & \vdots \\ 0 & 0 & \ddots & 0 & \vdots \\ \vdots & \vdots & 0 & \ddots & 0 \\ 0 & \cdots & \cdots & 0 & \frac{1}{1+2\lambda(w_{512}^k)^2} \end{bmatrix}. \tag{7.58}$$

De (7.57) e (7.58), concluímos que nas regiões onde os pesos w_i, isto é w_j^k, são maiores, a diferença entre os *pixels* vizinhos da imagem projetada é reduzida mais que em áreas onde os w_i são menores. No caso extremo em que $w_i = \infty$, os *pixels* vizinhos projetados são na verdade médias dos *pixels* vizinhos originais. Por outro lado, se $w_i = 0$ os *pixels* vizinhos permanecem inalterados. Portanto, sem dúvidas esse projetor é espaço-adaptativo.

Analogamente, é possível definir um conjunto C'_w para as linhas de blocos adjacentes e seu respectivo operador de projeção P'_w.

Porém, qual é o preço a ser pago por utilizar-se esses conjuntos e seus respectivos projetores ao invés dos conjuntos e projetores do Algoritmo 1? Resposta: em cada projeção é preciso calcular numericamente o parâmetro λ, resolvendo uma equação não-linear semelhante à equação (7.50).

7.3.1 O Parâmetro λ

Vamos examinar melhor a equação (7.50) e as suas eventuais raízes. Utilizando a notação já introduzida, temos

$$Df = x_0 - y_0; \quad \text{e} \quad D\tilde{f} = u - v. \tag{7.59}$$

Das equações (7.47) e (7.49),

$$u - v = (I + 2\lambda W^t W)^{-1}(x_0 - y_0), \tag{7.60}$$

ou ainda,

$$D\tilde{f} = (I + 2\lambda W^t W)^{-1} Df. \tag{7.61}$$

A equação (7.50) pode portanto ser escrita como

$$\| W(I + 2\lambda W^t W)^{-1} Df \| = E. \tag{7.62}$$

Assim, definindo-se

$$d = \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_{512\times 63} \end{bmatrix} = Df. \tag{7.63}$$

A equação (7.62) pode ser escrita como

$$\| W(I + 2\lambda W^t W)^{-1} d \| = E, \tag{7.64}$$

ou equivalentemente,

$$\sum_{i=1}^{512\times 63} \frac{w_i^2 d_i^2}{(1 + 2\lambda w_i^2)^2} = E^2. \tag{7.65}$$

Já mencionamos anteriormente que a raiz da equação acima correspondente ao operador de projeção P_w é positiva. Entretanto, é preciso perguntar primeiro se a equação (7.65) possui solução real positiva e além disso se é única e como obtê-la. O teorema abaixo responde às duas primeiras perguntas.

Teorema 12 *Para qualquer conjunto de* w_i, $i = 1, ..., 512 \times 63$, *reais, em (7.65), valem as seguintes afirmações :*

1. *a equação (7.65) possui uma única raiz positiva* λ;
2. *a equação (7.65) possui pelo menos uma raiz negativa;*
3. *as raizes negativas de (7.65) são todas maiores que a positiva em módulo;*
4. *a raiz positiva é a que corresponde ao operador* P_w.

Demonstração : Precisaremos do seguinte resultado: para quaisquer números reais $x < 0$, $y > 0$, e $\mid x \mid > y$, temos que

$$\left| \frac{x}{1+x} \right| > \frac{y}{1+y}. \tag{7.66}$$

Consideremos dois casos:

- $x < -1$. Temos que

$$\left|\frac{x}{1+x}\right| = \left|1 - \frac{1}{1+x}\right| = 1 + \left|\frac{1}{1+x}\right| > 1 > \frac{y}{1+y}. \tag{7.67}$$

- $x > -1$. Então, $0 < x + 1 < 1$. Logo,

$$\left|\frac{x}{1+x}\right| = \frac{|\,x\,|}{1+x} > |\,x\,| > y > \frac{y}{1+y}. \tag{7.68}$$

O que prova a desigualdade (7.66).

Seja

$$g(\lambda) = \sum_{i=1}^{n} \frac{w_i^2 d_i^2}{(1+2\lambda w_i^2)^2} - E^2. \tag{7.69}$$

Então,

$$g(0) = \| WDf \| - E \geq 0, \tag{7.70}$$

pois $f \notin C_w$.

Além disso,

$$\lim_{\lambda \to \infty} g(\lambda) = -E \leq 0. \tag{7.71}$$

Como a função $g(\lambda)$ é contínua para todo $\lambda \geq 0$, deve existir um $\lambda_+ \in (0, \infty)$ tal que $g(\lambda_+) = 0$. Notemos também que $g(\lambda)$ é estritamente decrescente para todo $\lambda \geq 0$. Assim, segue-se a unicidade da raiz positiva.

Seja

$$w_{min}^2 = \min\{w_i^2,\ i = 1, 2, ..., n\} \tag{7.72}$$

e seja

$$\lambda_{max} = -\frac{1}{2w_{min}^2}. \tag{7.73}$$

Então, $g(\lambda)$ é contínua para todo $\lambda \in (-\infty, \lambda_{max})$. Observemos que

$$\lim_{\lambda \to -\infty} g(\lambda) = -E \leq 0 \text{ e } \lim_{\lambda \to \lambda_{max}} g(\lambda) = +\infty. \tag{7.74}$$

Assim concluimos que deve existir um $\lambda_- \in (-\infty, \lambda_{max})$ de tal maneira que $g(\lambda_-) = 0$. Além disso, $g(\lambda)$ pode ter outras raízes negativas.

Para $\lambda \geq 0$, temos

$$(1 - 2\lambda w_i^2)^2 < (1 + 2\lambda w_i^2)^2, \tag{7.75}$$

para cada w_i. Logo, para $\lambda > 0$, $g(-\lambda) > g(\lambda)$. Seja λ_+ a raiz positiva de (7.65). Como $g(\lambda) > 0$ para todo $\lambda \in (0, \lambda_+)$, não existe nenhuma raiz para $g(\lambda)$ no intervalo $[-\lambda_+, 0)$. Portanto, todas as raízes negativas têm valor absoluto maior que a raiz positiva.

