

UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE ENGENHARIA ELÉTRICA
DEPARTAMENTO DE COMPUTAÇÃO E AUTOMAÇÃO INDUSTRIAL

Este exemplar corresponde à redação final da tese
defendida por Luciano Emídio Neves
da Fonseca e aprovada pela Comissão
Julgadora em 18 / 12 / 90.

Orientador



DESENVOLVIMENTO DE UM SISTEMA PARA
COLORIZAÇÃO DE IMAGENS DIGITAIS

Luciano Emídio Neves da Fonseca

Orientador: Prof. Dr. Clésio Tozzi

Tese apresentada à Faculdade de Engenharia Elétrica
da Universidade Estadual de Campinas, como parte
dos requisitos exigidos para a obtenção do título
de MESTRE EM ENGENHARIA ELÉTRICA.

8c/9105027

CONTEÚDO

1) INTRODUÇÃO	4
1.1) Motivação	4
1.2) Metodologia	11
2) UM MODELO PARA A COR	15
2.1) A física e a fisiologia da cor	15
2.2) Geração de cores a partir de cores primárias	20
2.3) Conversão de especificações	24
2.4) O algoritmo de conversão implementado	27
3) DETALHES DA IMPLEMENTAÇÃO	31
3.1) Hardware	31
3.2) Software	33
3.2.1) Ambiente de desenvolvimento	33
3.2.2) O pacote desenvolvido	35
4) DUAS SOLUÇÕES PARA O MESMO PROBLEMA	39
4.1) O princípio de segmentação e colorização	39
4.2) Arquiteturas de memórias de imagem	40
4.3) Compressão da imagem e partição da LUT	43
4.4) Geração de três matrizes de intensidade	47
4.5) Escolha da LUT ótima	48

ecografias, radiografias etc, carregam, muitas vezes, informações que estão mascaradas por pequenas diferenças de nível de cinza. O uso de um mapeamento de cores pode tornar visível algum sintoma, que estava inicialmente obscuro na chapa monocromática.

No caso específico da radiografia, pode-se fazer uma calibração inicial do aparelho de raio-x, tirando-se, para isto, algumas chapas com tensões controladas (40V, 70KV etc) de um calibrador padrão. O calibrador padrão, que pode ser visto na foto 7.11, é um objeto que apresenta orifícios de diversas profundidades. Da análise da chapa tirada deste objeto, consegue-se uma correspondência direta de nível de cinza na radiografia para profundidade do orifício no calibrador. A presença destas mesmas tonalidades em uma chapa específica de um paciente, tirada nas mesmas condições que foi tirada a chapa do calibrador, pode acusar algum sintoma que facilite o diagnóstico médico. No entanto, já que a radiografia é uma imagem monocromática, os níveis de cinza podem estar muito próximos, o que dificulta a interpretação da chapa. O comum é se usar um mapeamento de nível de cinza para cor, de forma que, cada cor utilizada, corresponderá a uma profundidade específica no objeto radiografado.

Outro exemplo prático de aplicações está na engenharia de petróleo, particularmente na engenharia de reservatórios e petrofísica. Atualmente, são usadas técnicas de tomografia computadorizada (raio-x e ressonância magnética nuclear), para a visualização de fenômenos interiores a testemunhos de rochas

7) COLORIZAÇÃO DE IMAGENS.....	110
7.1) Colorização por planos de corte.....	111
7.2) Colorização por funções.....	115
7.3) Colorização por interpolações nos níveis de cinza..	121
7.3.1) Interpolação linear.....	121
7.3.2) Interpolação cilíndrica.....	124
7.4) Colorização por falsa-cor.....	128
7.5) Colorização por conteúdo em frequência.....	129
7.6) Rotação da look-up-table.....	131
8) CONCLUSÕES.....	134
BIBLIOGRAFIA.....	141

1) Introdução

1.1) Motivação

O processo de colorização tem por objetivo a atribuição de uma cor específica a cada pixel de uma imagem digital monocromática. Uma imagem monocromática possui apenas a informação de luminância, isto é, o nível de cinza de cada um de seus pixels. Para que se possa atribuir uma cor a cada pixel da imagem, deve-se fazer um mapeamento de luminância para cromaticidade. Cromaticidade é a grandeza tridimensional que define unicamente uma cor.

A maior motivação para o uso do processo de colorização de imagens está na própria fisiologia do olho humano. Um observador médio não diferencia mais do que 32 níveis de cinza [Green/83]. Por outro lado, o olho humano tem imensa capacidade de distinção de cores, sendo capaz de distinguir algo entre 50.000 e 350.000 tons de cores.

Ainda não se pode definir quantitativamente, de forma única, a melhor maneira de se colorizar uma imagem digital monocromática. Como foi dito, a colorização de uma cena consiste num mapeamento de luminância para cromaticidade, isto é, de uma função unidimensional para uma tridimensional. O cálculo ótimo deste mapeamento é uma tarefa extremamente difícil [Bernstein/78], não só pelo aumento de dimensão envolvido, mas também pelo caráter

5) PACOTE BÁSICO DE PROCESSAMENTO DIGITAL DE IMAGENS	54
5.1) Convolução com núcleos	54
5.1.1) Especificações no domínio da frequência.....	55
5.1.2) Agudização de imagens.....	62
5.1.2.1) Filtro passa-altos.....	63
5.1.2.2) Laplaciano.....	64
5.1.2.3) Operadores de Sobel.....	65
5.1.3) Suavização de imagens.....	67
5.1.3.1) Filtro passa-baixas.....	68
5.1.3.2) Filtro do valor médio.....	68
5.1.3.3) Filtro do valor mediano.....	70
5.2) Manipulação de Histograma.....	71
5.2.1) Mapeamento de níveis de cinza.....	74
5.2.2) Equalização de histograma.....	77
5.2.3) Especificação de histograma.....	78
6) SEGMENTAÇÃO DE IMAGENS.....	80
6.1) Segmentação baseada em bordas.....	82
6.1.1) Thresholding com definição de bordas.....	83
6.1.2) Detecção de bordas.....	89
6.1.3) Edição direta na imagem.....	92
6.1.4) Correção de bordas.....	93
6.2) Segmentação baseada em textura.....	94
6.2.1) Crescimento de regiões.....	101
6.2.2) Subdivisão de área.....	105

subjetivo da colorização.

Devido à incapacidade de se definir um mapeamento ótimo de cores, é muito comum o uso de métodos empíricos na colorização de imagens - a arbitrariedade está sempre presente em qualquer processo de colorização. Pode-se atribuir uma cor a cada pixel da imagem baseando-se, por exemplo, em sua luminância (mapeamento direto de nível de cinza para cor). Há outras técnicas que, além da informação local (luminância), levam em conta relações espaciais com os pixels da vizinhança. No caso da colorização por conteúdo em frequência, associa-se uma cor a determinado pixel, a partir dos resultados da aplicação de filtros digitais numa vizinhança em torno deste pixel.

Já que o olho humano possui maior sensibilidade à cromaticidade, as cores podem ser usadas como meio de realçar pequenas diferenças de níveis de cinza, que seriam imperceptíveis na imagem monocromática. A imagem colorizada é, na maior parte das vezes, mais fácil de ser analisada e interpretada. Detalhes importantes da cena, ou mesmo um objeto específico, que estavam, possivelmente, mascarados por um baixo contraste na imagem monocromática, podem ser sensivelmente realçados, o que facilita o processo de interpretação dos dados e aumenta a velocidade de reconhecimento de alguma característica específica da cena analisada.

O campo de aplicações para o melhoramento por colorização de imagens é vasto. Imagens médicas como: tomografias,

[Paiva/90]. As imagens são geradas originalmente em níveis de cinza e sua colorização é de grande valia, não só para uma visualização mais sensível de regiões invadidas por fluidos de alta densidade (lama de perfuração), como também na identificação de algumas heterogeneidades (direções preferenciais de fraturas etc). Caso se queira, ainda, uma visualização clara e objetiva da distribuição de saturações em uma determinada secção da rocha, pode-se associar uma cor a cada fluido envolvido no processo (óleo, água etc). A proporção de cada cor em determinada região da imagem, acusa a concentração preferencial do fluido.

As imagens remetidas por satélites são, normalmente, adquiridas em uma janela espectral específica. Da análise destas fotos aeroespaciais, pode-se medir propriedades que estão associadas a uma determinada banda espectral. Pode-se detectar a presença de algum elemento como: água, vegetação, fonte de calor etc. Na maior parte das aplicações, cada elemento que se quer detectar está associado a um nível de cinza específico em determinada banda espectral adquirida ou, no caso mais geral, a níveis de cinza específicos em certas bandas espectrais. O uso de um mapeamento, de níveis de cinza para cor, pode-se associar diretamente uma cor distinta a cada elemento que se quer identificar.

Um outro campo de aplicações para o processo de colorização de imagens reside na controversa restauração e colorização de fotos e filmes monocromáticos. O processo utilizado em uma foto é o mesmo utilizado em cada quadro de um filme. Afora

considerações a respeito da validade ou não da aplicação de tais técnicas, o apelo a sua utilização é sempre o do melhoramento do aspecto estético de imagens [Ekstron/84], de forma que é sempre necessária a presença de um "diretor de arte" para a escolha das cores mais adequadas a cada cena.

Cada quadro do filme deve ser segmentado em regiões de interesse, que se que colorizar de forma independente. Cada segmento deve ser, então, colorizado, de acordo com algum conhecimento a priori da cena que está sendo processada. Os algoritmos mais adequados a este tipo de aplicação são as interpolações, principalmente a cilíndrica, que será apresentada no item 7.3. Procura-se, nesta interpolação, conservar a informação de luminância da imagem monocromática. No entanto, associa-se uma crominância diferente a cada nível de cinza de cada segmento da imagem. Pode-se ainda, aplicar a cada quadro da imagem, um algoritmo de processamento de imagens (manipulação de histograma, suavização de imagens etc), de forma a se restaurar algum contraste perdido ou mesmo retirar algum ruído ou falha presente na película original.

Para a completa colorização um filme monocromático, é necessário que se atribua uma cor específica a cada pixel, a cada quadro da película. Como normalmente os filmes possuem 24 quadros por segundo, a colorização de um longa-metragem de duas horas envolveria o processamento e colorização de 172.800 quadros, o que é uma tarefa proibitiva. Com o uso de segmentação, pode-se delimitar regiões em cada quadro, de forma a se atribuir uma cor a

cada região e não a cada pixel do quadro. Delimitadas as regiões, é possível o rastreamento automático destes segmentos nos quadros subsequentes. Neste caso, o volume de trabalho fica bastante reduzido. Mesmo assim, a cada nova cena do filme, a cada vez que a câmara muda de ângulo, a cada novo personagem que entra em cena, deve-se recomeçar o processo de colorização.

Como se vê, a segmentação possui aplicação direta na colorização de imagens. A colorização sempre se resume a atribuir cores específicas a determinados grupos de pixels ou regiões da imagem. Com a segmentação, determina-se quais pixels pertencem a cada objeto da imagem. Desta forma, fica viabilizada a colorização diferenciada de cada objeto da cena.

Outras aplicações importantes de segmentação de imagens são o rastreamento de objetos e a resconstituição de objetos 3D a partir de cortes 2D, como é o caso de tomografias médicas. No rastreamento de um objeto no espaço, dispõe-se de uma seqüência de imagens, na qual é necessário que se separe de uma maneira rápida e automática o objeto do fundo. Segmentado o objeto, pode-se gerar os sinais de controle necessários ao seu rastreamento. No caso de uma seqüência de tomografias, a segmentação permite a diferenciação, em cada corte ou slice, de regiões de interesse, como ossos, tecidos etc. A separação destas regiões é uma etapa necessária para a reconstrução da representação 3D do objeto.

A segmentação também é uma ferramenta muito útil à análise de uma imagem, pois os objetos ou entidades da cena são

destacados do fundo, facilitando-se, assim, um posterior reconhecimento ou descrição dos mesmos [Gonzalez/87]. No mais, a organização da informação visual em regiões ou segmentos é parte integrante do sistema visual humano, o que favorece qualquer processo de interpretação de cena.

O processo de digitalização sempre envolve ruído, perda de contraste etc. As imagens colorizadas no sistema foram, em sua totalidade, digitalizadas no próprio equipamento, tendo sido expostas a este tipo de problema. Desta forma, foi necessária a implementação de um pacote básico de processamento digital de imagens, para que se pudesse, entre outras coisas, corrigir possíveis deficiências provenientes da digitalização. As imagens processadas por possuírem, por exemplo, melhor contraste e escala de cinza otimizada, podem apresentar melhores resultados durante a colorização.

Em todas as aplicações descritas no trabalho, sempre se usou a cor de maneira a facilitar a identificação de alguma propriedade que estava associada a um determinado nível de cinza na imagem inicial monocromática. Dadas as características dos sistemas de digitalização, por mais que se tente uniformizar as condições de iluminação durante a digitalização, não é possível garantir a correspondência biunívoca entre os tons de cinza, presentes na foto ou cena, e níveis de cinza na imagem digitalizada pelo equipamento. Esta falta de uniformidade nos dados adquiridos prejudica a tentativa de se fazer uma associação consistente de uma cor com uma determinada propriedade a ser

identificada na imagem digitalizada. Isto requer uso de técnicas de melhoramento de imagens. A maneira encontrada para solucionar este problema foi o uso de técnicas de especificação de histograma, que será detalhada no item 5.2.4. Inicialmente se escolhia uma imagem para a calibração, cujo histograma de níveis de cinza seria tomado como padrão. As imagens adquiridas eram, então, pré-processadas, de forma a se conseguir um histograma de níveis de cinza semelhante ao da imagem usada na calibração.

O pré-processamento de imagens pode também auxiliar o processo de segmentação de imagens. A melhor definição de bordas e contornos permite a separação otimizada dos segmentos de interesse, diminuindo a incerteza provocada pelos baixos contrastes. Alguns métodos para colorização de imagens implementados utilizam diretamente algoritmos de processamento de imagens. Este é o caso da colorização por conteúdo em frequência, que usa a convolução com máscaras, como forma de aplicação de filtros digitais.

1.2) METODOLOGIA

O Sistema de Colorização e Segmentação de Imagens Digitais foi desenvolvido no Laboratório do Departamento de Computação e Automação Industrial da Faculdade de Engenharia Elétrica da Universidade Estadual de Campinas. As imagens processadas no sistema foram digitalizadas no próprio equipamento, com o auxílio de uma câmera monocromática. A seqüência de processamentos, sofrida por uma imagem no sistema,

pode ser vista na fig. 1.1.

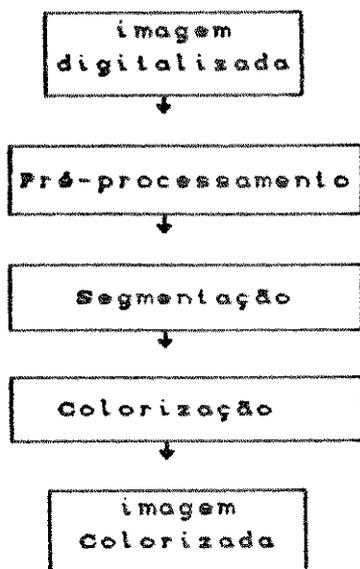


fig. 1.1 - Sequência de processamentos no sistema.

Primeiramente, a imagem adquirida pela câmera monocromática é melhorada através de um algoritmo de pré-processamento, para que sejam corrigidas possíveis deficiências do processo de digitalização ou mesmo realçadas algumas características, que facilitem algum processamento posterior ao qual será submetida a imagem. Os algoritmos de pré-processamento, que foram implementados no sistema, são discutidos no capítulo 5 - Pacote Básico de Processamento digital de Imagens. Após o pré-processamento, a imagem sofre um processo de segmentação, no qual são separadas regiões de interesse, as quais se quer colorizar de forma independente. Os algoritmos de

segmentação desenvolvidos e implementados no sistema, assim como uma discussão geral do processo de segmentação, são apresentados no capítulo 6 - Segmentação de Imagens. Uma vez segmentada a imagem, pode-se aplicar, a cada segmento, um algoritmo adequado de colorização. Os algoritmos de colorização desenvolvidos e implementados no sistema são discutidos no capítulo 7 - Colorização de Imagens.

Qualquer tentativa de se colorizar uma imagem exige, a princípio, familiaridade com a lei de formação das cores. O capítulo 2 - Um Modelo para a Cor - define fisicamente a cor e oferece ferramentas para a quantificação de uma cor no equipamento. O capítulo 3 trata diretamente dos detalhes da implementação do sistema. Discute-se o equipamento disponível e o ambiente de desenvolvimento (linguagem de programação, sistema operacional etc) utilizados. Apresenta-se também, neste capítulo, a estrutura básica do sistema desenvolvido. No capítulo 4 - Duas Soluções para o Mesmo Problema - são discutidas a influência das arquiteturas de memória de imagem na implementação de sistemas de segmentação e colorização. São também apresentadas as soluções adotadas para que as deficiências do equipamento disponível fossem contornadas.

Procurou-se utilizar o Sistema de Colorização e Segmentação desenvolvido no maior número possível de aplicações. Foram colorizadas imagens médicas, fotos aeroespaciais, tomografias computadorizadas etc. Foram também utilizados alguns exemplos de colorização, de modo a se ilustrar o processo de

restauração e colorização de filmes e fotos monocromáticos. Os resultados das colorizações são apresentados no capítulo 7, a medida que são discutidos os algoritmos de colorização implementados.

2) UM MODELO PARA A COR

A sensação de cor está intimamente relacionada com características físicas da luz, enquanto uma onda eletromagnética, e com características fisiológicas do olho humano. É feita uma análise destas características, de modo a se obter um modelo que quantifique adequadamente uma cor. Alguns modelos obtidos são orientados para o usuário, como é o caso do modelo HSV, enquanto outros, como o modelo RGB, parecem ser orientados para o equipamento. São feitas breves discussões sobre cada um destes modelos, assim como é apresentado o algoritmo desenvolvido, que permite a conversão de especificações em HSV para RGB e vice-versa.

2.1) A FÍSICA E A FISILOGIA DA COR

A luz é a fração do espectro eletromagnético que produz estímulos na retina do olho humano. Por ser uma onda eletromagnética, a luz pode ser descrita por algumas propriedades básicas, como: comprimento de onda (λ) e frequência (ν). Estas propriedades estão intimamente relacionadas, como pode ser visto na eq. 2.1, já que a luz se propaga com velocidade constante no vácuo ($c = 3 \times 10^8$ m/s).

$$c = \lambda \times \nu \quad (\text{eq. 2.1})$$

O olho humano é capaz de diferenciar comprimentos de onda

que vão de aproximadamente 350nm (violeta) a 780nm (vermelho). Todo este espectro de cores pode ser percebido graças à presença na retina de três diferentes tipos de receptores de luz (cones) [Harrington/87], que possuem picos de sensibilidade em comprimentos de onda que se aproximam do verde, vermelho e azul. A sensibilidade destes cones, em função do comprimento de onda, é mostrada na fig. 2.1. Os cones apresentam resposta positiva mesmo fora de seus picos de sensibilidade [Harrington/87]. Assim, qualquer comprimento de onda visível apresenta algum estímulo, embora bem atenuado, nos três receptores de luz.

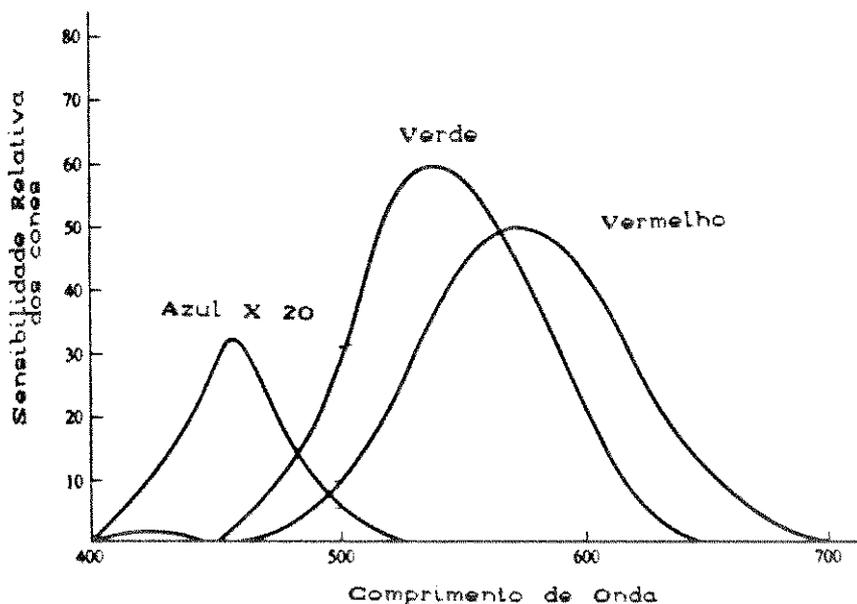


fig. 2.1 - Sensibilidade relativa dos diferentes cones da retina.

No diagrama de sensibilidade do olho humano da fig. 2.2, vê-se que o olho humano não apresenta a mesma sensibilidade aos diferentes comprimentos de onda. O olho é mais sensível ao verde, relativamente sensível ao vermelho e muito pouco sensível ao azul.

Uma energia luminosa incidente em uma superfície sofre reflexões, difrações e absorções seletivas, que dependem do comprimento de onda. O espectro refletido determina a cor percebida por um observador específico. Fala-se de um observador específico, pois a sensação de cor está fortemente associada às características fisiológicas de cada observador.

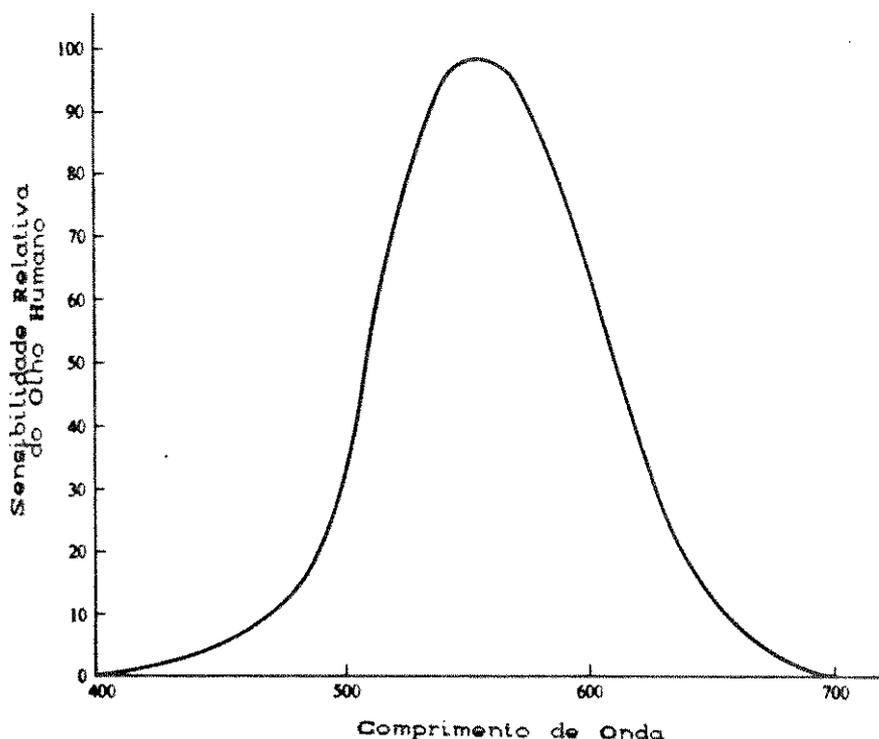


fig. 2.2 - Resposta relativa do olho humano à diferentes comprimentos de onda.

Não há uma correspondência biunívoca entre o espectro refletido e a cor percebida pelo observador. Muitas distribuições espectrais determinam a mesma sensação de cor no olho humano. Suponha-se uma luz monocromática com comprimento de onda 500nm, que excita os três cones receptores de luz. Observando-se o gráfico da figura 2.1, vê-se que o mesmo pode ser obtido a partir de duas luzes monocromáticas menos intensas, com comprimentos de onda 450nm (que excitará o cone azul) e 550nm (que excitará os cones vermelho e verde).

Esta redundância de informações sugere que o espectro luminoso possa ser analisado a partir de um modelo simplificado, que é mostrado na fig. 2.3. Neste modelo, a cor é subdividida em suas três componentes básicas de Ostwald [Hall/79]: matiz, saturação e luminância. O impulso de energia no comprimento de onda dominante L_1 , incorpora a noção subjetiva de matiz. O retângulo, que é uma energia uniformemente distribuída pelo espectro visível, pode ser entendido como a quantidade de branco diluído na cor. A relação entre a energia E_1 do impulso e a energia E_2 do retângulo quantifica a componente saturação da cor. A luminância está relacionada com valores absolutos de E_1 e E_2 .

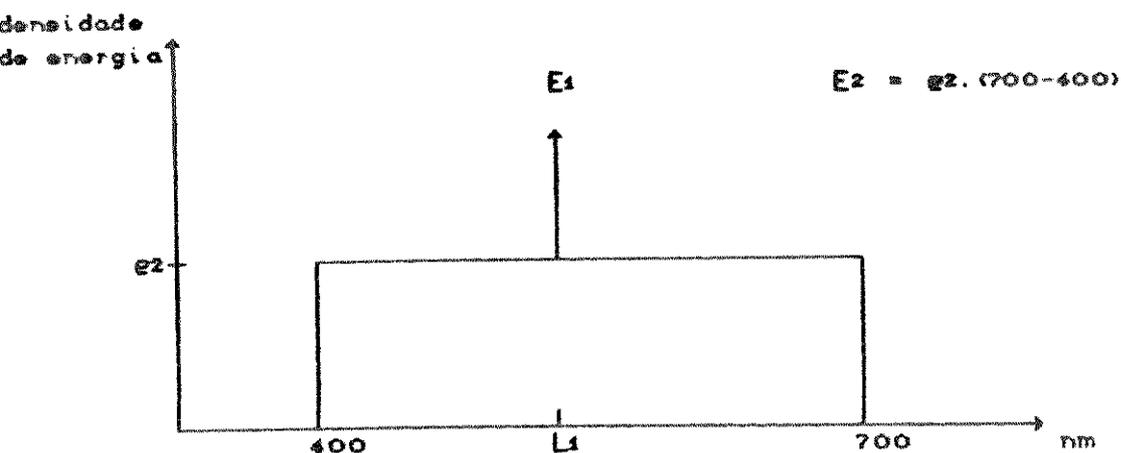


fig.2.3 -Distribuição espectral de energia de uma cor.

Matiz é a propriedade fundamental que permite a distinção de uma cor ou pigmento de outro. É a partir do matiz que se diferencia, com facilidade, um objeto amarelo de um vermelho, um verde de um azul. Quando se tenta distinguir um azul-pastel de um azul-rei ou um vermelho vivo de um rosa, utiliza-se a noção de saturação. A saturação quantifica a pureza da cor, isto é, o quanto a cor pura está diluída em branco. Quanto mais diluída em branco, menos saturada é a cor. Matiz e saturação determinam de forma única a cor básica. À informação contida nestas duas componentes dá-se o nome de Crominância. A terceira componente fundamental da cor - a Luminância - carrega a informação acromática de intensidade ou brilho da cor básica. O lugar geométrico gerado pela variação das três componentes básicas da cor é cilíndrico e pode ser visualizado na fig. 2.4.

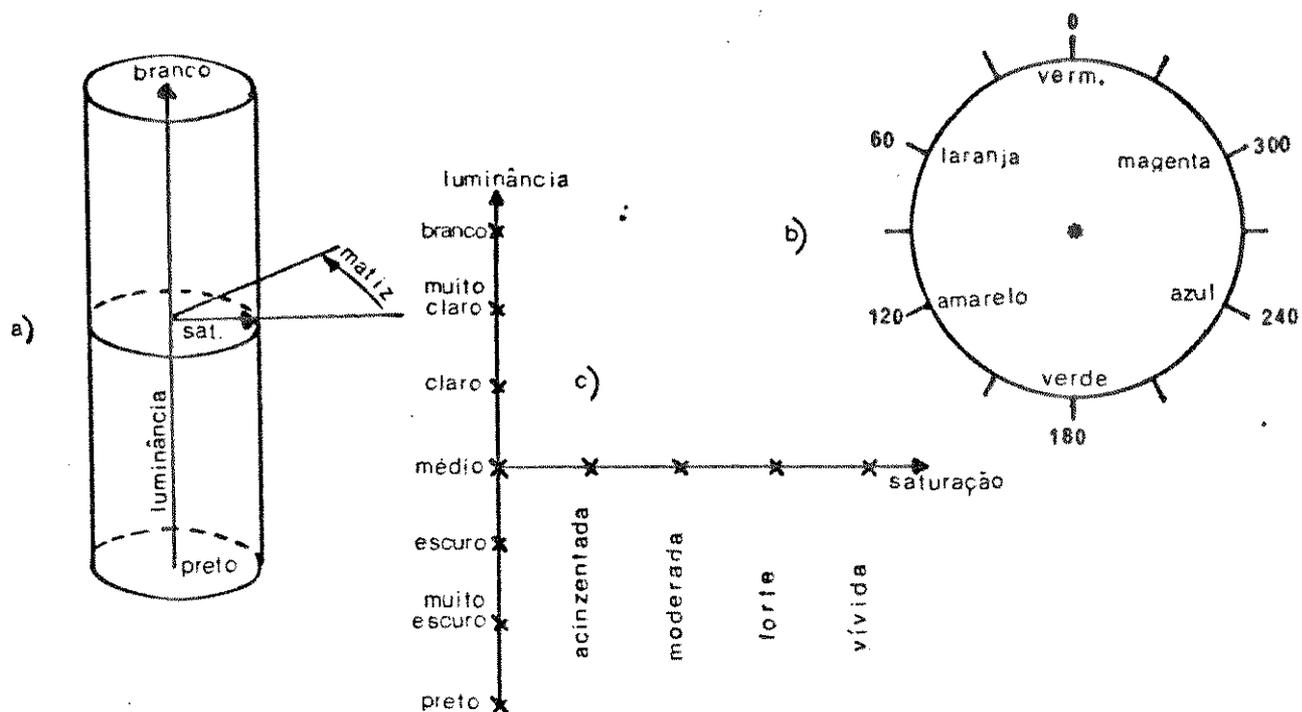


fig.2.4 - (a) Lugar geométrico das componentes fundamentais da cor. (b) detalhe da base do cilindro. (c) Um corte transversal

2.2) GERAÇÃO DE COR A PARTIR DA SOMA DE CORES PRIMÁRIAS

Apesar da cor estar bem definida em termos de suas componentes fundamentais: matiz, saturação e luminância, encontra-se, com frequência, situações nas quais se obtêm diferentes tons de cores a partir da combinação de três cores primárias. Um exemplo típico é o do televisor colorido. A tela do televisor é uma matriz de pontos triangulares, onde cada triângulo

possui, em seus vértices, uma quantidade adequada de fósforo emissor de luz. Um vértice com fósforo emissor de vermelho, um de verde e um de azul. A combinação adequada da intensidade de brilho de cada um dos vértices possibilita uma enorme variedade de cores. A propriedade fundamental reside no fato de que cores primárias, produzidas pelos fósforos emissores de luz, são somadas para que se produza uma cor secundária desejada. O lugar geométrico das cores obtidas pela soma de três primárias é cúbico, e pode ser visto na fig. 2.5. Na linha acromática, o vermelho, o verde e o azul assumem valores idênticos. Ao se percorrer esta linha, obtém-se uma escala que vai do preto ao branco, passando por todos os tons de cinza.

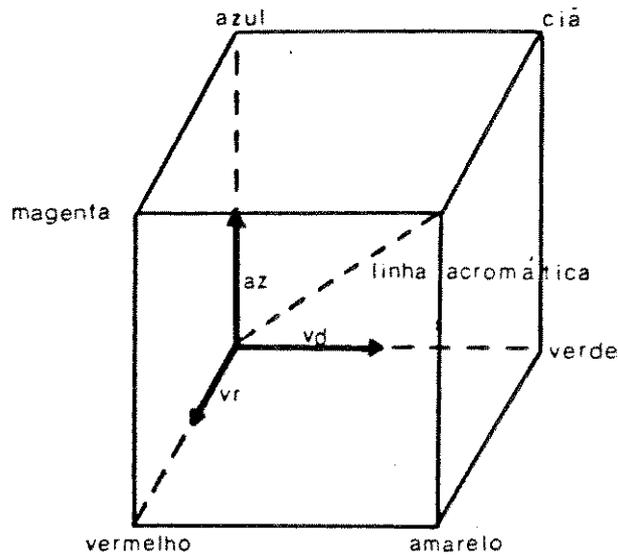


fig.2.5.- Lugar geométrico das cores obtidas a partir da soma das três cores primárias.

Nem todas as cores visíveis podem ser obtidas a partir da soma de três cores primárias quaisquer. A escolha das cores primárias determina o conjunto final de cores que se pode obter. Para fins de padronização, a CIE (Commission Internationale L'Eclairage) especificou as seguintes primárias fundamentais: vermelho/700nm, verde/541,1nm e azul/435.8nm [Foley/84]. Definiu-se também um diagrama de cromaticidade, representado na fig. 2.6, no qual todas as cores visíveis são normalizadas em relação à luminância. Um diagrama de cromaticidade colorido pode ser obtido em [Gonzalez/86] e [Pedrosa/82]. Para todas as cores do diagrama, tem-se que:

$$\text{Vermelho} + \text{Verde} + \text{Azul} = 1 \quad (\text{eq. 2.2})$$

Desta forma, todas as cores com mesma cromaticidade (matiz+saturação) e diferentes luminâncias serão mapeadas no mesmo ponto no diagrama. O diagrama é traçado com vermelho no eixo das abscissas e com o verde no eixo das ordenadas. O azul fica unicamente determinado pela equação 2.2. O ponto acromático, no meio do diagrama, satisfaz a seguinte relação:

$$\text{Vermelho} = \text{Verde} = \text{Azul} = 1/3 \quad (\text{eq. 2.3})$$

As cores puras estão nas bordas do diagrama. Caminhar pela borda equivale a variar o matiz da cor. Partindo-se de qualquer ponto da borda em direção ao ponto acromático, diminui-se a saturação da cor. A grande fração do diagrama destinada aos tons

de verde reflete o fato de o olho humano ser muito sensível a estes comprimentos de onda. Matizes que estiverem nas extremidades opostas de qualquer segmento que passe pelo ponto acromático são complementares. Cores complementares são aquelas cuja mistura produz o branco. A cor obtida pela mistura de duas cores quaisquer está situada em um ponto intermediário pertencente ao segmento de reta que une estas duas cores no diagrama.