Das equações (7.47) e (7.49), vemos que a projeção $\tilde{f}$ de f em C_w é uma função de λ. Seja

$$D_{(\lambda)} = \| \tilde{f} - f \|^2 . \tag{7.76}$$

Ou ainda,

$$D_{(\lambda)} = \| u - x_0 \|^2 + \| v - y_0 \| . \tag{7.77}$$

Sabemos que

$$u + v = x_0 + y_0 \quad \text{e} \quad u - v = (I + 2\lambda W^t W)^{-1}(x_0 - y_0), \tag{7.78}$$

logo

$$u - x_0 = y_0 - v. \tag{7.79}$$

Assim a equação (7.77) pode ser escrita como

$$D_{(\lambda)} = 2 \| u - x_0 \|^2 \tag{7.80}$$

$$= 2 \| \frac{1}{2}[(u - x_0) + (u - x_0)] \|^2 \tag{7.81}$$

$$= \frac{1}{2} \| (u - x_0) + (y_0 - v) \|^2 \tag{7.82}$$

$$= \frac{1}{2} \| (u - v) + (x_0 - y_0) \|^2 \tag{7.83}$$

$$= \frac{1}{2} \| [(I + 2\lambda W^t W)^{-1} - I](x_0 - y_0) \|^2 \tag{7.84}$$

$$= \frac{1}{2} \| [(I + 2\lambda W^t W)^{-1} - I] Df \|^2 . \tag{7.85}$$

Temos, portanto, que

$$D_{(\lambda)} = \frac{1}{2} \| [(I + 2\lambda W^t W)^{-1} - I] d \|^2 = \sum_{i=1}^{n} \left(\frac{2\lambda w_i^2}{1 + 2\lambda w_i^2} \right)^2 d_i^2. \tag{7.86}$$

Denotemos por λ_- uma raiz negativa e λ_+ a raiz positiva. Então, sabemos que $| \lambda_- | > \lambda_+$. Logo, utilizando a desigualdade (7.66), obtemos

$$\left(\frac{2\lambda_- w_i^2}{1 + 2\lambda_- w_i^2} \right)^2 > \left(\frac{2\lambda_+ w_i^2}{1 + 2\lambda_+ w_i^2} \right)^2 , \tag{7.87}$$

para cada i. Concluimos que

$$D_{(\lambda_-)} > D_{(\lambda_+)}. \tag{7.88}$$

Como λ_- foi escolhido de forma arbitrária, está demonstrado o teorema. □

Para responder como obter a raiz positiva da equação (7.65), precisaremos do teorema a seguir.

Teorema 13 *Seja*

$$g(\lambda) = \sum_{i=1}^{n} \frac{w_i^2 d_i^2}{(1 + 2\lambda w_i^2)^2} - E^2. \tag{7.89}$$

Então as iterações do método de Newton

$$\lambda_{k+1} = \lambda_k - \frac{g(\lambda_k)}{g'(\lambda_k)}, \quad k = 0, 1, 2, ... \tag{7.90}$$

com $\lambda_0 = 0$ convergem sempre para a raiz positiva de (7.65). Além disso, $\lambda_{k+1} > \lambda_k$ e $| \lambda_{k+1} - \lambda^* | < | \lambda_k - \lambda^* |$, onde λ^* é a solução exata.

Demonstração : Derivando a equação (7.89), obtemos

$$g'(\lambda) = -4 \sum_{i=1}^{n} \frac{w_i^4 d_i^2}{(1 + 2\lambda w_i^2)^3}. \tag{7.91}$$

$g'(\lambda)$ é uma função contínua para $\lambda \geq 0$, crescente e $g'(\lambda) < 0$ para todo $\lambda \geq 0$. Se $\lambda_k \geq 0$ é tal que $g(\lambda_k) > 0$, então pela iteração de Newton

$$\lambda_{k+1} = \lambda_k - \frac{g(\lambda_k)}{g'(\lambda_k)}, \tag{7.92}$$

concluimos que $\lambda_{k+1} > \lambda_k$. Por outro lado,

$$g(\lambda_{k+1}) = \int_{\lambda_k}^{\lambda_{k+1}} g'(\lambda) d\lambda + g(\lambda_k). \tag{7.93}$$

Para $\lambda \in (\lambda_k, \lambda_{k+1}]$, $g'(\lambda) > g'(\lambda_k)$, e daí,

$$g(\lambda_{k+1}) > \int_{la_k}^{\lambda_{k+1}} g'(\lambda_k) d\lambda + g(\lambda_k) = g'(\lambda_k)(\lambda_{k+1} - \lambda_k) + g(\lambda_k) = 0. \tag{7.94}$$

Logo, $g(\lambda_{k+1}) > 0$ também, e portanto, para $\lambda_0 = 0$, a iteração de Newton

$$\lambda_{k+1} = \lambda_k - \frac{g(\lambda_k)}{g'(\lambda_k)}, \quad k = 0, 1, 2, ... \tag{7.95}$$

gerará a sequência

$$0 = \lambda_0 < \lambda_1 < \lambda_2 < \cdots \tag{7.96}$$

que converge para a raiz positiva λ_+, pois $g(\lambda_+) = 0 < g(\lambda_k)$ garante que $\lambda_k < \lambda_+$ para $k = 0, 1, 2, \ldots$ □

Assim, para se obter a raiz positiva, basta usar o método de Newton com ponto inicial $\lambda_0 = 0$. O teorema acima garante a convergência.

Talvez uma forma mais elegante de obter os principais resultados dos teoremas acima fosse a seguinte: a unicidade e existência da raiz positiva da equação (7.65) é uma conseguência da existência da projeção sobre conjunto convexo fechado; a convergêcia do método de Newton com ponto inicial $x_0 = 0$ pode ser obtida utilizando a continuidade e a convexidade da função $g(\lambda)$ para valores positivos de λ . Observe que $g(0) > 0$ e que $g'(\lambda) < 0$ para valores positivos de λ, ou seja, $g(\lambda)$ é decrescente para $\lambda > 0$. Além disso, pelo teste da derivada segunda, $g(\lambda)$ tem concavidade para cima, ou seja, é convexa para $\lambda > 0$. Essas informações são suficientes para demonstrar os resultados.

7.3.2 Os elementos da Matriz W

A escolha dos pesos w_i da matriz W deve ser baseada em caracteríticas estatísticas locais da imagem e nas propriedades da percepção visual humana. Esta é uma tarefa delicada, e, geralmente, é realizada com ajuda de modelos que tentem melhor satisfazer as condições acima citadas.

Assim, o brilho (nível de cinza) de um pixel na posição (i, j) pode ser encarada como uma variável aleatória com média $\mu_{i,j}$ e variância $\sigma^2_{i,j}$. Localmente falando, a média pode servir como uma medida do brilho e a variância como uma estimativa para o grau de detalhes em volta do pixel na posição (i, j). Como já mencionamos que em regiões com muitos detalhes os artefatos entre os blocos são menos visíveis, então os pesos w_i têm que ser funções decrescentes de $\sigma_{i,j}$. Por exemplo,

$$w_{i,j} = \frac{1}{1 + \sigma_{i,j}}, \tag{7.97}$$

onde o número 1 no denominador foi acrescentado para eliminar problemas nos casos em que $\sigma_{i,j} = 0$. É claro que há um alto grau de liberdade na escolha da função $w_{i,j}$. A função em questão não pode crescer muito rápido por questões computacionais, portanto uma forma mais conveniente seria

$$w_{i,j} = ln\left(1 + \frac{1}{1 + \sigma_{i,j}}\right). \tag{7.98}$$

Uma outra observação importante é que os artefatos entre os blocos são mais visíveis em regiões claras que em regiões escuras. A função sugerida como um modelo adequado para estas observações no paper [2] é

$$w_{i,j} = ln\left(1 + \frac{\sqrt{\mu_{i,j}}}{1 + \sigma_{i,j}}\right). \tag{7.99}$$

E esta será a função que utilizaremos em nossos experimentos. Há muitas outras funções como esta na literatura. Elas são chamadas de *VFBA*, ou seja, *Visibility Functions of the Blocking Artifact*.