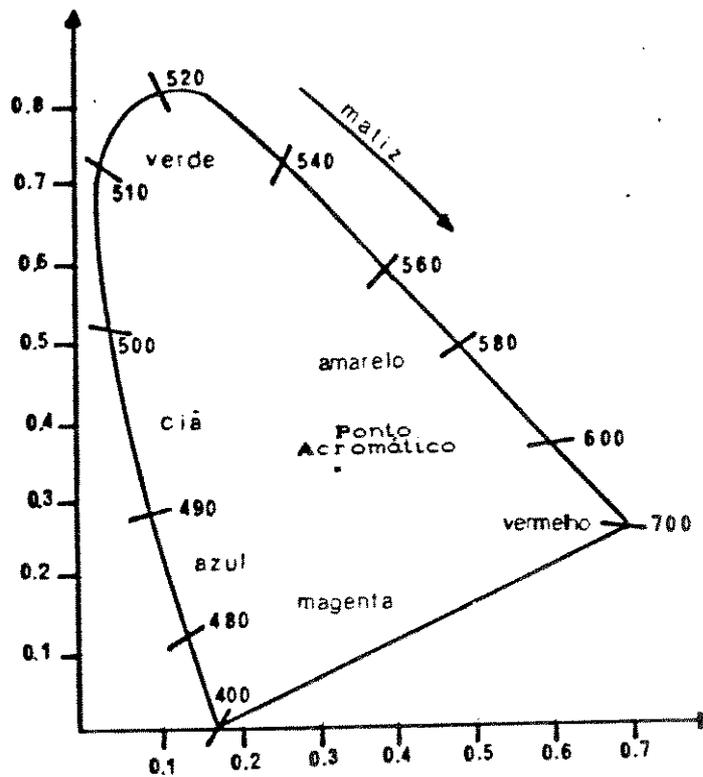


fig.2.6 - Diagrama de crominância do CIE.

A definição de cores a partir da soma de três primárias (vermelho, verde e azul) parece ser orientada para o equipamento, enquanto que a partir de suas componentes fundamentais (matiz,

saturação e luminância) parece ser orientada para o usuário [Encarnação/86]. As parcelas vermelho, verde e azul podem ser diretamente programadas no equipamento. No entanto, as componentes fundamentais parecem coincidir com a intuição.

Na colorização de a uma imagem, é fundamental que se tenha familiaridade com a formação das cores no equipamento. Trabalhando com as componentes fundamentais, o usuário percebe imediatamente estas leis de formação. No mais, pode-se escolher a cor básica diretamente do diagrama de cromaticidade do CIE, obtendo-se, assim, as suas componentes fundamentais.

2.3) CONVERSÃO DE ESPECIFICAÇÕES

No sistema de colorização de imagens desenvolvido, foram implementados algoritmos de conversão de especificações matiz-saturação-luminância (HSV) para vermelho-verde-azul (RGB), de forma que o usuário pode escolher interativamente uma cor, usando qualquer especificação que desejar. O usuário pode escolher, por exemplo, uma cor no sistema HSV, que é mais intuitivo. Em seguida, a especificação da cor é convertida para o sistema RGB, para que possa ser programada no equipamento.

A conversão de especificações RGB para HSV é feita em dois passos, de forma que se faz necessário o uso das variáveis intermediárias P, Q e T na eq. 2.4. Primeiramente é feita uma rotação, fazendo-se com que o eixo Z, da fig. 2.5, coincida com a reta dada pelas equações paramétricas $G=R$; $B=R$. Assim:

(eq. 2.4)

$$[P \ Q \ T] = [R \ G \ B] \begin{bmatrix} 2\sqrt{6} & 0 & 1\sqrt{3} \\ -1\sqrt{6} & 1\sqrt{2} & 1\sqrt{3} \\ -1\sqrt{6} & 1\sqrt{2} & 1\sqrt{3} \end{bmatrix}$$

O segundo passo corresponde à transformação para coordenadas cilíndricas, com o respectivo escalamento de intensidade. Assim:

$$H = \arctan (P / Q) \quad (\text{eq. 2.5})$$

$$S = (P^2 + Q^2)^{1/2}$$

$$V = T \sqrt{3}$$

Na conversão HSV pra RGB, usam-se as transformações inversas. Da mesma forma que na conversão anterior, utilizam-se as variáveis intermediárias P, Q e T, assim:

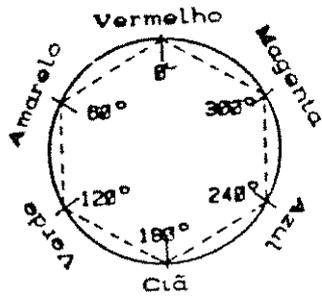
$$P = S \sin(H) \quad (\text{eq. 2.6})$$

$$Q = S \cos(H)$$

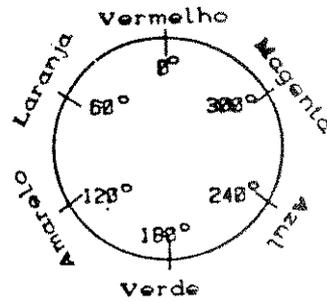
$$T = V / \sqrt{3}$$

$$[R \ G \ B] = [P \ Q \ T] \begin{bmatrix} 2\sqrt{6} & -1\sqrt{6} & -1\sqrt{6} \\ 0 & 1\sqrt{2} & -1\sqrt{2} \\ 1\sqrt{3} & 1\sqrt{3} & 1\sqrt{3} \end{bmatrix}$$

Nestas conversões tradicionais de HSV para RGB, obtem-se o círculo de cores mostrado na fig. 2.7.a. O ângulo, que assume valores entre 0° e 360° , descreve numericamente a variável matiz (H). Neste círculo, são reservados 120° para matizes entre o verde e o azul, enquanto há um sacrifício das matizes entre o vermelho e o amarelo - aos quais são reservados apenas 60° .



a)



b)

fig. 2.7 - a) círculo de cores HSV normal.
b) círculo de cores HSV modificado.

2.4) O ALGORITMO DE CONVERSÃO IMPLEMENTADO

No algoritmo de conversão implementado, procurou-se corrigir este problema, de forma que o vermelho, o amarelo e o azul se situassem à mesma distância no círculo das cores, o que corresponde mais à intuição do usuário [Encarnação/86], facilitando-se a interface homem-máquina. O círculo de cores resultante desta transformação pode ser visto na fig. 2.7-b. Como forma de facilitar os cálculos, a variável matiz foi aproximada pela distância ao longo do eixo de um hexacôno (fig. 2.8).

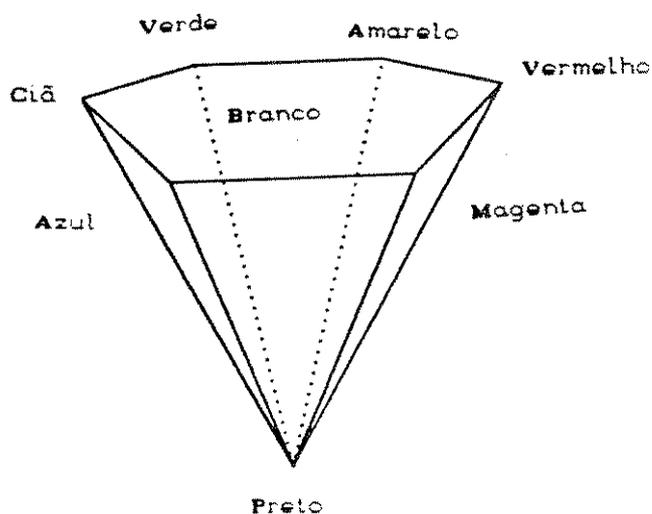


fig. 2.8 - Hexacôno de cores HSV.

A seguir, são apresentados os algoritmos de conversão de especificações implementados no Pacote de Processamento Digital de Imagens. Na implementação deste algoritmo, procurou-se fazer uma síntese de características adequadas encontradas nos vários algoritmos disponíveis na literatura. Exemplo destas características são a escala de cores otimizada e a simplicidade no cálculo da conversão para o hexacôno, ambas já discutidas. Outros algoritmos são apresentados em [Foley/84], [Encarnação/86] e [Harrington/87].

PROCEDIMENTO RGB_PARA_HSV;

(entrada r,g,b : valores no intervalo [0,1]
saída h,s,v : h no intervalo [0,360]
s e v no intervalo [0,1])

INÍCIO

max := MAX (r,g,b); (retorna o valor máximo)
min := MIN (r,g,b); (retorna o valor mínimo)
v := max;
s := 0;
h := 0;

SE (max < min) ENTÃO

INÍCIO

s := (max-min) / max;
ir := (max-r) / (max-min);
ig := (max-g) / (max-min);
ib := (max-b) / (max-min);

SE (r = max) ENTÃO

SE (g = min) ENTÃO h := 5 + ib
SENÃO h := 1 - ig

SENÃO SE (g = max) ENTÃO

SE (b = min) ENTÃO h := 1 + ir
SENÃO h := 3 - ib

SENÃO

SE (r = min) ENTÃO h := 3 + ig
SENÃO h := 5 - ir;

SE (h < 0) ENTÃO h := h + 6;

SE (h < 1) ENTÃO h := h * 120

SENÃO SE (h < 2) ENTÃO h := h * 60 + 60

SENÃO SE (h < 4) ENTÃO h := h * 30 + 120

SENÃO h := h * 60;

FIM;

FIM;

PROCEDIMENTO HSV_PARA_RGB;

Entradas h,s,v : h no intervalo [0,360]
s e v no intervalo [0,1]
saídas r,g,b : valores no intervalo [0,1])

INÍCIO

SE (h < 120) ENTÃO h := h / 120
SENÃO SE (h < 180) ENTÃO h = h / 60 - 1
SENÃO SE (h < 240) ENTÃO h = h / 30 - 4
SENÃO h = h / 60;

ih := INT(h);
f := h - ih;
p := v * (1 - s);
q := v * (1 - (s * v));
t := v * (1 - (s * (1 - f)));

CASO ih SEJA

0: (r,g,b) := (v,t,p);
1: (r,g,b) := (q,v,p);
2: (r,g,b) := (p,v,t);
3: (r,g,b) := (p,q,v);
4: (r,g,b) := (t,p,v);
5: (r,g,b) := (v,p,q);

FIM;

FIM;

DETALHES DA IMPLEMENTAÇÃO

De modo a se detalhar o Sistema de Processamento, Segmentação e Colorização de Imagens desenvolvido, primeiramente é apresentada uma breve descrição do equipamento utilizado para a implementação do sistema. No que diz respeito ao ambiente de desenvolvimento, são discutidos o sistema operacional e as linguagens de programação disponíveis. Por fim, é feita uma apresentação da estrutura básica do pacote de programas implementados para o sistema.

3.1) O HARDWARE

O equipamento utilizado possui um barramento local, padrão VME, pelo qual são interligados: uma CPU, a base de microprocessador motorola MC68000, e alguns periféricos opcionais, como: placas gráficas, placas digitalizadoras (Frame Grabber), disco rígido, disco flexível, co-processadores, impressora etc.

O módulo utilizado para o desenvolvimento pode ser visto na fig. 3.1. O Sistema Digitalizador de imagens, ou Frame Grabber, é composto de duas placas: PPI1/68K e PPI2/68K. Ambas as placas são conectadas ao barramento VME. A placa PPI1/68K possui uma memória RAM de 1024x2048 bytes, que é utilizada pela unidade de aquisição de imagens. A placa PPI2/68K faz a interface com o Terminal de Vídeo de média resolução MULTSYNC JC-1401P3ED e a Câmera de Vídeo CCD Kentec, modelo TK-204. A imagem que está sendo

adquirida pela câmara pode ser visualizada no monitor preto e branco, de forma a se poder focalizar adequadamente a imagem e escolher melhor o assunto. Também está conectada ao barramento VME uma interface, que monitora um Winchester Seagate ST-225 de 40 Mega bytes, uma impressora EPSON FX100 e um Floppy de 8 polegadas. À placa de CPU ELTEC E3-121/68K, a base de microprocessador Motorola 68000, estão ligados um terminal de vídeo VT100 com teclado e uma mesa digitalizadora (Tablet).

A placa PPI2/68K possui uma look-up-table de 256x18 bits. O nível de cinza de cada pixel (8 bits, logo 256 entradas) serve de endereço de entrada para a look-up-table, que é uma RAM de acesso rápido. Cada entrada de LUT, especificada por este endereço, oferece uma saída de 18 bits (6 para o vermelho, 6 para o verde, 6 para o azul), que constituem as saídas para os três canais de vídeo composto do monitor colorido de alta resolução. O preenchimento correto da look-up-table permite que se obtenha uma imagem final colorizada.

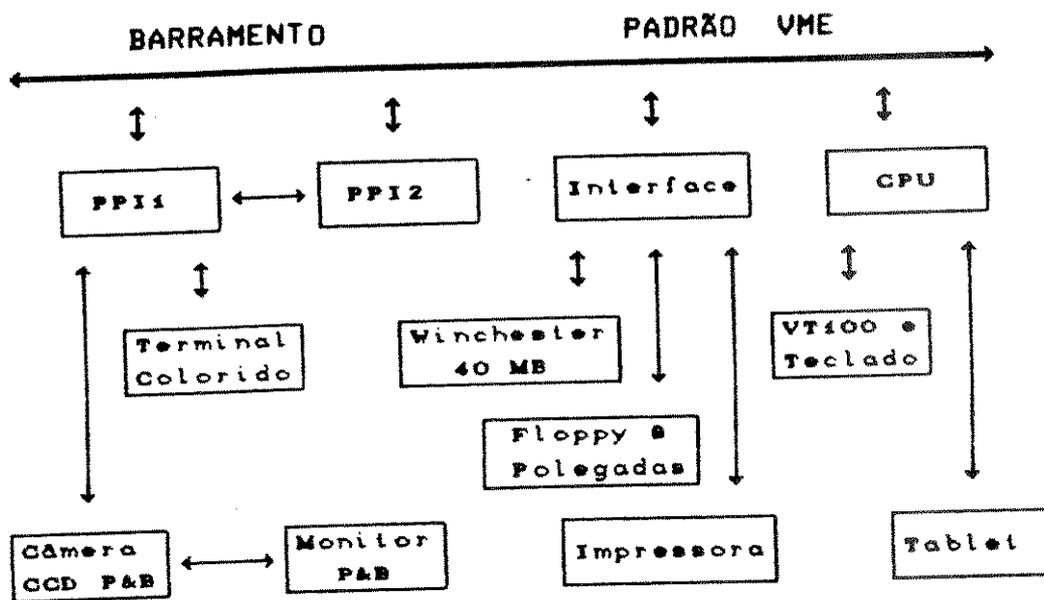


fig. 3.1 - Detalhe do módulo utilizado no desenvolvimento do trabalho.

3.2) SOFTWARE

3.2.1) AMBIENTE DE DESENVOLVIMENTO

O HOMUK utiliza o sistema operacional FLEXOS, que é muito semelhante ao sistema operacional HIGHER, desenvolvido para o computador Sperry Univac V70. É um sistema operacional de compartilhamento de tempo (time-sharing), que oferece um

gerenciamento dinâmico da memória e overlay automático de programas. Há uma alocação dinâmica de programas, de forma que estes nunca ocupam uma posição fixa na memória - o endereçamento é sempre relativo. A presença de um gerenciador de alocação residente na memória evita a necessidade de definição prévia dos endereços absolutos entre os módulos do programa (fase de link dos módulos). Os endereços absolutos são definidos durante a fase de execução, a medida que as rotinas são requisitadas e carregadas na memória principal.

O sistema inclui um compilador FLEXOS-FORTRAN e um montador DAS de códigos do Motorola MC68000. A linguagem FLEXOS-FORTRAN é uma extensão do FORTRAN IV padrão ANSI. A compilação é feita em dois passos. O primeiro passo converte o fonte em FLEXOS-FORTRAN para linguagem de montagem do MC68000 (assembly language). O segundo passo converte o fonte em linguagem de montagem para códigos de máquina. Isto permite que códigos mnemônicos do MC68000 sejam incluídos no meio do programa FLEXOS-FORTRAN.

Utilizou-se, também, um pré-processor FLECS, que é uma extensão da linguagem FORTRAN, principalmente no que se refere a mecanismos de controle. O sistema FLECS é baseado nos princípios de estruturação, de forma que se tem acesso a mecanismos mais elaborados de controle (WHILE, REPEATE UNTIL, WHEN ELSE etc), comuns a qualquer linguagem estruturada. Os programas escritos em FLECS estão na forma estruturada, o que facilita a visualização do algoritmo e a sua eventual tradução para uma outra linguagem

estruturada(C, PASCAL etc.)

Assim, os programas desenvolvidos em FLECS são submetidos ao pré-processador, que gera um outro fonte em FORTRAN. Este novo fonte é submetido ao compilador FORTRAN (fig.3.2) O processo é feito automaticamente, ficando transparente ao usuário.

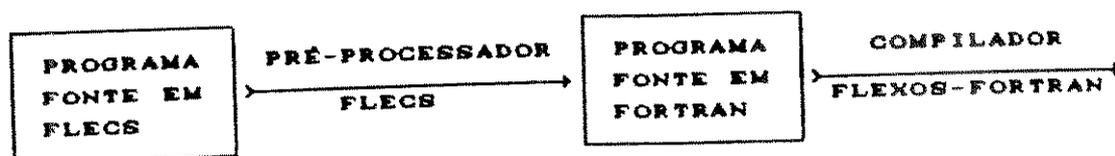


fig. 3.2 - Sequência para compilação de um programa em FLECS.

3.2.2) O PACOTE DESENVOLVIDO

O software desenvolvido para o sistema pode ser dividido nos seguintes módulos:

- 1- Gerenciador.
- 2- Pacote Básico de Processamento Digital de Imagens.
- 3- Pacote de Segmentação de Imagens.
- 4- Pacote de Colorização de Imagens.

O módulo gerenciador controla a aquisição de imagens via câmera preto e branco e o seu armazenamento na memória de vídeo da

placa PPI1/68K (2 Mega bytes). Pode-se ter até 16 imagens simultaneamente armazenadas na memória de vídeo. As imagens adquiridas e armazenadas nas memória podem ser visualizadas no monitor colorido de média resolução. O gerenciador também permite a transferência destas imagens, da memória para a unidade de disco rígido ou para o floppy de 8 polegadas e vice-versa.

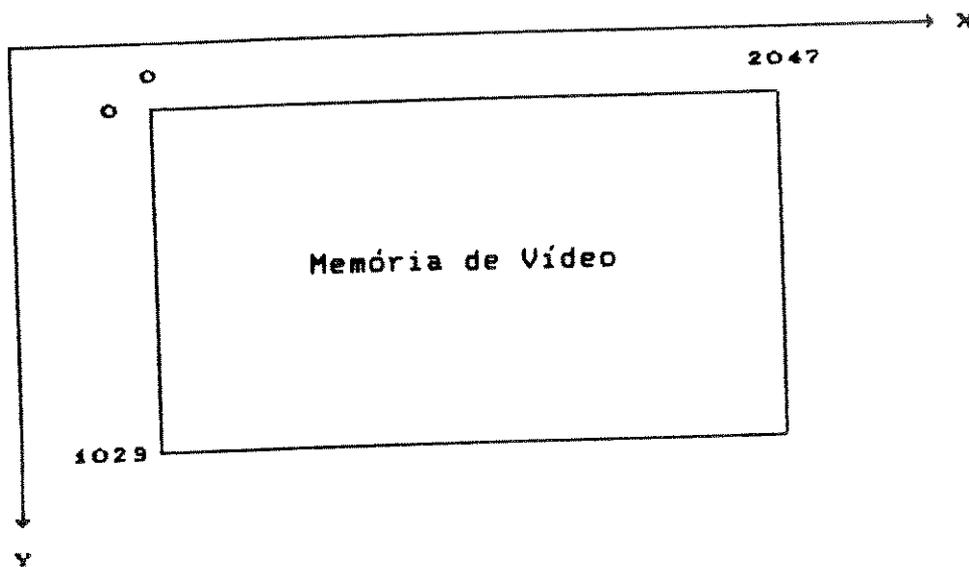


fig. 3.3 - Primeiro sistema de coordenadas para acesso à memória de vídeo.

Através de rotinas de alto nível implementadas no gerenciador, é possível o acesso a cada pixel da memória de vídeo. Este acesso é feito através de dois sistemas de coordenadas. No

primeiro sistema (fig. 3.3), usam-se dois eixos cartesianos X e Y, com a origem no canto superior esquerdo da memória. A variável X assume valores no intervalo [0,2047], enquanto a variável Y no intervalo [0,1023].

O digitalizador foi programado para a aquisição de imagens de 480x200 pixels de resolução, de forma que se pode ter até 16 imagens simultaneamente na memória de vídeo. Com o intuito de facilitar o acesso a esta memória, foi criado um segundo sistema de coordenadas, baseado no número da imagem (de 0 a 16). Escolhe-se o número da imagem e o acesso a um determinado pixel é feito pelo uso de um deslocamento relativo ao início desta imagem (fig. 3.4) [Alvarez/89].

Através do gerenciador, pode-se também ter acesso aos registradores da look-up-table. Assim, a tabela de cores pode ser alterada, por intermédio de uma rotina de alto-nível, o que facilita em demasia o processo de colorização. O pacote básico de processamento digital de imagens, o pacote de segmentação de imagens e o pacote de colorização de imagens são discutidos nos capítulos 5, 6 e 7 respectivamente.

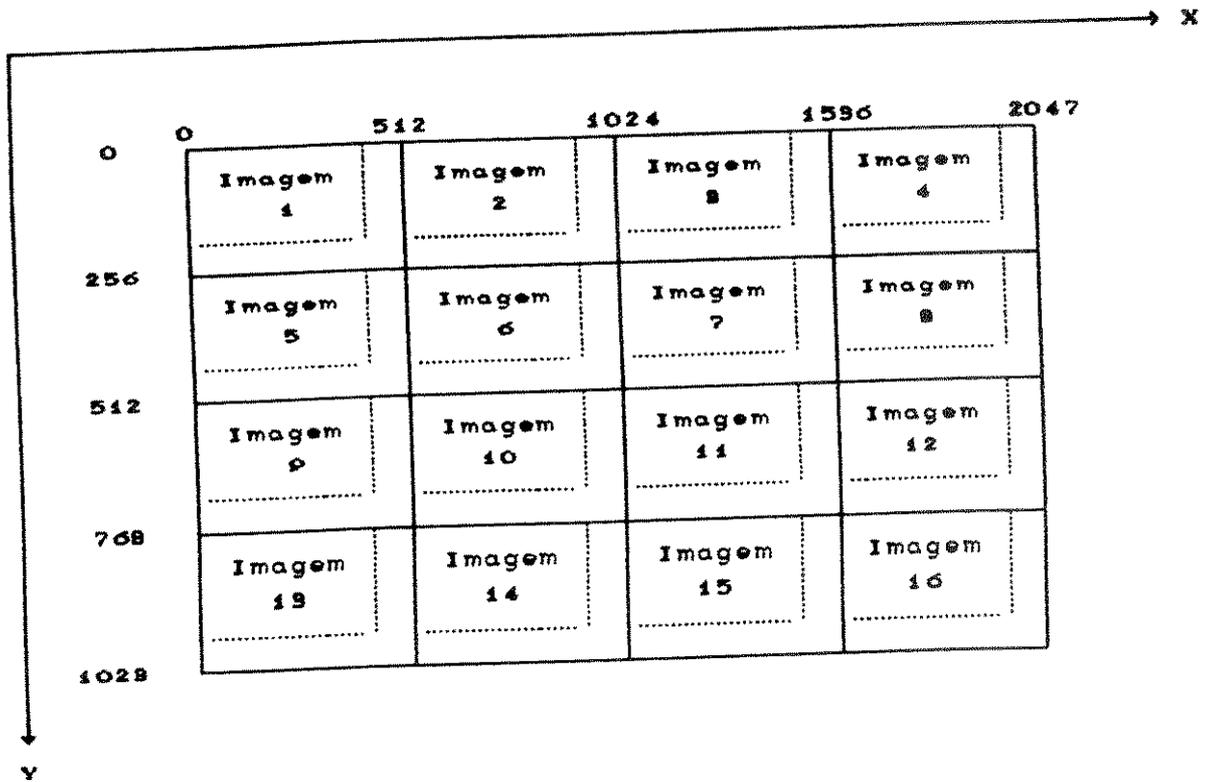


fig. 3.4 - Segundo sistema de coordenadas para acesso à memória de vídeo.

4) DUAS SOLUÇÕES PARA O MESMO PROBLEMA

A arquitetura da memória de imagem possui forte influência sobre os métodos de colorização e segmentação que se pode utilizar. Faz-se, primeiramente, neste capítulo, uma discussão do princípio de colorização e sua relação com o processo de segmentação de uma imagem digital. Em seguida, faz-se uma comparação das duas arquiteturas de memória de imagem mais comumente usadas. Por fim, apresenta-se as soluções adotadas para que as deficiências do equipamento disponível fossem contornadas.

4.1) O PRINCÍPIO DE SEGMENTAÇÃO E COLORIZAÇÃO

A colorização de uma imagem monocromática exige que se atribua uma cor específica a cada um de seus pixels. O processo fica facilitado, caso seja possível a colorização de regiões da imagem e não de cada pixel individualmente.

O processo de segmentação de uma imagem permite que se delimite regiões de interesse, as quais se quer colorizar de forma independente. Foram implementados no sistema desenvolvido, vários algoritmos de segmentação de imagens, de forma que é possível a separação de todos os objetos ou entidades da cena. Os algoritmos de segmentação implementados serão discutidos no capítulo 6.

Uma vez segmentada a imagem, deve-se atribuir, a cada região segmentada, uma cor específica ou, num caso mais geral, uma

mapeamento de níveis de cinza para cor. Os algoritmos de colorização de imagens, que serão detalhados no capítulo 7, permitem que se formalize a escolha de um mapeamento adequado a cada objeto da cena.

O princípio de segmentação e escolha de um mapeamento adequado a cada segmento constitui a base do processo de colorização de uma imagem. Este princípio está presente em uma colorização, seja qual for o equipamento que se tenha disponível. A arquitetura da memória de imagem apresenta influência determinante apenas na abordagem que se vai usar na implementação dos algoritmos. Os itens 4.3 e 4.4 apresentam as abordagens utilizadas no pacote desenvolvido.

4.2) ARQUITETURAS DE MEMÓRIA DE IMAGEM

Para que se possa gerar apropriadamente uma imagem digital colorida, deve-se, a princípio, utilizar três matrizes de intensidade $R(x,y)$, $G(x,y)$ e $B(x,y)$, onde x e y determinam a posição espacial do pixel. Cada uma das matrizes carrega informação referente a uma das componentes primárias da cor: vermelho, verde e azul. Uma ilustração de um equipamento que utiliza esta configuração pode ser vista na fig. 4.1. Este tipo de equipamento é conhecido como sendo de arquitetura segregada [Steavens/82]. Se cada componente primária puder assumir 256 níveis de intensidade (8 planos de memória), cada pixel da imagem poderá assumir 256^3 (16.777.216) cores diferentes!

Este tipo de sistema permite grande flexibilidade, pois é possível se atribuir, de forma independente, uma cor diferente a cada pixel da imagem. Este equipamento é muito usado para a visualização de imagens multiespectrais, recebidas de satélites. Pode-se escolher quaisquer três bandas espectrais da mesma imagem e armazená-las nas matrizes $R(x,y)$, $G(x,y)$ e $B(x,y)$. Obtém-se, assim, uma imagem final colorizada. Este processo se denomina colorização por falsa cor. No entanto, o uso de três planos de memória exige um equipamento mais elaborado. No mais, é necessário o triplo de memória que se utilizaria para armazenar uma imagem monocromática. Qualquer mudança na colorização da imagem exige processamento de grande quantidade de dados, diretamente na memória de vídeo.

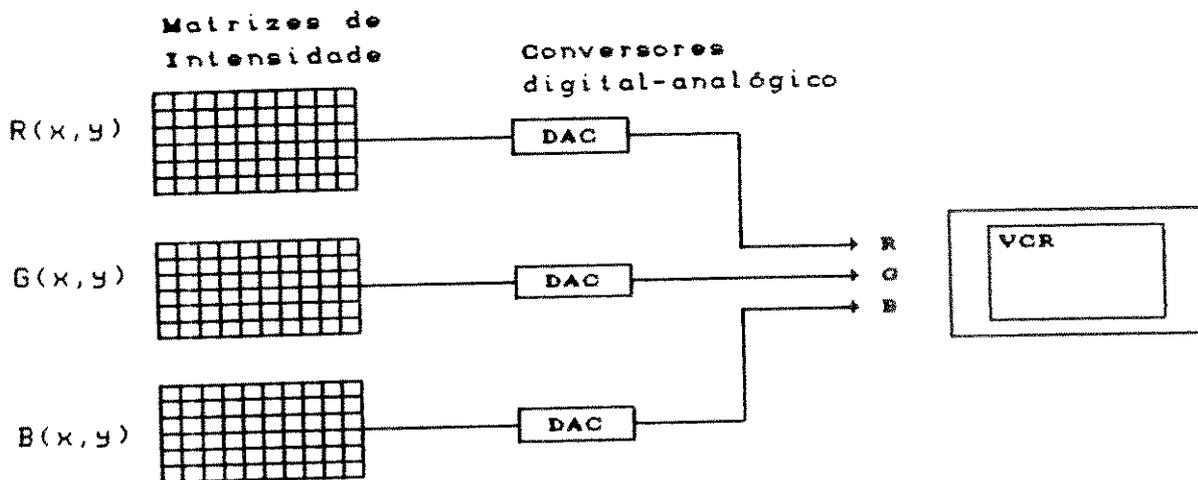


fig. 4.1.- Diagrama esquemático de hardware segregado, com três planos de memória.

Uma opção para se colorir uma imagem digital é o uso de tabela tríplice de consulta (ou LUT - Look-Up-Table). Basta que se tenha armazenado em memória a imagem monocromática $I(x,y)$ (veja fig.4.2). Este tipo de equipamento é conhecido como sendo de arquitetura integrada. A intensidade de cada pixel serve de índice de entrada para a LUT, que fornece um valor para o vermelho, verde e azul. Desta forma, tem-se um mapeamento direto da intensidade de nível de cinza para uma cor específica. O processo é muito rápido, pois só é necessário que se mude o conteúdo da LUT para se obter outra colorização da imagem.

A maior fragilidade do equipamento de arquitetura integrada reside na baixa seletividade da colorização. Como exemplo, suponha-se que, em uma imagem digitalizada de uma cena, se queira colorizar de verde um livro, que tenha sido digitalizado com o nível de cinza n . Para isto, basta que se coloque na entrada n da LUT a combinação vermelho-verde-azul necessária. Porém, não só os pixels do livro, mas todos os pixels da imagem que possuírem nível de cinza n , serão colorizados com verde. O sistema é mais eficiente quando se deseja um mapeamento direto de nível de cinza para cor, não se levando em conta considerações espaciais ou de vizinhança. No caso de se ter um equipamento de arquitetura segregada, não haverá nenhum problema de seletividade, pois é possível a atribuição de uma cor diferente a cada pixel da imagem.

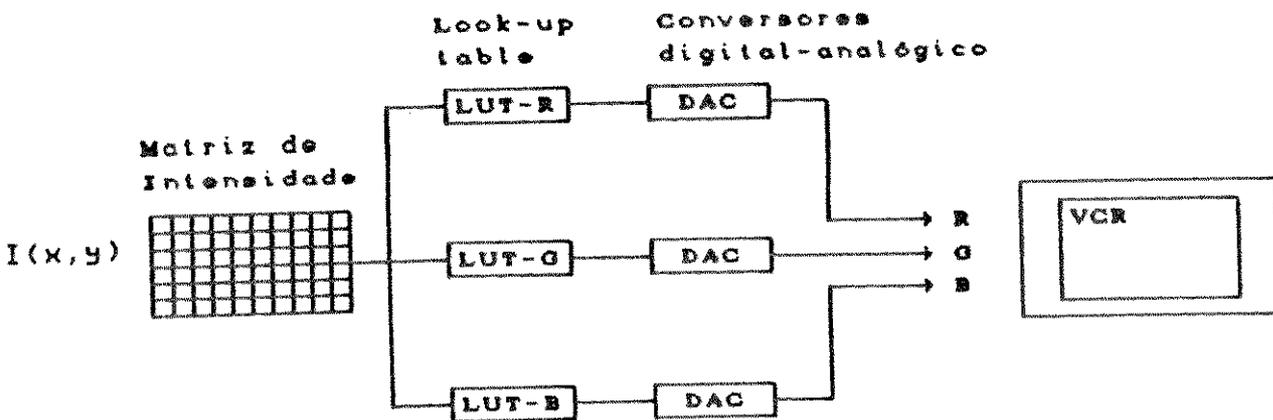


Fig.4.2.- Diagrama esquemático de hardware integrado, com um plano de memória e uma look-up-table.

O equipamento disponível em laboratório é um equipamento de arquitetura integrada. Ele possui uma matriz de intensidade com 8 planos de memória e uma look-up-table com 6 bits por componente. Os métodos que serão apresentados a seguir foram desenvolvidos com o intuito de se permitir uma seletividade razoável na colorização de imagens, mesmo com esta limitação do equipamento.

4.3) COMPRESSÃO DA IMAGEM E PARTIÇÃO DA LOOK-UP-TABLE

Como foi discutido, um equipamento com arquitetura integrada possui baixa seletividade na colorização. No entanto, alguma seletividade pode ser obtida através do processo de segmentação e compressão da imagem, com posterior partição da look-up-table [Fonseca/90]. Define-se, como segmento, uma porção da imagem que se quer colorizar de forma independente (veja item

6). Tem-se de garantir, que os níveis de cinza dos pixels de cada segmento não sejam encontrados em nenhum outro pixel da imagem, isto é, que cada segmento possua entradas independentes na look-up-table. No entanto, não se pode garantir que haja alguma entrada não utilizada na look-up-table, que possa ser usada, de forma independente, por um segmento específico.

O equipamento possui 8 planos de memória, o que permite, a princípio, que se tenha 256 níveis de cinza. No entanto, devido a uma limitação da look-up-table (esta possui apenas 6 bits), só é possível a visualização de 64 níveis de cinza. Desta forma, uma compressão da imagem de 256 para 64 níveis de cinza não alteraria em nada a resolução do sistema. No entanto, estariam disponíveis, após a compressão, $256 - 64 = 192$ entradas vazias na look-up-table. Dada uma imagem inicial $f(x,y)$ com 256 níveis de cinza, a imagem $g(x,y)$ comprimida com 64 níveis de cinza será:

(eq. 4.1)

$$g(x,y) = f(x,y) / 4 \quad ; \text{ divis\~ao inteira}$$

O olho humano não é capaz de diferenciar mais que 32 níveis de cinza [Green/83]. Vê-se, assim, que a imagem inicial $f(x,y)$ poderia, a princípio, ser comprimida para 32 níveis de cinza. No entanto, para que isto seja possível, é necessário que se tenha uma quantização ótima durante o processo de digitalização, o que não ocorre no equipamento disponível.

O pacote básico de processamento digital de imagens desenvolvido permite a aplicação de algoritmos de manipulação de histograma, que podem minimizar os efeitos de uma quantização imperfeita. Estes algoritmos serão detalhados no item 5.2. A imagem processada por estes algoritmos pode, sem muita perda de resolução, ser comprimida para 32 níveis de cinza. Com esta compressão, tem-se disponível $256-32=224$ entradas independentes na look-up-table. A imagem comprimida $g(x,y)$ com 32 níveis de cinza será:

(eq. 4.2)

$$\begin{array}{ccc}
 & \begin{array}{c} \text{minimização de erros} \\ \text{de quantização} \end{array} & \\
 f(x,y) & \xrightarrow{\hspace{2cm}} & h(x,y) \\
 \\
 g(x,y) & = & h(x,y) / 8 \quad ; \text{ divis\~ao inteira}
 \end{array}$$

Comprimida a imagem, tem-se 224 níveis de cinza não utilizados que podem servir como entradas na look-up-table para os segmentos gerados pelo processo de segmentação. Aos níveis de cinza de cada segmento, deve-se somar uma constante C_n , de forma a deslocá-los para estas entradas não utilizadas na look-up-table. Os níveis de cinza do segmento n assumir\~ao valores no intervalo $[C_n, 32+C_n]$. As constantes devem ser escolhidas de forma que não haja sobreposição de segmentos na look-up-table.