7.3.3 Algoritmo 2

Além dos conjuntos C_w, C'_w, C_T e seus respectivos operadores de projeção , ainda podemos definir o conjunto abaixo

$$C_p \triangleq \{f : 0 \leq f_{i,j} \leq 255, \quad 1 \leq i,j \leq N\}, \tag{7.100}$$

cujo operador de projeção é semelhante ao operador P_T.

De posse desses conjuntos, o algoritmo 2 tem a seguinte estrutura básica:

1. faça $f_0 = f'$;
2. para $k = 1, 2, ...$ calcule iterativamente f_k da seguinte forma

$$f_k = P_w P'_w P_T P_p f_{k-1}$$

3. continue até que $\| f_k - f_{k-1} \| < TOL$, onde TOL é um limitante pré-estabelecido.

Observemos que para fazer as projeções sobre C_w e C'_w precisamos resolver uma equação não linear para obter o parâmetro λ.

Comparando-se com o Algoritmo 1, a diferença fundamental está nos operadores P_w e P'_w. Veremos nos experimentos que esse último algoritmo é superior ao primeiro.

Capítulo 8

Experimentos

O objetivo deste capítulo é descrever os experimentos realizados com os dois algoritmos descritos anteriormente. Discutiremos aspectos relacionados a implementação computacional e,na medida do possível, deixaremos indicado material bibliográfico para eventual consulta assim como endereços na Internet para obtenção de softwares utilizados.

8.1 O Pacote Jpeg e o Viewer

Para comprimir as imagens empregando várias tabelas de quantização , utilizamos um pacote disponível no seguinte endereço: **ftp://ftp.uu.net** no diretório *graphics/jpeg*. Esta versão do Jpeg está bem documentada, isto é, traz documentos explicando as estruturas de dados utilizadas, as convenções de programação e um pequeno manual para a instalação e utilização do software. O código fonte está escrito em linguagem *C* e é portátil, ou seja, mudando-se apenas o *Makefile* é possível rodar o programa tanto na *SUN* como num IBM-PC. Vale mencionar que neste endereço sempre encontramos uma versão atualizada periodicamente.

Para visualização e impressão das imagens na estação de trabalho *SUN*, usamos o **xv**; já no IBM-PC Pentium rodando sob o Windows 95, o melhor viewer é o **LView Pro** que pode ser obtido nos seguintes endereços:

- **ftp://ftp.lsi.usp.br/pub/www/**
- **ftp://oak.oakland.edu/SimTel/Win3/graphics/**
- **ftp://ftp.ncsa.uiuc.edu/Mosaic/Windows/Viewers/**

8.2 Formato das Imagens

Tanto o **xv** quanto o **LView Pro** são compatíveis com o formato Jpeg. Entretanto, o formato Jpeg não é o mais adequado para a manipulação computacional das imagens. Pois, para efetuarmos, por exemplo, a transformada cosseno precisamos da imagem em sua forma matricial. Por isso, preferimos converter as imagens para os formatos *PNM* ou *PGM*. Estes são os mais simples de todos porque apenas armazenam a imagem justamente em sua forma matricial. Para uma descrição detalhada desses formatos, veja o livro [20].

8.3 Implementação do Algoritmo I

No primeiro algoritmo, foi necessário apenas definir uma estimativa para a largura do conjunto C_c, veja equação (7.14), ou seja, o número E_c. Relembrando a definição (7.33), consideremos a imagem com os artefatos entre os blocos $f' = \{f'_1, f'_2, ..., f'_{512}\}$, definamos S_k como

$$S_k \triangleq \left[\sum_{i=1}^{63} \| f'_{8i+k} - f'_{8i+k+1} \|^2\right]^{1/2}, \tag{8.1}$$

e, por fim, tomemos E_c como uma média dos S_k, ou seja,

$$E_c = \frac{1}{7}\sum_{k=1}^{7} S_k. \tag{8.2}$$

Notemos que E_c é uma média das descontinuidades entre colunas consecutivas de toda a imagem. Tal estimativa nos forneceu bons resultados. Fizemos escolhas diferentes para E_c, especialmente, no sentido de diminuir o seu valor, porém tais experimentos demonstraram que não é bom exigir muita suavidade das colunas (linhas) entre blocos adjacentes.

Determinamos C_l de forma completamente análoga ao que foi acima exposto.

8.4 Implementação do Algoritmo II

Para a implementação do segundo algoritmo, precisamos definir várias coisas deixadas em aberto na sua descrição teórica. Seguindo as convenções do paper [2], procuraremos, a seguir, enumerar as principais definições .

8.4.1 Pesos da Matriz W

Como definir a média e a variância para calcular os pesos $w_{i,j}$? Uma primeira simplificação é ao invés de tomarmos a média e variância para cada pixel, consideramos a média e a variância para cada bloco 8×8.

Lembrando-se que o coeficiente dc [1] é uma média a menos de uma constante multiplicativa, podemos utilizá-lo para estimar a média em cada bloco.

O procedimento para calcular a média é o seguinte. Consideremos a fronteira entre duas colunas de blocos vizinhos e chamemo-la de ℓ, e sejam DC_e e DC_d os coeficientes dc dos blocos à esquerda e à direta da fronteira ℓ, respectivamente. A estimativa da média μ_{c_ℓ} usada nos cálculos dos pesos para a matriz W é dada por

$$\hat{\mu}_{c_\ell} = \frac{DC_e + DC_d}{2 \times 8^2}. \tag{8.3}$$

Uma equação similar é utilizada para calcular μ_{l_ℓ} que é a média entre linhas de blocos consecutivos.

Para obtermos uma estimativa para a variância entre colunas de blocos adjacentes, definimos VAC_e e VAC_d que são as variâncias obtidas a partir da soma dos coeficientes ac[2] elevados ao quadrado. A variância σ_ℓ^2 é dada por

$$\hat{\sigma}_{c_\ell}^2 = \frac{VAC_e + VAC_d}{2 \times 8^2}. \tag{8.4}$$

Analogamente para a variância, $\sigma_{l_\ell}^2$ entre as linhas de blocos vizinhos.

Os pesos são calculados utilizando as médias, variâncias acima descritas e a $VFBA$ da equação (7.99).

8.4.2 Estimativas para E_c e E_l

Para determinar uma estimativa para E_c, tomamos, mais uma vez, a imagem com artefatos entre os blocos f' em sua forma $f' = \{f'_1, ..., f'_{512}\}$ e definimos S_k como segue

$$S_k \triangleq \left[\sum_{i=1}^{63} \| \mathbf{w_k}(f'_{8i+k} - f'_{8i+k+1} \|^2 \right]^2, \tag{8.5}$$

onde $\mathbf{w_k}$ é definida em (7.52).

Seguindo o raciocínio do Algoritmo I, definimos E_c como uma média de S_k, isto é,

[1] O coeficiente dc é o primeiro coeficiente transformado, isto é, aquele de índices $(0, 0)$.

[2] São todos os outros coeficientes, exceto o dc.

$$E_c = \frac{1}{7}\sum_{k=1}^{7} S_k. \tag{8.6}$$

Repetindo os mesmo passos para as linhas, obtemos E_l.

8.5 Uma Medida de Distância entre Imagens

Suponhamos que tivessemos duas imagens reconstruidas com métodos distintos, e além disso dispusessemos da imagem original para eventual avaliação visual dos métodos a partir da comparação direta. A avaliação da qualidade de reconstrução pode ser efetuada visualmente, porém carece de certa objetividade. Por isso a necessidade de utilizarmos uma métrica para "medir" a distância entre uma imagem reconstruida e a imagem original. Assim quanto mais "próxima" a imagem reconstruida estivesse da imagem original tanto melhor seria determinado método.

Uma métrica muito utilizada é a **PSNR**[3]. Sejam g e f as imagens $(N \times N)$ reconstruida e original, respectivamente. Se estivermos trabalhando com 256 níveis de cinza, a medida PSNR, em dB, é assim definida

$$PSNR = 10\log_{10}\left[\frac{N^2 \times 255^2}{\| g - f \|^2}\right]. \tag{8.7}$$

Pela própria definição , observamos que quanto mais "próximas" estiverem as imagens tanto maior será o valor medido em dB. Um ganho de, por exemplo, $0.9dB$ numa recuperação significa muito em termos de qualidade da imagem como veremos nos experimentos.

8.6 Tabelas de Quantização

Ambos os papers [1] e [2], utilizam, para exemplificação , a seguinte tabela de quantização , chamada *tabela.tes*, que apresenta uma *bit-rate* de $0.24bpp$[4]:

[3]Peak-Signal-to-Noise-Ratio.

[4]Bits Per Pixels.

$$tabela.tes = \begin{pmatrix} 50 & 60 & 70 & 70 & 90 & 120 & 255 & 255 \\ 60 & 60 & 70 & 96 & 130 & 255 & 255 & 255 \\ 70 & 70 & 80 & 120 & 200 & 255 & 255 & 255 \\ 70 & 96 & 120 & 145 & 255 & 255 & 255 & 255 \\ 90 & 130 & 200 & 255 & 255 & 255 & 255 & 255 \\ 120 & 255 & 255 & 255 & 255 & 255 & 255 & 255 \\ 255 & 255 & 255 & 255 & 255 & 255 & 255 & 255 \\ 255 & 255 & 255 & 255 & 255 & 255 & 255 & 255 \end{pmatrix}. \tag{8.8}$$

Entretanto, nesta dissertação , para deixar bem clara a origem dos artefatos entre os blocos e a sua relação com os coeficientes transformados, tomamos mão de mais duas tabelas de quantização . A primeira, chamada *tabela.baixa*, que realiza uma pequena quantização (passo de quantização iqual a 1) nos coeficientes de baixa frequência e quantiza rigorosamente o restante dos coeficientes; e uma outra, denotada por *tabela.alta*, que faz exatamente o inverso. A idéia é verificar em qual caso encontraremos os artefatos entre os blocos, isto é, se os responsáveis pelos artefatos entre blocos são os coeficientes de baixas ou de altas frequências. Você tem alguma intuição ?

$$tabela.baixa = \begin{pmatrix} 1 & 1 & 1 & 1 & 255 & 255 & 255 & 255 \\ 1 & 1 & 1 & 1 & 255 & 255 & 255 & 255 \\ 1 & 1 & 1 & 255 & 255 & 255 & 255 & 255 \\ 1 & 1 & 255 & 255 & 255 & 255 & 255 & 255 \\ 255 & 255 & 255 & 255 & 255 & 255 & 255 & 255 \\ 255 & 255 & 255 & 255 & 255 & 255 & 255 & 255 \\ 255 & 255 & 255 & 255 & 255 & 255 & 255 & 255 \\ 255 & 255 & 255 & 255 & 255 & 255 & 255 & 255 \end{pmatrix}. \tag{8.9}$$

$$tabela.alta = \begin{pmatrix} 50 & 60 & 70 & 1 & 1 & 1 & 1 & 1 \\ 60 & 60 & 70 & 1 & 1 & 1 & 1 & 1 \\ 70 & 70 & 80 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}. \tag{8.10}$$

Realizamos muitos outros testes com ambos algoritmos, e, consequentemente, empregamos igual número de tabelas de quantização . Para deixar bem documentados, pelo menos os experimentos mencionados nesta dissertação , apresentamos a seguir duas outras tabelas de quantização , com taxas de compressão 0.43 *bpp* e 0.19 *bpp*, respectivamente.

$$tabela2.tes = \begin{pmatrix} 35 & 30 & 30 & 16 & 24 & 40 & 255 & 255 \\ 32 & 32 & 34 & 19 & 26 & 255 & 255 & 255 \\ 34 & 33 & 36 & 24 & 200 & 255 & 255 & 255 \\ 14 & 17 & 22 & 29 & 255 & 255 & 255 & 255 \\ 18 & 22 & 37 & 56 & 255 & 255 & 255 & 255 \\ 24 & 35 & 55 & 64 & 255 & 255 & 255 & 255 \\ 255 & 255 & 255 & 255 & 255 & 255 & 255 & 255 \\ 255 & 255 & 255 & 255 & 255 & 255 & 255 & 255 \end{pmatrix}. \tag{8.11}$$

$$tabela3.tes = \begin{pmatrix} 90 & 100 & 95 & 110 & 120 & 130 & 255 & 255 \\ 110 & 110 & 120 & 116 & 120 & 255 & 255 & 255 \\ 110 & 110 & 95 & 120 & 200 & 255 & 255 & 255 \\ 150 & 150 & 155 & 155 & 255 & 255 & 255 & 255 \\ 110 & 130 & 200 & 255 & 255 & 255 & 255 & 255 \\ 120 & 255 & 255 & 255 & 255 & 255 & 255 & 255 \\ 255 & 255 & 255 & 255 & 255 & 255 & 255 & 255 \\ 255 & 255 & 255 & 255 & 255 & 255 & 255 & 255 \end{pmatrix}. \tag{8.12}$$

8.7 Resultados

Como uma primeira observação relevante, vamos a seguir apresentar os resultados das compressões utilizando as tabelas *tabela.baixa* e *tabela.alta*, respectivamente, a fim de constatar a "origem" dos artefatos entre blocos numa imagem comprimida pelo *Jpeg*.