O número de segmentos que podem ser gerados depende

basicamente da necessidade do usuário. Pode-se ter, por exemplo, $224 / 32 = 7$ segmentos independentes de 32 níveis de cinza. No entanto, raramente um segmento necessita de 32 níveis de cinza. Muitas vezes 7 ou 8 entradas na look-up-table são suficientes (quando não apenas uma!). O banco de dados desenvolvido permite que se tenha segmentos de qualquer tamanho, o que maximiza o uso das entradas disponíveis na look-up-table. A fig. 4.3 ilustra uma possível partição para os segmentos.

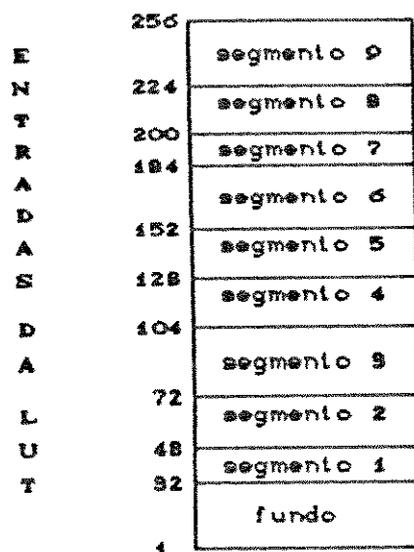


fig 4.3 - Ilustração de uma possível partição da look-up-table.

O processo de compressão e partição da look-up-table é sempre acompanhado de alguma perda de resolução. Porém, desde que não se comprometa em demasia a qualidade da imagem, ampliam-se satisfatoriamente as possibilidades de colorização, quando se tem

um equipamento de arquitetura integrada.

4.4) GERAÇÃO DE TRÊS MATIZES DE INTENSIDADE

Como se viu em 4.3, a geração de uma imagem colorizada, a partir do método da compressão da imagem e partição da LUT, é sempre acompanhada de perda de resolução. Pode-se resolver este problema, caso se opte pela geração direta de três matrizes de intensidade $R(x,y)$, $G(x,y)$ e $B(x,y)$. Nesta abordagem, todo o processamento é feito como se houvesse disponível um equipamento true-color (arquitetura segregada). Da mesma maneira que se procedia com o método anterior, a imagem deve ser separada em segmentos que se quer colorizar de forma independente. Atribui-se uma paleta de cores a cada um destes segmentos.

Porém, não há necessidade de se comprimir a imagem inicial monocromática, pois serão geradas 3 imagens com 256 níveis de cinza. Uma carrega a informação da componente básica vermelho ($R(x,y)$), uma do verde ($G(x,y)$) e a outra do azul ($B(x,y)$). Como não há partição da LUT, é necessário que se use uma quarta imagem auxiliar ($A(x,y)$), que carrega a informação dos limites de cada segmento, ou melhor, a que segmento pertence cada pixel da imagem. Os algoritmos de colorização alteram diretamente o conteúdo das três matrizes de intensidade na memória de vídeo, se referenciando à imagem $A(x,y)$ para a delimitação do segmento que está sendo colorizado.

Com o uso desta abordagem, nenhum dos métodos de

segmentação e colorização desenvolvidos no sistema ficam invalidados, caso se tenha disponível um equipamento de arquitetura segregada (3 matrizes de intensidade). O método de quantização da look-up-table pela mediana, que será descrito em 4.5, faz, de certa forma, uma ponte entre estas duas possibilidades de colorização. Este método comprime três matrizes de intensidade, de uma especificação segregada, em uma matriz de intensidade mais uma look-up-table. Desta forma, pode-se visualizar no equipamento disponível, com alguma perda de qualidade, o resultado que se teria em um equipamento segregado.

4.5) ESCOLHA DA LUT ÓTIMA

Caso se opte pela abordagem descrita em 4.5, que utiliza de três matrizes de intensidade, esbarra-se com limitação do equipamento disponível, que é do tipo integrado. Com o uso destas três imagens, já que vermelho, o verde e o azul podem assumir valores no intervalo $[0,255]$, pode-se escolher, para cada pixel da imagem, uma cor entre 16.777.216 cores diferentes. Como o monitor foi programado para uma resolução de 200×480 pixels, pode-se ter, no pior caso, 96.000 cores diferentes, simultaneamente na tela. Do outro lado, está o equipamento disponível, que é do tipo integrado. Neste tipo de hardware, pode-se ter no máximo 256 cores simultâneas na tela, escolhidas de uma paleta de $64 \times 64 \times 64 = 262.144$ cores (a look-up-table só possui 6 bits).

Desta maneira, caso se queira visualizar esta imagem no equipamento disponível, deve-se fazer um truncamento dos dados, de

forma a se escolher 256 cores, que melhor representem o universo de cores existentes nas imagens iniciais $R(x,y)$, $G(x,y)$ e $B(x,y)$.

Cada cor é representada por uma palavra de 24 bits (8 para cada componente fundamental), que deve ser truncada em uma palavra de 8 bits. Esta palavra servirá de endereço para a look-up-table.

Vários métodos já foram propostos para se conseguir a melhor partição do espaço RGB - promoção do truncamento ótimo da informação de cor. Alguns métodos, assim como a comparação dos diversos resultados, podem ser encontrados em [Heckbert/82, Stevens/82, Schaale/89, Vilaca/89]. O método implementado foi o da partição pela mediana. Este método subdividirá o espaço RGB em 256 sub-espacos, de forma a minimizar os erros de quantização na partição. O ideal é que cada sub-espaco possua o mesmo número de pixels.

Faz-se, inicialmente, um truncamento da informação, de cada componente fundamental, de 8 para 5 bits (32 níveis). O uso de 5 bits por cor primária é um truncamento legítimo, pois de acordo com [Wyszecki/82], o olho humano só distingue algo entre 50.000 e 350.000 cores.

A partir destas três imagens de 5 bits, deve-se construir um histograma tridimensional, que registrará a frequência absoluta de ocorrência de cada cor na imagem. O seguinte fragmento de algoritmo resume o cálculo deste histograma:

```

VAR      Histo3D array[0..31,0..31,0..31] of integer;
PARA    y = 1 , número_linhas FAÇA:
  PARA  x = 1 , número_colunas FAÇA:
    Histo3D (R(x,y),G(x,y),B(x,y)) :=
      Histo (R(x,y),G(x,y),B(x,y)) + 1;
  FIM;
FIM;

```

O algoritmo de partição pela mediana se inicia com a compressão do histograma tridimensional, de forma a se descartar as coordenadas extremas, que sejam nulas (veja fig. 4.4-a). Deve-se, então, escolher o maior eixo coordenado (R, G ou B) do maior sub-espaco do histograma (inicialmente só há um: o próprio histograma). Este sub-espaco deve ser segmentado em sua mediana, em relação a este eixo escolhido (veja fig. 4.4-b). Assim, ter-se-ão dois novos sub-espacos com um número equivalente de pixels (caso a partição seja realmente feita na mediana). Estes dois novos sub-espacos devem ser comprimidos, para a eliminação das coordenadas extremas que sejam nulas. Se já houver 256 sub-espacos, o algoritmo retorna. Caso contrário, um novo sub-espaco e um novo eixo são escolhidos, para que seja feita uma nova partição pela mediana.

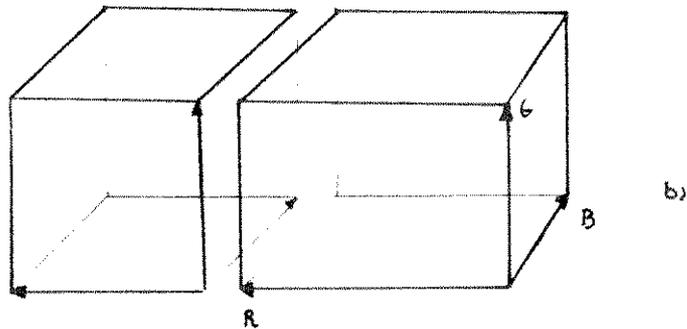
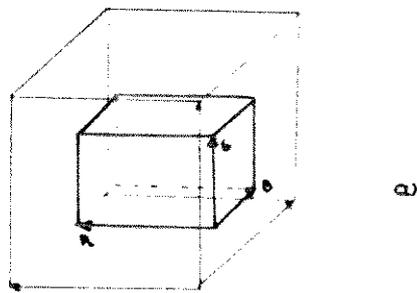


fig. 4.4 - (a)- Eliminação das coordenadas extremas nulas em um sub-espaco. (b) - Segmentação do sub-espaco em sua mediana, em relação ao seu maior eixo (R).

Subdividido o espaco, todas as cores, que estiverem dentro de cada sub-espaco, serão mapeadas em uma única cor. A cor escolhida para representar o sub-espaco deve ser o centro geométrico deste sub-espaco (veja fig. 4.5).

A seguinte algoritmo que resume o processo de partição do espaco RGB pela mediana:

Calcule Histo3D;

Elimine os zeros extremos de Histo3D;

Tome Histo3D como sendo o primeiro sub-espaco;

número-sub-espacos := 1;

REPITA ATÉ (número-sub-espacos = 256);;

1 - Escolha o maior sub-espaco;

2 - Escolha o maior eixo deste sub-espaco;

3 - Segmente este sub-espaco em relação à este eixo,
na sua mediana, em dois novos sub-espacos;

4 - Elimine os zeros extremos destes dois novos
sub-espacos;

5 - número-sub-espacos := número-sub-espacos + 1;

FIM;

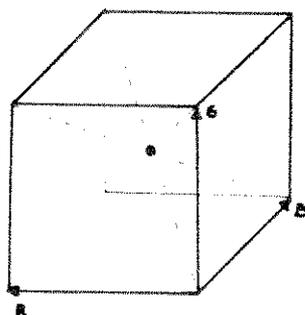


fig. 4.5 - Escolha do centro de geométrico para melhor representar o conjunto de cores do sub-espaco.

Escolhidas as 256 cores mais significativas para a imagem, é montada uma imagem índice $I(x,y)$, cujo valor de cada pixel se refere ao número do sub-espaco que contém a cor, referenciada pelas componentes R, G e B no ponto (x,y) . A look-up-table é montada, de forma que cada uma de suas entradas contenha a cor selecionada para o sub-espaco correspondente. Esta imagem índice, com sua respectiva look-up-table, pode ser visualizada no equipamento integrado disponível.

Na maioria das aplicações, o método de compressão da imagem e partição da look-up-table (veja item 3.1) se mostrou mais eficiente que este método apresentado. O truncamento de informação de cores de 24 para 8 bits implica sempre numa severa perda de resolução em cor. No entanto, esta implementação foi importante, pois permitiu que se tivesse uma noção do comportamento de um equipamento de arquitetura segregada (true-color). No caso do uso deste tipo de equipamento, nenhum dos métodos de colorização fica invalidado, pois serão geradas 3 imagens com cores primárias. A única diferença residirá no fato de não mais ser necessário o uso deste algoritmo de escolha da look-up-table ótima.

5) PACOTE BÁSICO DE PROCESSAMENTO DIGITAL DE IMAGENS

O pacote básico de processamento digital de imagens foi desenvolvido com o intuito de oferecer ferramentas que pudessem ser utilizadas na correção de possíveis deficiências advindas do processo de digitalização. No item 5.1, são apresentados os núcleos de convolução, que possuem aplicação direta, entre outras, na agudização e suavização de imagens. No item 5.2, são discutidos os algoritmos de manipulação de histograma de níveis de cinza, que são de grande valor quando se deseja padronizar dados de entrada para o sistema de colorização.

5.1) CONVOLUÇÃO COM NÚCLEOS

O processamento de imagens digitais, com o auxílio de núcleos de convolução, permite que se aplique de uma maneira simplificada, filtros digitais a uma imagem inicial. Pode-se, deste modo, utilizar toda uma metodologia já desenvolvida de análise no domínio da frequência. O item 5.1.1 formaliza o método de obtenção dos coeficientes do núcleo de convolução a partir de especificações no domínio da frequência. Dentro das aplicações dos núcleos de convolução, o item 5.1.2 trata sobre a agudização de imagens, que é usada no melhoramento da definição dos contornos e bordas da imagem. O item 5.1.2 trata da suavização de imagem, que é usada na redução do efeito dos ruídos presentes uma imagem digital.

5.1.1) ESPECIFICAÇÕES NO DOMÍNIO DA FREQUÊNCIA

Uma transformação linear \mathbb{T} sobre uma imagem $f(x,y)$, que produz uma imagem transformada $g(x,y)$, pode ser expressa de uma maneira geral como :

$$g(x,y) = \mathbb{T} [f(x,y)] \quad (\text{eq. 5.1})$$

A transformação \mathbb{T} é definida em uma vizinhança do ponto (x,y) . A vizinhança mais comumente usada é a quadrada e centrada no ponto (x,y) . Os coeficientes n_i (figura 5.1) da transformação \mathbb{T} definem o núcleo de convolução.

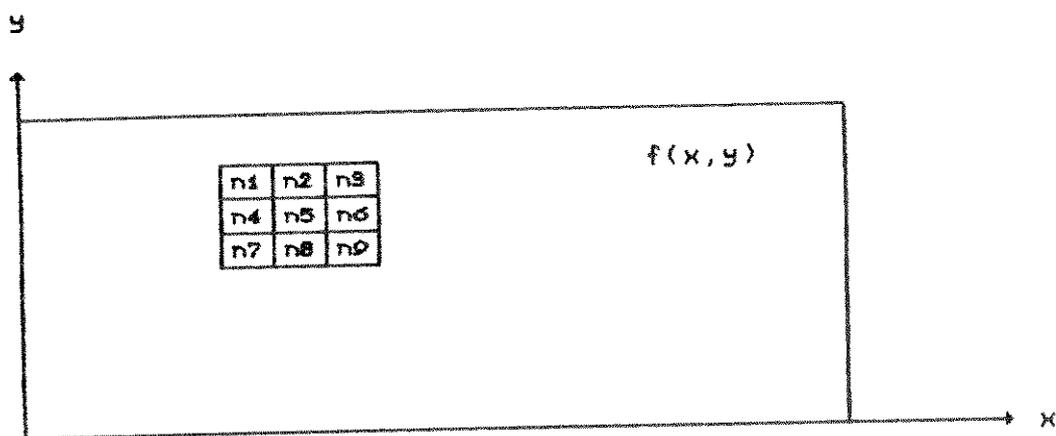


fig.5.1 - Núcleo de convolução aplicado a uma imagem $f(x,y)$.

Desta forma, para esta vizinhança quadrada 3x3 em torno do ponto (x,y) , $g(x,y)$ é expressa por:

(eq. 5.2)

$$g(x,y) = n1.f(x-1,y-1) + n2.f(x,y-1) + n3.f(x+1,y-1) + \\ n4.f(x-1,y) + n5.f(x,y) + n6.f(x+1,y) + \\ n7.f(x-1,y+1) + n8.f(x,y+1) + n8.f(x+1,y+1)$$

O núcleo de convolução é aplicado a todos os pixels da imagem $f(x,y)$, de modo a se obter a imagem final processada. A escolha adequada dos coeficientes do núcleo de convolução possibilita uma enorme variedade de efeitos na imagem processada, como por exemplo, a eliminação de ruídos, detecção de bordas etc.

Pode-se analisar um núcleo de convolução como se este fosse a resposta impulsional de algum filtro que se queira aplicar à imagem $f(x,y)$ [Oppenheim/75]. Desta maneira, é possível se obter os coeficientes n_i do núcleo de convolução, a partir de alguma especificação em frequência que se deseje obter. Da análise de sinais bidimensionais, sabe-se que bordas, contornos e outras mudanças rápidas de intensidade estão relacionadas com as componentes de alta frequência da transformada de Fourier da imagem. Assim, pode-se realçar as bordas da imagem, amplificando-se as componentes de alta frequência e/ou atenuando-se as de baixa frequência, ou melhor, aplicando-se um filtro passa-altas. Da mesma maneira, uma imagem pode ser suavizada (borrada), pela aplicação de um filtro passa-baixas, pois as componentes de alta frequência - que estão relacionadas com contornos, ruídos etc. - serão atenuadas.

Para o projeto dos filtros, foi escolhida a aproximação FIR (Finite Impulse Response), de forma a se poder limitar o tamanho da resposta impulsional do filtro à dimensão escolhida para o núcleo de convolução. O filtro é projetado em uma dimensão e posteriormente estendido para o caso bidimensional pelo método da resposta impulsional separável [Dudgeon/83].