Observando a imagem *Lena.baixa*, percebemos que, sem afetar muito os coeficientes de baixa frequência, os artefatos entre blocos são quase imperceptivos. Não há muita perda da qualidade da imagem e nem muita compressão, como se era de esperar. O ponto importante é: não temos os efeitos catastróficos entre os blocos adjacentes.

Por outro lado, utilizando a *tabela.alta*, a imagem torna-se nitidamente quadriculada. Determinando assim o papel crucial dos coeficientes de baixa frequência no surgimento dos artefatos entre blocos. Veja a imagem *Lena.alta*.

Portanto fica respondida a pergunta feita no início desta seção . Podemos compreender a origem dos artefatos entre blocos como uma conjunção de dois fatores basicamente:

- o processamento separado de cada bloco 8×8;
- uma quantização "rigorosa" dos coeficientes de baixa frequência, principalmente.

O primeiro fator é evidente, pois na segmentação da imagem em blocos e posterior transformação -quantização não é levada em consideração a relação entre os blocos da imagem, em particular, entre os blocos adjacentes. Isso pode ser constatado facilmente se

Figura 8.1: Imagem quantizada com a tabela.baixa

Figura 8.2: Imagem quantizada com a tabela_alta

ao invés de dividirmos a imagem em blocos, tomássemos ela como um todo e efetuássemos a transformação -quantização (com uma alta taxa de compressão). Evidentemente, não teríamos o efeito "quadriculado" na imagem (outros artefatos aparecem na imagem).

Contudo, somente o processamento separado de cada bloco não é suficiente para o surgimento dos artefatos entre blocos. Considere, por exemplo, uma tabela de quantização toda formada pelo número 1. Tal tabela não afeta muito (passo de quantização é pequeno) os coeficientes transformados, e por consequência, não afetará significativamente a qualidade da imagem (na verdade, é impossível, visualmente falando, distinguir a imagem original da imagem comprimida nesse exemplo). Obviamente esse é um caso muitíssimo particular; entretanto, é possível obter taxas de compressão relativamente altas sem afetar significativamente a qualidade da imagem (se isso não fosse possível, não haveria justificativa para se falar em compressão). E, geralmente, essas tabelas tentam não perder muita informação ao quantizar os coeficientes de baixa frequência. Essa última afirmação é a explicação para a importância desses coeficientes. Lembremos que a transformada cosseno tende a concentrar o máximo de energia (informação) possível no mínimo de coeficientes. E, os coeficientes que mais têm informações são aqueles mais próximos do coeficiente *dc*, isto é, os coeficientes de baixa frequência.

Para uma verificação objetiva da qualidade da imagem, as *PSNR* das imagens *Lena.baixa* e *Lena.alta* são 31.7784 *dB* e 27.1724 *dB*, respectivamente.

Feita essa consideração acerca da origem dos artefatos entre os blocos, podemos passar para os resultados propriamente ditos. Apresentamos a seguir uma tabela contendo os resultados de nossos experimentos utilizando 3 tabelas de quantização e os dois algoritmos descritos no capítulo precedente. Afim de chamar a atenção para a principal diferença dos dois algoritmos, chamaremos o algoritmo I de *Algoritmo Não-Adaptativo* e o algoritmo II de *Algoritmo Adaptativo*.

Algoritmo de Reconstrução	**Taxa de Compressão**		
	0.43 bpp	0.24 bpp	0.19 bpp
Descodificador JPEG	34.5120 dB	31.1645 dB	29.2326 dB
Algoritmo Não-Adaptativo	34.8794 dB	31.8976 dB	30.1205 dB
Algoritmo Adaptativo	34.9097 dB	31.9047 dB	30.1261 dB

Tabela de Métodos

A primeira coluna da tabela nos diz os métodos utilizados para recuperar a imagem. Veja que a primeira linha diz respeito as imagens reconstruidas a partir do pacote *JPEG*, isto é, utilizando simplesmente o descodificador padrão recomendado. As taxas de compressão indicam, na verdade, as tabelas de quantização empregadas em cada experimento. Assim, olhando a coluna referente a taxa de compressão 0.43 *bpp*, encontramos as distâncias das imagens recuperadas a imagem original, medidas em *dB*. Por exemplo, ainda na mesma coluna, verificamos que a imagem recuperada pelo *JPEG* está 34.5120 *dB* da imagem original. Por outro lado, se utilizarmos o algoritmo não-adaptativo, obtemos

uma imagem mais próxima da original, 34.8794 dB. Esse resultado ainda pode ser melhorado para 34.9097 dB, se o algoritmo adaptativo for empregado.

Observando-se a tabela globalmente, verificamos que o algoritmo não-adaptativo é melhor que o descodificador *Jpeg*. Por outro lado, o algoritmo adaptativo é melhor que o não-adaptativo. O termo "melhor", neste contexo, significa que o algoritmo em questão apresenta uma imagem reconstruida mais próxima, no sentido da *PSNR*, da imagem original, além de apresentar imagens de melhor qualidade (veja as imagens da *Lena*), visualmente falando.

Estes resultados são coerentes com os demais experimentos realizados no decorrer desta dissertação . Além da imagem *Lena*, trabalhamos com outras imagens que estão no apêndice. Entre as mais conhecidas, a imagem do macaco *Baboon* e o *Navio*, etc.

Figura 8.3: Imagem original

Figura 8.4: Imagem quantizada com a tabela.tes

Figura 8.5: Imagem recuperada com o Algoritmo I

Figura 8.6: Imagem recuperada com o Algoritmo II

Capítulo 9

Conclusões e Novas Perspectivas

Este último capítulo pretende ser um apanhado geral das conclusões que chegamos ao estudar os algoritmos apresentados e uma pequena indicação dos novos rumos que o campo da *Compressão de Imagens* vem tomando mais recentemente.

9.1 Filosofia do Método de Projeções sobre Conjuntos Convexos

A abordagem do método das projeções ortogonais sobre conjuntos convexos é limitada no seguinte sentido: não ataca diretamente a raiz do problema que é a quantização dos coeficientes transformados de baixa frequência. Isto significa que, apesar de atenuados, os artefatos entre blocos estarão sempre presentes na imagem recuperada.

Para ser mais preciso, a única possibilidade de *indiretamente* recuperar os coeficientes de baixa frequência são as projeções sobre os conjuntos C_w e C'_w seguidas da projeção sobre C_T para eventual correção dos dados. Digo "indiretamente", porque o objetivo expresso é suavizar a descontinuidade entre as colunas (ou linhas)de blocos vizinhos, levando-se a uma falsa impressão que é somente entre estas colunas (ou linhas) que há a "descontinuidade". Obviamente a descontinuidade está presente no bloco como um todo. Para ilustrar, imagine um bloco de uma imagem todo branco e ao seu lado um bloco preto. Suavizar a "descontinuidade" entre eles é somente tornar cinza as colunas da fronteira? Este, é claro, representa um caso extremo,porém é bem ilustrativo.

Os autores dos papers [1] e [2] sugerem definir outros conjuntos para suavizar as colunas (ou linhas) interiores aos blocos. Entretanto, tal procedimento não leva em conta explicitamente a relação do bloco em questão com os blocos vizinhos, e esta relação é outro aspecto fundamental no surgimento dos artefatos entre os blocos.

Essas são em nosso entender as fragilidades da filosofia do Método das Projeções

sobre Conjuntos Convexos, e elas indicam uma nova abordagem ao problema: tentar construir métodos capazes de alguma forma recuperar os coeficientes transformados de baixa frequência, levando-se em conta a interrelação entre os blocos. E isto já vem sendo pesquisado, veja a homepage do pesquisador Kenji Tanaka, **http://rainbow. ece. ucsb.edu /alumni/kenji/acad.html**, orientando do Prof. Allen Gersho **http://www. ece. ucsb. edu /Faculty /Gersho /default.html**.

No entanto, são indiscutíveis os bons resultados do método apresentado nesta dissertação . Basta para isso, comparar as imagens antes e depois de aplicado o algoritmo.

9.2 Novos Rumos

De 16 a 20 de setembro de 1996, em Goiânia, ao participarmos do *XIX Congresso Nacional de Matemática Aplicada e Computacional* tivemos a oportunidade de trocar informações com dois pesquisadores que se interessam pelo assunto "Compressão de Imagens".

O primeiro, Prof. Gilbert Strang, indicou claramente a relevância das pesquisas utilizando *Wavelets* e *Filter Banks* no processamento de sinais, e, em especial, na compressão de imagens. Com uma base Wavelet adequada, excelentes resultados têm sido obtidos em compressão de imagens. Em breve, o grupo *Jpeg* terá que rever seu método padrão de compressão, se quiser continuar sendo fiel ao seu objetivo : "*be at or near the state of the art with regard to compression rate and accompanying image fidelity, over a wide range of image quality ratings*", veja página 31 do paper [7]. Para mais informações sobre Wavelets e Filter Banks veja o livro [24] dos Professores Strang e Truong na bibliografia.

Um "approach" totalmente diferente para tratar os "Blocking Artifacts" foi apresentado pelo Prof. Henrique S. Malvar da Picturetel Corp. Ele usou uma transformada nova chamada de "Lapped Transform", ou de forma geral, "Lapped Orthogonal Transform" (LOT). Apresentou alguns resultados que impressionaram pela qualidade da imagem recuperada. Praticamente eliminou os artefatos entre os blocos. Vale mencionar que a Lapped Transform possibilita recuperar a inter-relação entre os blocos adjacentes e, consequentemente, uma melhor "aproximação " dos coeficientes transformados de baixa frequência. O método proposto por ele é análogo ao sugerido pelo *Jpeg*, exceto a transformada, é claro. Para maiores detalhes sobre a "Lapped Transform", veja o livro [25].

Para quem quiser ter uma idéia dos resultados obtidos com Wavelets e a Lapped Transform, há disponível na Internet um programa escrito em *Matlab* chamado *UICODER* que faz compressão com esses novos métodos. O *UICODER* foi desenvolvido pelo grupo do Prof. Truong Nguyen. O endereço é: **ftp://eceserv0.ece.wisc.edu/pub/nguyen/SOFTWARE/UICODER/**.

Sem mais, foi um prazer estudar compressão de imagens...

Apêndice A

O objetivo deste apêndice é dar uma idéia de um tipo de codificação simbólica: a codificação de *Huffman*.

A codificação simbólica tem por objetivo representar com o menor número de bits possíveis os dados mais frequentes e os dados com menos ocorrências com um número de bits relativamente maior. Essa é a idéia para obter compressão no momento da codificação

.

O método de *Huffman*, criado em 1952, pode assim ser exemplificado. Considere um conjunto de dados $q_1, q_2, ..., q_8$ cujas probabilidades de ocorrências são as seguintes:

$$\begin{array}{ll} p_1 = p\{q_1\} = 0.4 & p_5 = p\{q_5\} = 0.12 \\ p_2 = p\{q_2\} = 0.08 & p_6 = p\{q_6\} = 0.08 \\ p_3 = p\{q_3\} = 0.08 & p_7 = p\{q_7\} = 0.04 \\ p_4 = p\{q_4\} = 0.2 & p_8 = p\{q_8\} = 0 \end{array}$$

onde p_i é a probabilidade de ocorrência do dado q_i. Na prática, em compressão de imagens, as probabilidades são calculadas com a ajuda do *Histograma* [1].

Realizamos os seguintes passos:

1. procuramos na lista de probabilidades pelos dois dados com probabilidades menores. Em nosso exemplo, os dados p_7 e p_8 com probabilidades 0.04 e 0, respectivamente.

2. somamos suas probabilidades. Formamos assim um nova lista composta das probabilidades não afetadas no passo 1 e da soma gerada no passo 1. A nova lista tem um elemento a menos.

3. retorne ao passo 1 e repita os procedimentos até sobrar apenas um elemento na lista.

Para auxiliar a compreensão do algoritmo, faremos uso de dois gráficos.

Veja que no passo 2, só temos 7 itens, e, assim por diante, até o passo 7, restando apenas um item.

[1] O histograma de uma imagem representa a frequência de ocorrência dos vários níveis de cinza nessa imagem.