Dadas a função de transferência $H(e^{j\omega})$ do filtro e sua resposta impulsional $h(n)$, tem-se pelo teorema da convolução que $H(e^{j\omega})$ é a transformada de Fourier de $h(n)$, assim:

$$H(e^{j\omega}) = \sum_{n=-\infty}^{\infty} h(n) e^{-j\omega n} \quad (\text{eq. 5.3})$$

Supondo, como exemplo, que se deseja aplicar um núcleo de convolução 3×3 à imagem, o filtro FIR correspondente, dado pela equação 5.3, pode ser expresso como:

(eq. 5.4)

$$H(e^{j\omega}) = \sum_{n=-1}^1 h(n) e^{-j\omega n}$$

$$\begin{aligned} H(e^{j\omega}) &= b e^{j\omega} + a + b e^{-j\omega} \\ &= a + 2 \cdot b \cdot \cos \omega \end{aligned}$$

Onde:

$$a = h(0)$$

$$b = h(-1) = h(1)$$

Dado que a frequência de corte do filtro é ω_c e que o filtro é passa-baixas:

(eq. 5.5)

$$a = 1 - 2.b$$

$$b = \frac{\sqrt{2} - 2}{2(\cos\omega_c - 1)}$$

pois:

$$a + 2.b = 1$$

$$\sum_n h(n) = 1$$

$$a + 2.b.\cos\omega_c = \sqrt{2}/2$$

$$H(e^{j\omega_c}) = \sqrt{2}/2$$

Quando se deseja projetar um filtro de ordem superior a 3x3 (5x5, 7x7 etc), chega-se a um sistema de equações com múltiplas soluções, de forma que os coeficientes da resposta impulsional não ficam unicamente definidos. Para que se obtenha uma solução única e otimizada, deve-se impor condições adicionais ao projeto do filtro. Restrições típicas são: atenuação mínima na banda de rejeição, oscilação máxima na banda de passagem, largura máxima da banda de transição etc. O método utilizado para o

cálculo dos coeficientes foi o do Janelamento de Hanning [Rabiner/74].

Caso se deseje projetar um filtro passa-altas com frequência de corte ω_c , deve-se primeiro projetar um passa-baixas com frequência de corte $\pi - \omega_c$. Os coeficientes da resposta impulsional passa-altas estão relacionados com os da resposta impulsional passa-baixas pela seguinte relação:

(eq. 5.6)

$$h(n)_{PA(\omega_c)} = (-1)^n \cdot h(n)_{PB(\pi - \omega_c)}$$

Caso se queira um filtro passa-faixa, a sua função de transferência $H_{PF}(e^{j\omega})$ pode ser obtida pela multiplicação das funções de transferência $H_{PB}(e^{j\omega})$ de um filtro passa-baixas pela função de transferência $H_{PA}(e^{j\omega})$ de um filtro passa-altas. Vendo-se de outra maneira, a resposta impulsional $h_{PF}(n)$ do filtro passa-faixa pode ser obtida pela convolução da resposta impulsional $h_{PA}(n)$ de um passa-altas pela resposta impulsional $h_{PB}(n)$ de um passa-baixas. Assim:

(eq. 5.7)

$$H_{PF}(e^{j\omega}) = H_{PA}(e^{j\omega}) \cdot H_{PB}(e^{j\omega})$$

$$h_{PF}(n) = h_{PA}(n) * h_{PB}(n)$$

Projetando-se inicialmente um passa-altas e um passa-baixas de ordem N , consegue-se um passa-faixa de ordem $2N-1$. Os coeficientes $h_{PF}(n)$ do núcleo de convolução podem ser calculados diretamente pela equação 5.7. Um resultado equivalente à convolução com o filtro $h_{PF}(n)$, de ordem $2N-1$, pode ser obtido pela aplicação em cascata de $h_{PA}(n)$ e $h_{PB}(n)$, ambos de ordem N , à uma imagem inicial. Evita-se, com isto, a necessidade de se calcular os coeficientes de $h_{PF}(n)$. No mais, a aplicação em cascata de dois filtros de ordem N é normalmente mais rápida que a aplicação de um filtro de ordem $2n-1$, pois o tempo de processamento aumenta muito com a ordem do filtro.

Projetado o filtro unidimensional, pode-se estendê-lo para o caso bidimensional pelo método da resposta impulsional separável. Assim, para um filtro unidimensional com resposta impulsional $h(n)$, tem-se o seguinte filtro bidimensional:

(eq. 5.8)

$$h(n_1, n_2) = h(n_1) \cdot h(n_2)$$

Desta forma, o elemento $n(x, y)$ do núcleo de convolução pode ser dado por:

(eq. 5.9)

$$n(x, y) = h(x, y) = h(x) \cdot h(y)$$

Assim, para o caso do filtro de ordem 3 projetado, com $h(-1)=b$, $h(0)=a$ e $h(1)=b$, o núcleo de convolução equivalente pode ser expresso por:

b^2	$a \cdot b$	b^2
$a \cdot b$	a^2	$a \cdot b$
b^2	$a \cdot b$	b^2

fig. 5.2 - Núcleo obtido pelo método da resposta impulsional variável

Os núcleos de convolução são operadores lineares, que podem ser aplicados em cascata sobre a mesma imagem. Caso o resultado da aplicação de um filtro não se mostre satisfatório, o mesmo filtro poderá ser reaplicado à imagem processada. Desta maneira, pode-se alcançar um melhor resultado, evitando-se usar um filtro de ordem maior. Vendo-se por outro lado, qualquer função de transferência escolhida para um filtro pode ser decomposta em seus pares de pólos no plano \mathcal{Z} . Cada par de pólos dá origem a um núcleo de convolução 3×3 [Oppenheim/75, Rabiner/74]. Assim, vê-se que os núcleos de convolução 3×3 se mostram satisfatórios e eficientes na maior parte das aplicações. No mais, núcleos de convolução 3×3 são facilmente implementados em hardware, o que reduz consideravelmente o tempo de processamento, possibilitando, inclusive, aplicações em tempo real [Castanho/90].

A seguir são apresentadas as máscaras de convolução implementadas, com suas respectivas aplicações.

5.1.2) AGUDIZAÇÃO DE IMAGENS

A agudização de imagens tem por objetivo melhorar a visibilidade de contornos e bordas que se apresentem com baixo contraste na imagem original. O uso desta técnica se faz necessário, já que o processo de degradação de imagens sempre envolve borramento ou perda de contraste. Nas imagens agudizadas, as transições entre diferentes regiões se tornam mais nítidas e a percepção dos detalhes é substancialmente aumentada. O maior limitante para o processo de agudização de imagens é que este sempre produzirá o efeito indesejável de diminuir a relação sinal/ruído da imagem processada [Mascarenhas/89].

Como as bordas, contornos e transições abruptas estão relacionadas com as componentes de alta frequência da transformada de Fourier de uma imagem, a agudização pode ser obtida pela amplificação destas componentes, ou mesmo pela atenuação das componentes de baixa frequência. Estas operações equivalem à aplicação de um filtro passa-altas à imagem. Por outro lado, o borramento de uma imagem sempre está relacionado com operações de média ou integração, o que sugere que a agudização possa ser obtida por uma operação de diferenciação. Este é o caso dos operadores de Sobel e do Laplaciano, que são detalhados a seguir.

5.1.2.1) FILTRO PASSA-ALTAS

Como foi visto, a aplicação de um filtro passa-altas amplifica as componentes de alta frequência da transformada de Fourier da imagem, produzindo o efeito de agudização. O pacote de processamento digital de imagens permite o projeto e a aplicação de um filtro FIR fase linear às imagens, bastando para isto que se especifiquem a frequência de corte e a ordem desejadas para o filtro. Escolhidos os parâmetros do filtro, é feito um janelamento de Hanning, para se obter os elementos do núcleo de convolução, como já foi exposto em 5.1.

Um filtro de ordem superior normalmente apresenta melhores especificações que um de ordem inferior. Estas especificações são, por exemplo: maior atenuação na banda de rejeição ou mesmo pequena ondulação na banda de passagem. No entanto, quanto maior for a ordem do filtro, maior será o tempo de processamento, sem que se consiga resultados muito superiores. A aplicação do núcleo passa-altas 3x3 se mostrou muito adequada na maior parte das imagens processadas. Como o tempo de processamento do núcleo 3x3 é relativamente baixo, é possível um processo interativo para a escolha da frequência de corte mais adequada para o filtro.

5.1.2.2) LAPLACIANO

A operação de diferenciação de uma imagem, em oposição à integração, que é um processo de borramento, promove a agudização de uma imagem. Um operador diferencial bem adequado à uma função bidimensional é o Laplaciano. Assim, para uma imagem $f(x,y)$, o Laplaciano $\nabla^2 f(x,y)$ é definido como:

(eq.5.10)

$$\nabla^2 f(x,y) = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

No caso de uma imagem digital, com valores discretos para x e y , o Laplaciano pode ser definido a partir das diferenças finitas da imagem, assim:

(eq. 5.11)

$$\begin{aligned}\nabla^2 f(i,j) &= \Delta x^2 f(i,j) + \Delta y^2 f(i,j) \\ &= f(i+1,j) + f(i-1,j) + f(i,j+1) + \\ &\quad f(i,j-1) + 4.f(i,j)\end{aligned}$$

Analisando-se o operador acima, vê-se que sua aplicação é equivalente à convolução com um núcleo 3x3, com os seguintes elementos:

0	1	0
1	-4	1
0	1	0

fig. 5.3 - Operador Laplaciano.

A aplicação do Laplaciano é simples e rápida, evitando-se a demorada fase de projeto de um filtro passa-altas. Em várias situações, os resultados obtidos por este operador se adequaram às necessidades.

5.1.2.3) OPERADORES DE SOBEL

Uma outra maneira de se obter a diferenciação de uma imagem é pela aplicação do operador vetorial gradiente. Assim, para uma imagem $f(x,y)$, o vetor gradiente será dado por:

(eq. 5.12)

$$G [f(x,y)] = \begin{bmatrix} \partial f / \partial x \\ \partial f / \partial y \end{bmatrix} = \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix}$$

O módulo da função vetorial gradiente é dado por :

(eq. 5.13)

$$G [f(x,y)] = \left(\Delta x^2 + \Delta y^2 \right)^{1/2}$$

$$\cong |\Delta x| + |\Delta y|$$

No caso de uma imagem digital - portanto uma função bidimensional discreta - , aproximam-se as diferenciais pelas primeiras diferenças. Assim:

(eq. 5.14)

$$\Delta x \cong \left(f(i+1,j-1) + 2 \cdot f(i+1,j) + f(i+1,j+1) \right)$$

$$- \left(f(i-1,j-1) + 2 \cdot f(i-1,j) + f(i-1,j+1) \right)$$

$$\Delta y \cong \left(f(i-1,j+1) + 2 \cdot f(i,j+1) + f(i+1,j+1) \right)$$

$$- \left(f(i-1,j-1) + 2 \cdot f(i,j-1) + f(i+1,j-1) \right)$$

O cálculo de Δx e Δy pode ser feito pela aplicação dos operadores de Sobel, que são os seguintes núcleos de convolução:

-1	-2	-1
0	0	0
1	2	1

-1	0	1
-2	0	2
-1	0	1

fig. 5.4 - Operadores de Sobel.

Os módulos dos resultados da aplicação destes dois núcleos de convolução devem ser somados, de forma a se obter a imagem final agudizada.

5.1.3) SUAUIZACÃO DE IMAGENS

Muitos tipos de ruído podem estar presentes em uma imagem digital. As fontes de ruído podem ser, por exemplo, o sistema de amostragem ou mesmo o canal de transmissão. A suavização de imagens tem por objetivo minimizar os efeitos destes ruídos na imagem final. Caso a suavização seja aplicada indiscriminadamente, ocorrerá o efeito indesejável de borramento de linhas e contornos, com conseqüente perda de resolução. Há um compromisso entre suavização e borramento. O ideal é que se consiga suavizar uma imagem sem que se perca a definição de seus contornos.

O ruído está normalmente associado a componentes de alta frequência da transformada de Fourier de uma imagem. Desta forma,

a suavização pode ser obtida pela atenuação destas componentes de alta frequência e/ou pela amplificação das componentes de baixa frequência, ou melhor, pela aplicação de um filtro passa-baixas à imagem. Um outro aspecto importante é que o processo de suavização e borramento de imagens está sempre relacionado com operações de média ou integração - o que contrasta com a agudização, que está relacionada com a diferenciação. O filtro do valor médio e o do valor mediano se valem desta propriedade para atingirem a suavização de imagens.

5.1.3.1) FILTRO PASSA-BAIXA

Os filtros passa-baixas atenuam as componentes de alta frequência da transformada de Fourier de uma imagem, produzindo o efeito de suavização, com conseqüente redução do ruído. Da mesma maneira que nos filtros passa-altas (veja item 5.2.1), o pacote implementado permite o projeto e a aplicação de um filtro FIR fase linear. Escolhidas a frequência de corte e a ordem do filtro, é feito um janelamento de Hanning, para que se obtenha os elementos do núcleo de convolução. Valem aqui as mesmas considerações feitas para a escolha da ordem de um filtro passa-altas. A escolha de uma frequência de corte muito baixa poderá provocar o borramento da imagem.

5.1.3.2) FILTRO DO VALOR MÉDIO

As operações de média ou integração tendem a promover a suavização ou mesmo o borramento de uma imagem. A aplicação do

filtro do valor médio tenta tirar proveito desta propriedade. Esta operação consiste em substituir o nível de cinza de cada pixel da imagem pela média dos níveis de cinza dos pixels de uma vizinhança em torno deste pixel. Supondo uma vizinhança quadrada 3x3 em torno de cada pixel da imagem inicial $f(x,y)$, a imagem suavizada $g(x,y)$ é dada por:

(eq. 5.15)

$$g(x,y) = 1/9 \sum_{i=-1}^1 \sum_{j=-1}^1 f(x+i,y+j)$$

Esta operação pode ser obtida pela convolução da imagem $f(x,y)$ com o seguinte núcleo de convolução:

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

fig. 5.5 - Filtro do valor médio.

A aplicação do filtro do valor médio é rápida, eficiente e intuitiva. A escolha de uma vizinhança quadrada maior (4x4, 5x5 etc) promoverá uma suavização mais severa da imagem. O pacote de processamento digital de imagens implementado permite a aplicação

deste filtro a uma vizinhança quadrada de qualquer tamanho. Caso se opte pelo uso de um núcleo de ordem mais elevada, deve-se ter cuidado para não se obter uma imagem excessivamente borrada.

5.1.3.3) FILTRO DO VALOR MEDIANO

A aplicação do filtro do valor mediano consiste em substituir o nível de cinza de cada pixel da imagem, pela mediana dos níveis de cinza de uma vizinhança em torno deste pixel. Deve-se tomar os pixels desta vizinhança e colocá-los em ordem crescente de nível de cinza. Supondo uma vizinhança 3x3 em torno de um pixel, o quinto elemento do conjunto ordenado, formado a partir dos pixels desta vizinhança, é a mediana deste conjunto. O pacote permite a aplicação do filtro do valor mediano a uma vizinhança quadrada de qualquer tamanho.

O filtro do valor mediano se mostrou, na maior parte das aplicações, superior ao filtro do valor médio. Para um mesmo tamanho de vizinhança, este filtro promoveu melhor ou igual suavização do que o filtro do valor médio, produzindo, no entanto, menor borramento na imagem processada, isto é, a imagem foi melhor suavizada, sem que se comprometesse a definição das bordas e contornos.

5.2) MANIPULAÇÃO DE HISTOGRAMA.

Dada uma imagem $f(x,y)$, define-se como histograma de níveis de cinza de f , a função $r(n)$, que denota a frequência relativa com que o nível de cinza n aparece na imagem $f(x,y)$. No caso de uma imagem digital, n é uma variável discreta e o valor de $r(n)$ é dado por:

(eq. 5.16)

$$r(n) = N_n / N_t.$$

Onde N_n é o número de ocorrências do nível de cinza n na imagem e N_t é o número total de pixels da imagem. O seguinte algoritmo resume o cálculo do histograma da imagem $f(x,y)$:

```
VAR Histograma matriz [0..número_níveis_cinza] de inteiro
PARA y = 1 , número_linhas FAÇA:
  PARA x = 1 , número_colunas FAÇA:
    Histograma(f(x,y)) := Histograma (f(x,y)) + 1;
  FIM;
FIM;
PARA nível_cinza = 1 , número_níveis_cinza FAÇA:
  Histograma(nível_cinza) := Histograma(nível_cinza)
                          / número_de_pixels;
FIM;
```

O histograma de níveis de cinza carrega muita informação útil ao processo de segmentação, que é uma etapa necessária antes da colorização de uma imagem. Como a distribuição dos níveis de cinza de cada objeto da cena tende a ser normal e, no caso mais geral, há um bom contraste entre os objetos da cena, é comum se encontrar histogramas de níveis de cinza multimodais. Moda, em uma distribuição de frequência, é o valor da variável que corresponde a um máximo local. A segmentação dos objetos da cena pode ser feita a partir desta informação sobre o nível de cinza da moda de cada objeto, como será discutido no item 6.2.

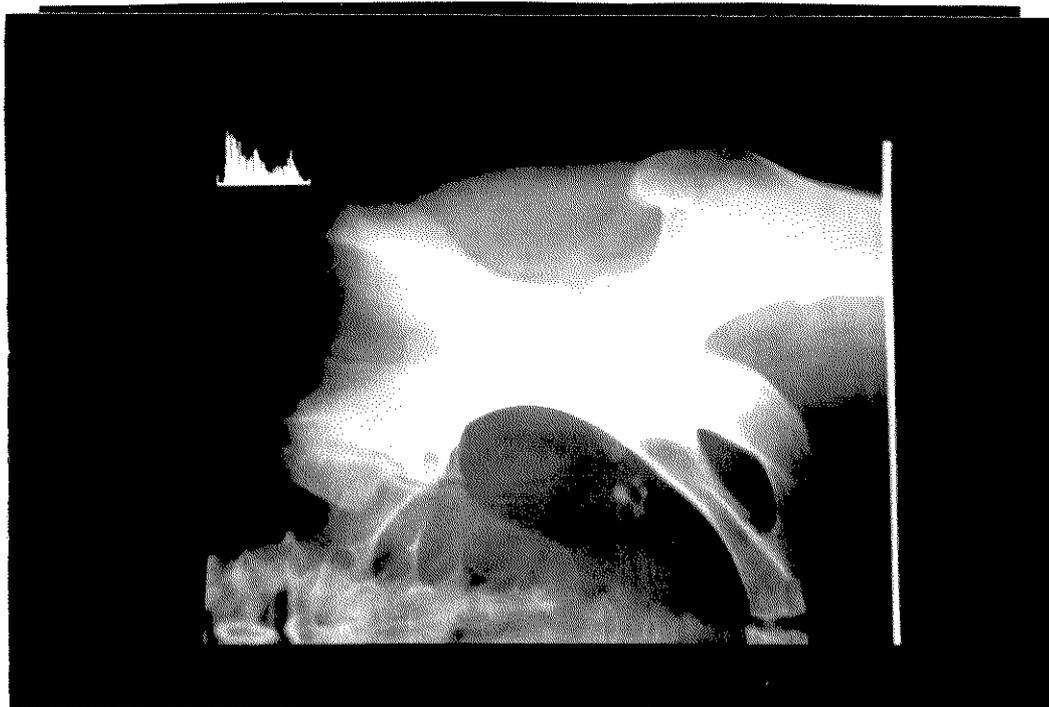


foto 5.1 - Radiografia da articulação coxo-femural, com seu respectivo histograma de níveis de cinza.