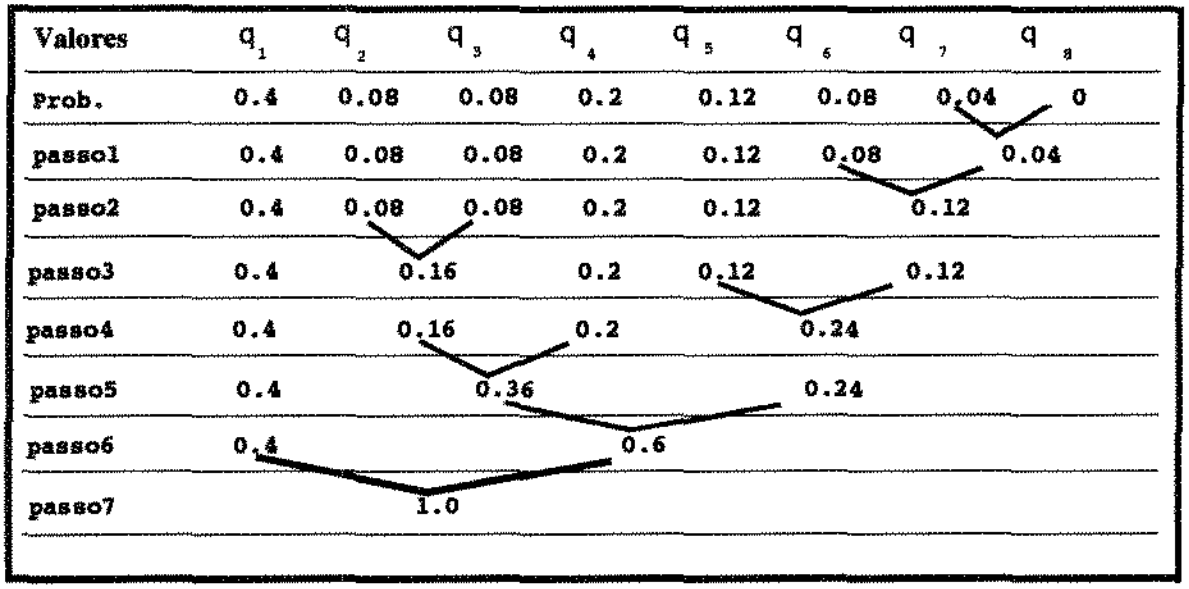


Figura 9.1: Esquema da Codificação de Huffman

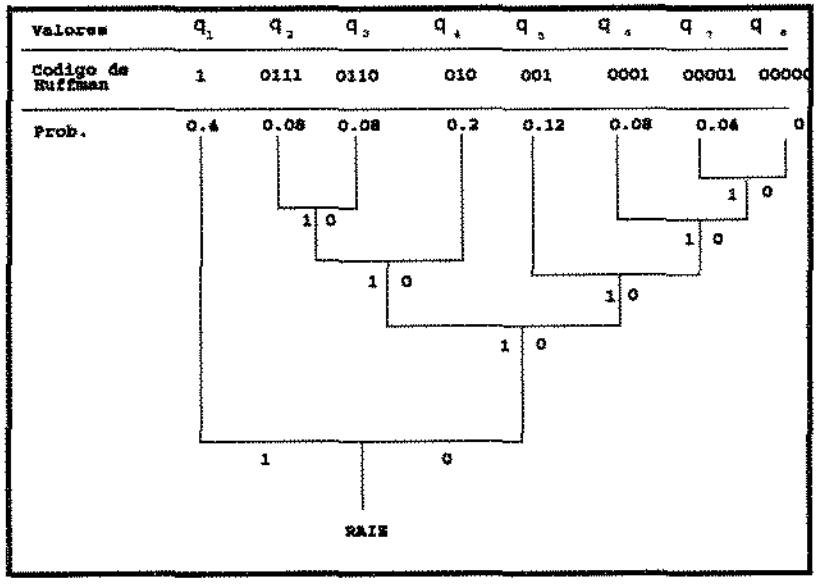


Figura 9.2: Codificação de Huffman em forma de árvore

Neste último gráfico, fica mais clara a forma de codificação . Os códigos são obtidos começando na **RAIZ** e associando-se o número 1 para um ramo e 0 para o outro. Assim, para obter o código de um elemento basta percorrer a árvore da raiz até o elemento, formando-se um código binário.

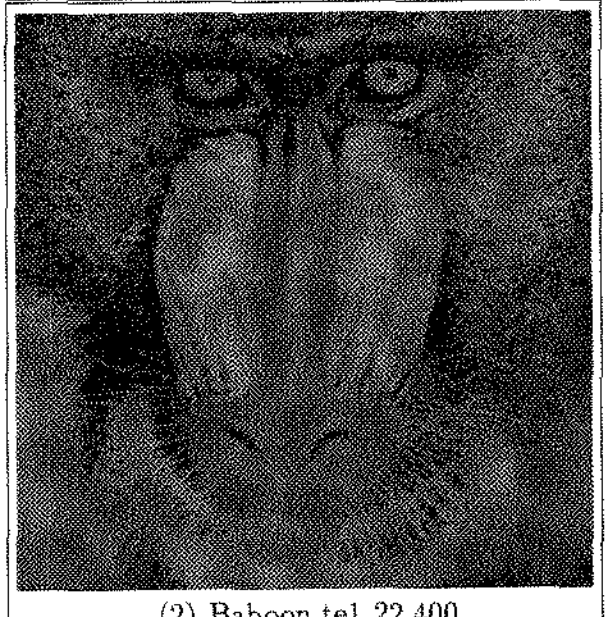
Note que os elementos com maior probabilidade de ocorrência possuem um código com menos bits, por exemplo, o elemento q_1 que tem a maior probabilidade, 0.4, será codificado utilizando-se apenas um bit = 1. Por outro lado, aqueles que têm probabilidade baixa serão codificados utilizando-se um maior número de bits, por exemplo o elemento q_7 será representado por 00001, isto é, 5 bits.

Apêndice B

Nas próximas páginas,segue-se uma lista de algumas imagens empregadas nos experimentos dessa tese. O número ao lado do nome da imagem é o *PSNR* em relação à imagem original. A tabela de quantização utilizada é a *tabela.tes* que nos fornece uma taxa de compressão da ordem de 0.24 *bpp*.

Omitimos a imagem original, apresentando apenas as imagens "teste" e as imagens "recuperadas" pelo algoritmo II, isto é, pelo algoritmo adaptativo.

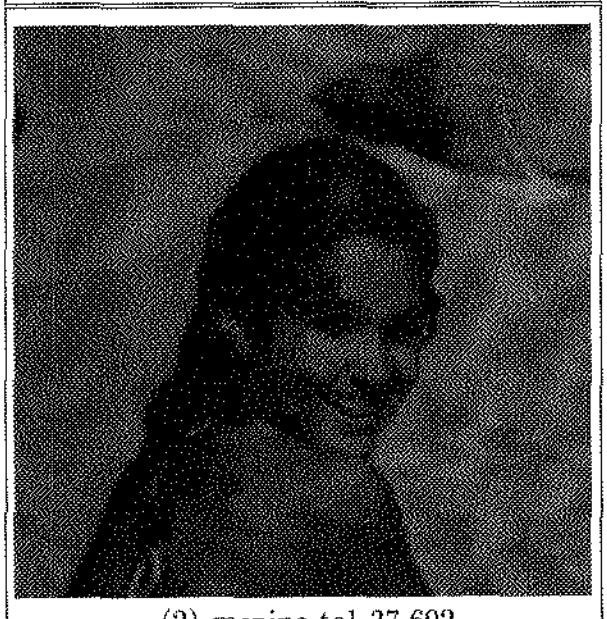
(1) Baboon.tes 22.061



(2) Baboon.te1 22.400

Figura 9.3: Imagens do Macaco

(1) menina.tes 37.130



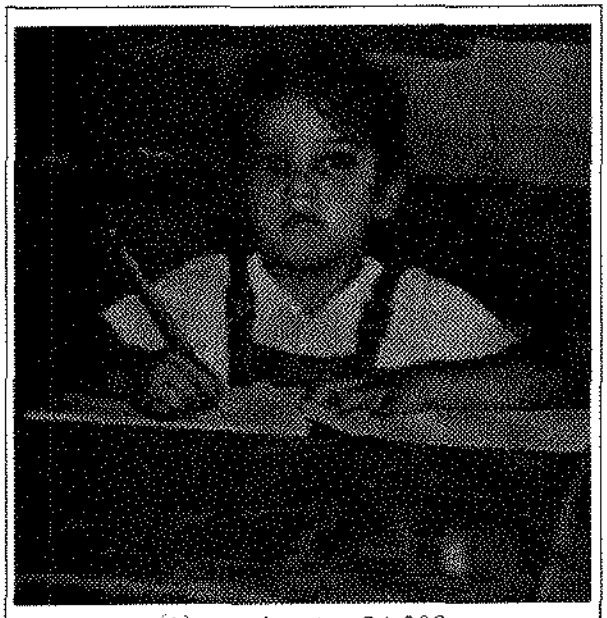
(2) menina.tel 37.692

Figura 9.4: Imagens da Menina

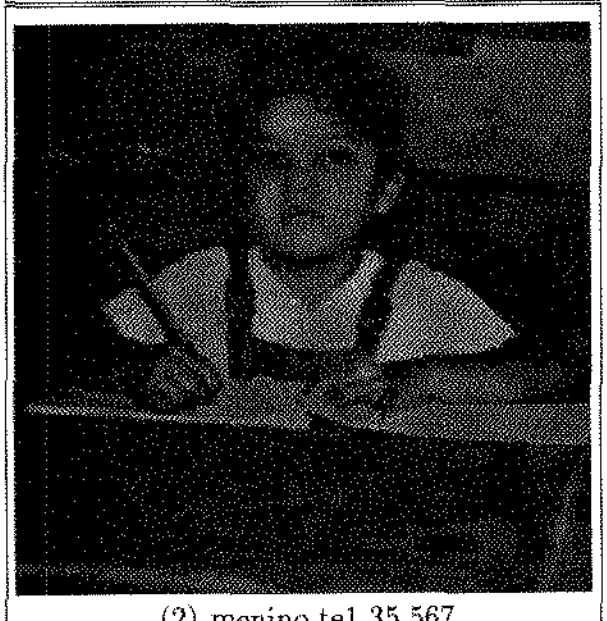
(1) mulher.tes 35.877

(2) mulher.te1 36.571

Figura 9.5: Imagens da Mulher



(1) menino.tes 34.098



(2) menino.te1 35.567

Figura 9.6: Imagens Menino

Bibliografia

[1] Yang, Y; Galatsanos, N. P. and Katsaggelos; A. K., REGULARIZED RECONSTRUCTION TO REDUCE BLOCKING ARTIFACTS OF BLOCK DISCRETE COSINE TRANSFORM COMPRESSED IMAGES, IEEE Trans. Circuits Syst. [Video Technol], Vol 3, no 6, pp. 421–432, Dec. 1993

[2] Yang, Y; Galatsanos, N. P.; and Katsaggelos, A. K., PROJECTION-BASED SPATIALLY ADAPTATIVE RECONSTRUCTION OF BLOCK-TRANSFORM COMPRESSED IMAGES, IEEE Transactions on Image Processing, Vol 4, *no* 7,pp 896–908, Jul. 1995

[3] Youla, Dante C.,GENERALIZED IMAGE RESTORATION BY THE METHOD OF ALTERNATING PROJECTIONS, IEEE Transactions on Circuits and Systems, Vol. 25, no. 9,pp. 694–702,Sept., 1978

[4] Youla,Dante C. and Webb, H.,IMAGE RESTORATION BY THE METHOD OF CONVEX PROJECTIONS: PART 1 – THEORY, IEEE Transactions on Medical Imaging, Vol. 1, no 2,pp. 81–94, Oct. 1982

[5] Stark, H., ED., IMAGE RECOVERY: THEORY AND APPLICATION, New York: Academic, 1987

[6] Jain, Anil K., FUNDAMENTALS OF DIGITAL IMAGE PROCESSING, Prentice Hall, New Jersey,1989

[7] Wallace, Gregory K.,THE JPEG STILL PICTURE COMPRESSION STANDARD, Communications of the ACM, Vol. 34, No. 4,pp. 30–45, April, 1991

[8] Pennebaker, William B. and Mitchell, Joan L.,JPEG STILL IMAGE DATA COMPRESSION STANDARD, Van Nostrand Reinhold, New York, 1993

[9] Gonzales, Rafael C. and Woods, Richard E., DIGITAL IMAGE PROCESSING, Addison-Wesley, New York, 1993

[10] Browder, F. E., FIXED-POINT THEOREMS FOR NONCOMPACT MAPPINGS IN HILBERT SPACE, Proc. Nat. Acad. Sci.,U.S.A., Vol. 53, pp. 1272–1276, 1965

[11] Halpern, B.,FIXED POINTS OF NONEXPANDING MAPS., Bull of the Am. Math. Soc., Vol. 73, pp. 957–961, 1967

[12] Gomes, Jonas e Velho, Luiz,COMPUTAÇÃO GRÁFICA: IMAGEM, Série de Computação e Matemática,IMPA/SBM, Rio de Janeiro, 1994

[13] Rao, K. R. and Yip, P., DISCRETE COSINE TRANSFORM: ALGORITHMS, ADVANTAGES, APPLICATIONS., New York: Academic, 1990

[14] Kreyszig, Erwin,INTRODUCTORY FUNCTIONAL ANALYSIS WITH APPLICATIONS, John Wiley & Sons,New York,1978

[15] Castleman, Kenneth R.,DIGITAL IMAGE PROCESSING,Prentice-Hall,London,1979

[16] Rosenfeld, Azriel and Kak, Avinash C.,DIGITAL PICTURE PROCESSING,Vol. 1, Academic Press, San Diego, 1982

[17] Combettes, Patrick L.,THE FOUNDATION OF SET THEORETIC ESTIMATION, Proceedings of the IEEE, Vol. 81, No. 2, pp. 182–208, Feb. 1993

[18] Lee, Byeong,A NEW ALGORITHM TO COMPUTE THE DISCRETE COSINE TRANSFORM,IEEE Transactions on Acoustics, Speech, and Signal Processing, Vol. ASSP-32, No. 6, pp. 1243–1245, Dec. 1984

[19] Jain, Anil K.,IMAGE DATA COMPRESSION: A REVIEW, Procedings of the IEEE, Vol. 69, No. 3, pp. 349–389, Mar. 1981

[20] Kay, David C. and Levine, John R.,GRAPHICS FILE FORMATS, Windcrest/McGraw-Hill,New York, 1995

[21] Golub, Gene H. and Van Loan, Charles F.,MATRIX COMPUTATIONS,The Johns Hopkins University Press, London, 1989

[22] Martinez, José Mario e Santos, Sandra Augusta,MÉTODOS COMPUTACIONAIS DE OTIMIZAÇÃO , 20 Colóquio Brasileiro de Matemática, IMPA,1995

[23] Gnedenko, B. V.,THE THEORY OF PROBABILITY, MIR Publishers, Moscow,1978

[24] Strang, Gilbert and Nguyen, Truong,WAVELETS AND FILTER BANKS, Wellesley-Cambridge Press, USA, 1996

[25] Malvar, Henrique S., SIGNAL PROCESSING WITH LAPPED TRANSFORM, Artech House, 1992.