Em várias situações se faz necessária a modificação de um histograma de níveis de cinza de uma imagem. As imagens digitais são normalmente amostradas nas mais diversas condições de iluminação. Caso se queira medir certas propriedades da imagem, que estejam relacionadas com níveis de cinza, deve-se normalizar as imagens, de forma a se ter um histograma padrão, previamente especificado. No caso da colorização de imagens médicas, como RX, tomografias, ecografias etc., deve-se padronizar um tipo de mapeamento de nível de cinza para cor. Este mapeamento padronizado facilita a interpretação da imagem, pois se consegue uma correspondência direta do nível de cinza com alguma propriedade que se quer medir. Para isto, é necessário que o histograma de níveis de cinza das imagens digitalizadas no sistema estejam normalizados em um histograma padrão.

A modificação de histograma pode ser usada também para a correção de possíveis erros de quantização na amostragem [Rosenfeld/82]. A técnica de equalização de histograma, na qual se procura atingir uma distribuição uniforme para os níveis de cinza, é usada como forma de minimizar este erro de quantização. No mais, sempre que houver necessidade de mudança de contraste na imagem, otimização do uso da escala de cinza, ou mesmo mudança do brilho da cena, pode-se aplicar um mapeamento de níveis de cinza. A escolha cuidadosa de um mapeamento permite a modificação adequada do histograma de níveis de cinza da imagem.

5.2.1) MAPEAMENTO DE NÍVEIS DE CINZA.

A maneira mais direta de se mudar o histograma de níveis de cinza de uma imagem é a partir da aplicação de uma transformação T , que pode ser linear ou não linear. A transformação T é um mapeamento de nível de cinza para nível de cinza. Assim, dado o nível de cinza da imagem inicial n , tem-se que o nível de cinza transformado m da imagem será:

(eq. 5.17)

$$m = T(n)$$

Dado que o nível de cinza n assume valores no intervalo $[n_i, n_f]$, a transformação T deve satisfazer as seguintes condições básicas:

1) $T(n)$ deve ser monotonicamente crescente no intervalo $[n_i, n_f]$.

2) $n_i \leq T(n) \leq n_f$ para $n_i \leq n \leq n_f$.

A primeira condição garante que a ordenação inicial do preto para o branco será mantida. A segunda condição garante que o mapeamento será feito para um nível de cinza consistente.

Uma aplicação direta do mapeamento de níveis de cinza é a transformação de escala. Um problema típico resultante do

processo de digitalização é a concentração dos níveis de cinza da imagem digitalizada em um sub-intervalo $[a,b]$. Como os níveis de cinza podem assumir valores no intervalo $[n_i, n_f]$, o aumento de contraste pode ser obtido a partir da seguinte transformação:

(eq. 5.18)

$$m = \begin{cases} \frac{(n_f - n_i) \cdot (n - a)}{(b - a)} + n_i & \text{para } a \leq n \leq b \\ n_i & \text{para } n < a \\ n_f & \text{para } n > b \end{cases}$$

No caso específico do hardware usado na implementação deste trabalho, os níveis de cinza podem variar no intervalo $[0,255]$. No entanto, por mais que se tomasse cuidado com a iluminação durante a digitalização, conseguia-se, na média, imagens cujos níveis de cinza variavam no intervalo $[8-190]$. Desta forma, uma transformação de escala de cinza era sempre necessária.

Os limites para a expansão de histograma podem ser encontrados automaticamente, através da análise do próprio histograma $r(n)$ da imagem. Um critério possível de ser adotado é o da eliminação dos pontos extremos. Utiliza-se como nível de cinza inicial (a) para a expansão, o menor nível de cinza para o qual:

$$\sum_0^a r(n) \geq 1\% \quad (\text{eq.5.19})$$

Como nível de cinza final (b) para a expansão, toma-se o menor nível de cinza para o qual:

$$\sum_0^b r(n) \geq 99\% \quad (\text{eq.5.20})$$

Os valores a e b calculados para a expansão automática podem ser utilizados diretamente na eq. 5.18, para que seja feita a expansão. A escolha das porcentagens 1% e 99% foi empírica e seu uso sempre apresentou bons resultados. O algoritmo implementado permite que quaisquer outros valores sejam utilizados.

De uma maneira geral, a transformação T pode ser qualquer função não linear, como por exemplo: logaritmica, exponencial, quadrática etc. Basta que se faça uma translação e uma multiplicação por um fator de escala adequado, de modo a se garantir que as condições básicas apresentadas sejam satisfeitas. Assim, dada uma função f qualquer, e que $f_{\min} = f(n_i)$ e $f_{\max} = f(n_f)$, a transformação modificada pode ser expressa como:

(eq. 5.21)

$$m = \frac{(nf - ni) \cdot [f(n) - fmin]}{(fmax - fmin)} + ni$$

Uma transformação muito útil é a $\log(a + n)$, onde a é uma constante. A sua aplicação melhora sensivelmente a visibilidade dos detalhes de baixo contraste, assim como a definição das bordas e contornos da imagem. A função logarítmica parece refletir a resposta do sistema visual humano [HALL79], o que de certa forma justifica os bons resultados desta transformação.

O pacote implementado permite a aplicação direta de qualquer função, linear ou não linear. A definição da função pode ser feita a partir de sua definição matemática ou através da edição ponto a ponto na tela. Para isto, foi desenvolvido um software interativo, que permite que se veja o gráfico da função durante a edição.

5.2.2) EQUALIZAÇÃO DE HISTOGRAMA.

A equalização do histograma de níveis de cinza pode ser usada como forma de minimização dos efeitos causados pelos erros de quantização cometidos durante a digitalização da imagem. O ideal é que os níveis de cinza da imagem apresentem distribuição uniforme, isto é, todos os níveis de cinza tenham a mesma

probabilidade de ocorrer. Após a aplicação deste algoritmo, um número equivalente de pixels tenderá a ser atribuído a cada nível de quantização da imagem.

Primeiramente, deve-se calcular o histograma $r(n)$ da imagem. A seguir, aplica-se a seguinte transformação à imagem:

(eq. 5.22)

$$m = E(n) = \sum_{k=0}^n r(k)$$

A imagem equalizada apresenta uma distribuição otimizada de níveis de cinza. Os detalhes da imagem são sensivelmente realçados. O pacote implementado permite o cálculo e a aplicação do mapeamento necessário à equalização do histograma de uma imagem.

5.2.3) ESPECIFICAÇÃO DE HISTOGRAMA.

A especificação de um histograma é extremamente importante quando é necessária uma padronização dos histogramas das imagens a serem analisadas. Várias propriedades observadas em uma imagem estão diretamente relacionadas com o nível de cinza. Já que as condições de iluminação durante a digitalização são as mais diversas possíveis, se faz necessária a especificação do histograma, de forma a se garantir uma certa uniformidade nos dados.

Dado que $r_i(n)$ é o histograma da imagem original e que $r_f(n)$ é o histograma desejado para a imagem, a transformação do histograma pode ser obtida da seguinte forma:

1- equalize $r_i(n)$.

$$m = E_i(n) = \sum_{k=0}^n r_i(k)$$

2- obtenha a transformação que equaliza $r_f(n)$.

$$E_f(l) = \sum_{k=0}^l r_f(k)$$

3- aplique a transformação inversa $n = E_f^{-1}(m)$ na imagem inicial equalizada.

A transformação inversa $E_f^{-1}(m)$ nem sempre conduz a valores únicos para n . A solução adotada na implementação foi o ajuste para o valor mais próximo possível. No pacote implementado, o histograma pode ser especificado por meio da definição matemática de uma função ou pela edição direta pelo usuário no terminal. Em ambos os casos, pode-se visualizar interativamente o gráfico da função que está sendo editada.

SEGMENTAÇÃO DE IMAGENS

No processo de segmentação de uma imagem, tem-se por objetivo o agrupamento de pixels em regiões distintas, que estão associadas a estruturas de alto nível. Este agrupamento não é feito de uma maneira arbitrária, procura-se agrupar os pixels de forma a se facilitar algum processamento posterior ao qual será submetida a imagem. O termo estruturas de alto nível é usado, pois as regiões ou segmentos de uma imagem estão sempre associadas a entidades da cena tridimensional, como: nuvens, rosto, água, artérias, ponte etc.

Cada segmento de uma imagem é o conjunto de pixels que foram agrupados segundo algum padrão de homogeneidade. Assim, pixels em diferentes regiões são necessariamente heterogêneos, com respeito a este padrão escolhido. Algumas medidas de homogeneidade são a similaridade e a continuidade. Algoritmos como o de crescimento de regiões e o de subdivisão de área tentam agrupar pixels de acordo com alguma medida de similaridade ou melhor, segundo alguma homogeneidade de textura [Schalkoff/89]. Já outros métodos de segmentação, como a detecção de bordas e o thresholding, se baseiam na descontinuidade ou mudança abrupta de nível de cinza na imagem.

A segmentação possui aplicação direta na colorização de imagens. A colorização sempre se resume a atribuir cores específicas a determinados grupos de pixels ou regiões da imagem.

Com a segmentação, determina-se quais pixels pertencem a cada objeto da imagem. Desta forma, fica viabilizada a colorização diferenciada de cada objeto da cena.

O ideal é que a segmentação seja feita de maneira automática, isto é, escolhe-se a região que se quer colorizar e seus limites são encontrados, valendo-se de alguma propriedade de homogeneidade de regiões. Como se viu, pode-se delimitar uma região pela variação abrupta de nível de cinza, por alguma propriedade estatística dos níveis de cinza da região de interesse, por alguma informação de textura etc.

Em alguns casos, não é possível a segmentação automática de uma região da imagem. Problemas de iluminação e sombreamento da cena, durante a digitalização, podem fazer com que os níveis de cinza do objeto se confundam com os do fundo, dificultando o processo de segmentação. Nessas situações, pode-se delimitar as bordas da região por edição direta na imagem, usando-se cursores, tablet etc. No entanto, a edição direta dos limites da região deve ser evitada, pois produz bordas falsas na imagem, criando descontinuidades, que comprometem o realismo da cena. Nos casos onde a única alternativa é a edição direta, pode-se minimizar os efeitos desagradáveis das bordas falsas com o uso de um algoritmo de Correção de Bordas (veja item 6.1.4). A seguir são detalhados os algoritmos de segmentação implementados no pacote.

6.1) SEGMENTAÇÃO BASEADA EM BORDAS

Como foi discutido, as regiões são definidas de acordo com algum critério de homogeneidade. Partindo-se do princípio de que as regiões possuem níveis de cinza constantes, ou bastante uniformes, pode-se usar esta similaridade dos pixels como forma de se definir um critério para a segmentação. Uma mudança abrupta de nível de cinza (ou melhor uma borda) pode significar o fim de uma região, pois foi detectada uma dissimilaridade.

Na técnica de thresholding, procura-se separar o objeto do fundo a partir da escolha de um nível de cinza adequado para o plano de corte. A segmentação é feita baseada numa informação global de contraste. Caso se tenha um objeto escuro com o fundo claro ou um objeto claro com o fundo escuro, pode-se aplicar com sucesso esta técnica.

No entanto, pode-se ter uma imagem que tenha sido submetida a um gradiente de iluminação. Analisando-se esta imagem, vê-se que o objeto sempre contrasta com o fundo, mas os valores absolutos dos níveis de cinza mudam. Nestes casos, é necessária a análise das diferenças locais de contraste, pois a escolha de um nível de thresholding não segmentará adequadamente a imagem. Os algoritmos de detecção de bordas valem-se destas diferenças locais de nível de cinza como forma de segmentar a região de interesse.

Um dos maiores problemas da segmentação baseada em

deteccão de bordas decorre da necessidade de se ter segmentos definidos a partir de bordas que formem um contorno fechado no plano de imagem. No mais, a borda de uma região deve ser definida a partir de uma curva com apenas 1 pixel de largura. Na prática, o gradiente de iluminação entre o objeto e o fundo pode gerar contornos de vários pixels de largura. Tal problema pode ser solucionado com o uso de algoritmos de afinamento de bordas, que foram introduzidos no pacote implementado e são discutidos em [Loureiro/89].

A seguir serão apresentadas discussões mais detalhadas sobre os algoritmos de Detecção de Bordas (contraste local) e Thresholding (contraste global) implementados no Pacote de Segmentação de Imagens.

6.1.1) THRESHOLDING COM DEFINIÇÃO DE BORDAS

A segmentação por thresholding ou limiar de nível de cinza é o processo mais simples que pode ser implementado, sendo porém, muito eficiente na maioria dos casos. A segmentação é feita a partir de uma informação global de contraste na imagem. Para que o algoritmo funcione adequadamente, espera-se que haja um bom contraste entre o objeto que se quer segmentar e o fundo da cena.

Deve-se, inicialmente, escolher um nível de cinza T , que será o thresholding. A partir desta escolha, definem-se como pertencentes ao fundo da cena, os pixels com nível de cinza menor ou igual a T , e como pertencentes ao objeto que se quer segmentar,

os pixels com nível de cinza maior que a T ou vice-versa. Assim, dada uma imagem inicial $f(x,y)$ e que $\{objeto\}$ e $\{fundo\}$ são níveis de cinza arbitrários, a imagem segmentada $g(x,y)$ será:

(eq. 6.1)

$$g(x,y) = \begin{cases} \{objeto\}, & \text{se } f(x,y) \leq T. \\ \{fundo\}, & \text{caso contrário.} \end{cases}$$

No caso de um objeto mais claro que o fundo, basta que se inverta os valores $\{objeto\}$ e $\{fundo\}$ na equação 6.1, para se segmentar adequadamente a imagem.

Em algumas situações, o fundo da cena pode apresentar simultaneamente pixels mais claros e mais escuros que o objeto. Nestes casos, deve-se definir um intervalo de níveis de cinza $[T_1, T_2]$ para o thresholding. Os pixels cujos níveis de cinza se situem neste intervalo são ditos como pertencentes ao objeto. Assim:

(eq. 6.2)

$$g(x,y) = \begin{cases} \{objeto\}, & \text{se } f(x,y) \in [T_1, T_2]. \\ \{fundo\}, & \text{caso contrário.} \end{cases}$$

A escolha do nível de cinza de thresholding pode ser feita a partir da análise do histograma de níveis de cinza (item 5.2). Em um histograma bimodal, como o apresentado na fig. 6.1, pode-se facilmente separar o objeto do fundo, pela escolha de um thresholding entre as duas modas do histograma [Rosenfeld/82]. No mais, a escolha do melhor nível de thresholding deve ser sempre interativa, isto é, envolve sempre o processo de tentativa e erro. O pacote implementado permite a aplicação de um thresholding rápido, pela modificação direta na look-up-table, o que torna o processo de escolha altamente interativo. Este Thresholding é rápido, pois se evita a modificação do conteúdo da memória de vídeo, que no caso de nossa aplicação envolve a modificação de $480 \times 200 = 96.000$ bytes. A mudança do conteúdo da look-up-table só envolve a atualização de $256 \times 3 = 768$ bytes.

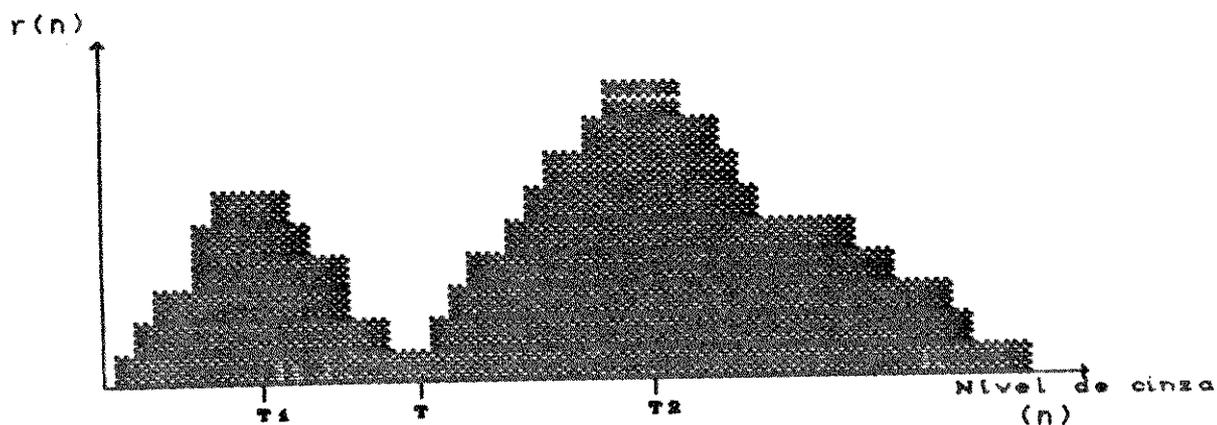


fig 6.1 - Exemplo de histograma de níveis de cinza com distribuição bimodal.

Em uma imagem que apresente um histograma de níveis de cinza bimodal - ou genericamente um histograma que contenha picos e vales significativos - a escolha do nível de cinza T de thresholding pode ser automatizada da seguinte forma:

- 1 - Encontre os dois maiores máximos locais - T_1 e T_2 do histograma $r(n)$ da imagem.
- 2 - Encontre o ponto T (valor de thresholding) no intervalo $[T_1, T_2]$ onde $r(n)$ assume valor mínimo.

NO	N	NE
L	P	L
SO	S	SE

fig. 6.2 - Vizinhos do ponto P analisados pelos algoritmos de definição e afinamento de bordas.

A segmentação por thresholding é mais eficiente quando o objeto, que se quer segmentar, guarda um bom contraste com o fundo. A distribuição dos níveis de cinza dos pixels do objeto e do fundo devem ser unimodais e com diferentes médias, o que

raramente ocorre. No entanto, este é um processo rápido e interativo e pode ser usado em conjunto com outros métodos de segmentação, principalmente quando se quer incluir a propriedade de continuidade de regiões.

As imagens geradas pelo algoritmo de thresholding são imagens binárias, isto é, possuem apenas dois níveis de cinza (objeto e fundo). Desta maneira, a definição das bordas de um segmento se resume em encontrar os pixels que estejam na interface do dois níveis de cinza. O algoritmo de definição de bordas desenvolvido usa esta propriedade para extrair as bordas de uma imagem binária. Muitas vezes, as bordas geradas não possuem a espessura mínima possível (1 pixel). Com a finalidade de corrigir este problema, foi desenvolvido um algoritmo de afinamento de bordas [Loureiro/89]. Assim, o processo completo de segmentação por thresholding envolve também a determinação e o afinamento das bordas, de forma que a borda final gerada é necessariamente contínua e possui apenas 1 pixel de espessura. Os algoritmos desenvolvidos levam em conta a relação de cada pixel da imagem com seus oito vizinhos (fig. 6.2). A seguir, serão apresentados os algoritmos desenvolvidos:

PROCEDIMENTO DEFINIÇÃO_DE_BORDAS;

PARA todo pixel P da imagem FAÇA:

SE P={objeto} FAÇA:

SE (N={fundo} ou S={fundo} ou
L={fundo} ou O={fundo}) FAÇA:

P := {borda}

SENÃO P := {fundo};

FIM;

FIM;

FIM;

No algoritmo de afinamento de bordas, serão usadas as seguintes nomenclaturas: P, quando $P=\{borda\}$, e \bar{P} , quando $P=\{fundo\}$. Assim as condições referenciadas no algoritmo de afinamento de bordas serão:

(eq. 6.3)

Condição1 := (NE ou \bar{L}) e (NO ou \bar{O}) e (\bar{S} ou (L e SE) ou (SO e O))

Condição2 := (SO ou \bar{O}) e (SE ou \bar{L}) e (\bar{N} ou (O e NO) ou (NE e L))

Condição3 := (SE ou \bar{S}) e (NE ou \bar{N}) e (\bar{O} ou (S e SO) ou (NO e N))

Condição4 := (NO ou \bar{N}) e (SO ou \bar{S}) e (\bar{L} ou (N e NE) ou (SE e S))

PROCEDIMENTO AFINAMENTO_DE_BORDAS;

REPITA ATÉ que nenhum ponto tenha sido eliminado:

PARA todo pixel da imagem FAÇA:

SE (P=objeto) e N=objeto) e condição1)

N := (fundo);

FIM;

PARA todo pixel da imagem FAÇA:

SE (P=objeto) e S=objeto) e condição2)

S := (fundo);

FIM;

PARA todo pixel da imagem FAÇA:

SE (P=objeto) e L=objeto) e condição3)

L := (fundo);

FIM;

PARA todo pixel da imagem FAÇA:

SE (P=objeto) e O=objeto) e condição4)

O := (fundo);

FIM;

FIM;

FIM;

6.1.2) DETECÇÃO DE BORDAS

Nos algoritmos de thresholding, a segmentação é feita a partir de uma informação global de contraste. Faz-se, se possível, uma análise do histograma de níveis de cinza da imagem. Já na

deteccção de bordas, focaliza-se o interesse em variações locais de contraste. A determinação dos limites da região é feita, mesmo que haja diferentes níveis de iluminação na cena. Ainda assim, é necessário que objeto e fundo guardem sempre - pelo menos localmente - uma diferença mensurável de nível de cinza.

A detecccção de bordas pode ser feita pela aplicação de operadores diferenciais [Gonzalez/87]. Estes operadores assumem valores significativos em regiões onde ocorram grande disparidade de níveis de cinza. Estas regiões de transição, onde há variação abrupta de nível de cinza, representam bordas em potencial. A primeira derivada de uma imagem indica a presença de uma borda, pois esta assume valor nulo nas regiões com nível de cinza constante e um valor constante nas regiões de mudança de nível de cinza. O sinal da segunda derivada indica se a transição ocorrida foi do claro para o escuro ou do escuro para o claro.

Estes operadores diferenciais podem ser aplicados à imagem, com o uso de núcleos de convolução (veja item 5.1). O operador gradiente ou operador de Sobel (veja item 5.1.2.3) pode ser usado para a obtenção da primeira derivada da imagem. O operador Laplaciano (veja item 5.1.2.2), que é um operador de 2ª ordem, pode ser usado como forma de se ter uma noção da 2ª derivada da imagem.

Outros núcleos de convolução usados na segmentação por bordas são normalmente operadores orientados à detecccção de linhas. Na fig. 6.2, são mostrados núcleos de convolução que detectam

linhas horizontais, verticais, inclinadas a 45° e inclinadas a 135° respectivamente.

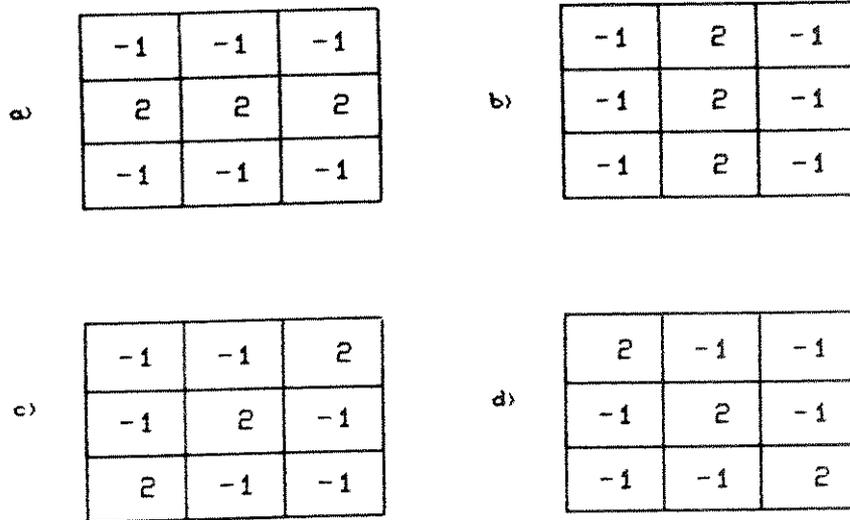


fig 6.3 - Núcleos de convolução para detecção de linhas.
a) horizontal. b) vertical. c) 45° . d) 135° .

Uma vez geradas as bordas, é feita uma aglomeração das informações, de forma a se detectar os limites da região. Esta aglomeração pode ser feita a partir de um algoritmo de crescimento de regiões, mesmo as bordas geradas não sejam contínuas. Pode-se, para isto, sobrepor à imagem inicial as bordas geradas pela convolução com um dos núcleos da fig. 6.3. Nesta nova imagem, deve-se colocar uma semente de crescimento dentro dos limites definidos pelas bordas desta região. O algoritmo de crescimento de regiões pode se então executado, até que estas bordas sejam encontradas. Os algoritmos de crescimento de regiões serão melhor detalhados em 6.2.1.

6.1.3) EDIÇÃO DIRETA NA IMAGEM

Em algumas situações, devido possivelmente a problemas de iluminação durante a digitalização, a imagem, a ser segmentada, apresenta muito pouco contraste. Como nestas situações as bordas e contornos da imagem estão mascarados pelo baixo contraste, os algoritmos de segmentação automática se mostram bastante ineficientes.

Nestes casos, se faz necessária a edição direta das bordas da região que se quer segmentar. Estas bordas servem de guia para a delimitação da região de interesse. O pacote implementado permite a edição direta de bordas, com o auxílio de cursores na imagem, ou com o uso de tablet. Editada a borda, pode-se utilizar o algoritmo de crescimento de regiões (veja item 6.2.1) com o seguinte predicado de homogeneidade H :

(eq. 6.4)

$$H = \begin{cases} [\text{Falso}] & \text{se encontrou a borda editada;} \\ [\text{Verdadeiro}] & \text{caso contrário;} \end{cases}$$

Vale ressaltar, que a edição de bordas pode provocar o aparecimento de falsos contornos na imagem colorizada, o que prejudica o realismo que se busca na colorização. Os algoritmos de Correção de Bordas podem minimizar estes efeitos prejudiciais.

6.1.4) CORREÇÃO DE BORDAS

De acordo com o exposto no item 6.1.3, a edição direta na imagem provoca o aparecimento de falsos contornos na imagem colorizada. Pode-se reduzir o efeito desagradável das bordas falsas, pelo uso de um algoritmo severo de suavização de imagens. Como foi discutido no item 5.13, uma suavização severa provoca o borrimento das bordas e contornos da imagem. Com o uso de um operador de borrimento, as bordas falsas perdem a sua definição, confundindo-se com o contínuo da imagem.

No caso de uma imagem colorida, os operadores de suavização e borrimento devem ser aplicados a cada uma das componentes da soma $R(x,y)$, $G(x,y)$ e $B(x,y)$. Desta forma, a imagem deve estar especificada em sua forma segregada. Caso a imagem esteja especificada na forma integrada (imagem índice $I(x,y)$ e look-up-table), deve-se fazer, inicialmente, a separação das três imagens componentes da soma, com o seguinte algoritmo:

```
PARA x = 1 , Número_Colunas FAÇA:
  PARA y = 1 , Número_Linhas FAÇA:
    R(x,y) = LUT_R( I(x,y) ) * 4;
    G(x,y) = LUT_G( I(x,y) ) * 4;
    B(x,y) = LUT_B( I(x,y) ) * 4;
  FIM;
```

O multiplicador 4 é necessário, pois os valores da look-up-table variam no intervalo $[0,63]$ (6 bits), enquanto que a memória de vídeo possui 8 planos de memória (valores no intervalo $[0,255]$). Os vetores LUT_R(), LUT_G() e LUT_B() indicam o conteúdo da look-up-table, nos índices especificados, para cada uma de suas componentes (R, G e B).

Uma vez separadas as imagens, aplica-se a cada uma delas um algoritmo de suavização (média da vizinhança por exemplo). As três imagens borradas devem ser comprimidas por um algoritmo de escolha da look-up-table ótima, de modo a ser possível a sua visualização no equipamento disponível.

6.2) SEGMENTAÇÃO BASEADA EM TEXTURA

Na segmentação baseada em textura, procura-se agrupar pixels em regiões ou segmentos, segundo algum padrão de similaridade. Uma região que possua uma certa similaridade entre seus pixels, possui uma homogeneidade na sua textura. A segmentação da imagem se torna possível, pois os limites de uma região podem ser determinados a partir de uma mudança significativa no padrão de textura medido.

Para que se possa medir quantitativamente uma textura, é necessário que se tenha um modelo que caracterize a sua evolução espacial. No entanto, uma definição única e universalmente aceita de textura não é encontrada ainda na literatura. Textura é uma

noção muitas vezes intuitiva e está fortemente relacionada com sensações visuais e táteis. A textura pode muitas vezes referir-se unicamente a uma variação percebida de nível de cinza em uma região. No entanto, através da textura, pode-se distinguir facilmente padrões como o de deserto, floresta, algodão, metal, madeira, concreto etc. Do ponto de vista do Processamento Digital de Imagens, a textura pode ser definida como sendo um arranjo repetitivo de um certo padrão básico ou melhor, por uma certa homogeneidade estatística nos pixels de uma região.

Grande parte dos padrões de textura podem ser diferenciados através da análise de distribuições de probabilidade estatística de primeira e segunda ordem. A média (μ) e a mediana (m) dos níveis de cinza de uma região são exemplos de estatísticas de primeira ordem, que dão uma medida global da luminosidade da região analisada. Da mesma forma, a variância (σ^2) e o desvio padrão (σ) dão uma idéia do contraste global da região. Estas estatísticas de primeira ordem podem ser tiradas diretamente do histograma de níveis de cinza ($r(n)$) da região. Assim:

$$a) \mu = 1 / N \sum_n n \cdot r(n) \quad (\text{eq. 6.5})$$

$$b) \sigma^2 = 1 / n \sum_n (n - \mu)^2 \cdot r(n)$$

$$\text{onde } N = \sum_n r(n)$$

c) A mediana é o nível de cinza n_m tal que metade dos pixels da região são mais claros que n_m e metade mais escuros.

A partir destas medidas estatísticas de primeira ordem, pode-se calcular algumas medidas de homogeneidade de regiões. Uma medida, por exemplo, de similaridade dos pixels da região, pode ser a variável sensibilidade S , dada pela eq. 6.6. A variável S assume valor nulo em regiões uniformes e valor unitário em regiões de alto contraste.

$$S = 1 - 1 / (1 + \sigma^2) \quad (\text{eq. 6.6})$$

Embora estas medidas estatísticas de primeira ordem - que partem da análise do histograma de níveis de cinza - dêem uma informação global a respeito da distribuição dos níveis de cinza de uma região, elas não dão nenhuma informação a respeito da maneira pela qual cada nível de cinza se relaciona com seus vizinhos em um padrão de textura. De forma a se quantificar melhor esta relação espacial entre os níveis de cinza, deve-se recorrer a matrizes de co-ocorrência, que são uma medida estatística de segunda ordem.

Considere-se uma relação espacial R que relaciona o pixel (x,y) com o pixel (x^*,y^*) em uma imagem $f(x,y)$. Um exemplo de relação seria: (x^*,y^*) está à direita de (x,y) . Para qualquer par de níveis de cinza s e t , $Q(s,t)$ é definido como sendo o número de pares de pixels (x,y) e (x^*,y^*) para os quais.

(eq. 6.7)

1 - $(x, y) \in R \iff (x^*, y^*)$; os pixels satisfazem a relação R .

2 - $f(x, y) = s$ e $f(x^*, y^*) = t$;

Dado que os níveis de cinza podem assumir valores no intervalo $[0 \dots M]$, a matriz de co-ocorrência C_R - o índice R indica que há uma matriz de co-ocorrência para cada relação R escolhida - é definida como:

(eq. 6.8)

$$C_R(i, j) = \begin{bmatrix} Q(0,0) & Q(0,1) & Q(0,2) & \dots & Q(0,M) \\ Q(1,0) & Q(1,1) & Q(1,2) & \dots & Q(1,M) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ Q(M,0) & Q(M,1) & Q(M,2) & \dots & Q(M,M) \end{bmatrix}$$

A matriz de co-ocorrência $C_R(i, j)$ não fornece diretamente, por meio de um único valor, uma medida quantitativa da textura da região. Algumas medidas empíricas são usadas para prover esta necessidade. A seguir serão listados alguns exemplos destes parâmetros derivados a partir de $C_R(i, j)$.

(eq. 6.9)

$$\text{Contraste} = \sum_i \sum_j (i-j) C_R(i,j)$$

$$\text{Entropia} = \sum_i \sum_j C_R(i,j) \cdot \log [C_R(i,j)]$$

$$\text{Uniformidade} = \sum_i \sum_j C_R(i,j)^2$$

$$\text{Homogeneidade} = \sum_i \sum_j C_R(i,j) / [1 + (i-j)^2]$$

Observa-se das equações 6.9, que o contraste e a entropia assumem valores altos quando há pouca concentração de valores na diagonal principal, isto é, há uma grande dispersão de níveis de cinza na região. Já a uniformidade e a homogeneidade assumem valores altos quando há grande concentração de valores na diagonal principal, isto é, a distribuição dos níveis de cinza na região é bastante uniforme.

Uma outra maneira de se abordar o problema de segmentação de uma região com textura seria pela eliminação da própria textura. Como foi discutido, a textura pode ser entendida como sendo um arranjo repetitivo de um padrão básico. Este arranjo está sempre relacionado com componentes de alta frequência da transformada de Fourier de uma imagem [Schalkoff/89]. Desta forma, a aplicação de um filtro passa-baixas (veja item 5.1.3.1) à

imagem pode reduzir ou mesmo eliminar a textura, ficando a região com um padrão mais uniforme na distribuição dos seus níveis de cinza. Em uma região mais uniforme, a aplicação de um algoritmo de crescimento de regiões, ou mesmo um de subdivisão de área, pode solucionar o problema de segmentação, mesmo com o uso de parâmetros estatísticos de primeira ordem.

Nos algoritmos de segmentação baseada em regiões (ou textura), são sempre definidos predicados de homogeneidade e conectividade para possíveis testes. O predicado H de homogeneidade é definido para a região R como:

(eq. 6.10)

$$H(R) = \begin{cases} [\text{Verdadeiro}] & \text{se } R \text{ for homogênea;} \\ [\text{Falso}] & \text{caso contrário;} \end{cases}$$

O predicado H pode ser avaliado a partir de uma medida estatística de primeira ou segunda ordem, ou mesmo a partir de combinações com outros métodos de segmentação, como será melhor detalhado nos algoritmos de crescimento de regiões e subdivisão de área.

O predicado C de conectividade dos pixel P_1 e P_2 na região R é definido como:

(eq. 6.11)

$$C(P_1, P_2) = \begin{cases} [\text{Verdadeiro}] & \text{se os pixel } P_1 \text{ e } P_2 \text{ são} \\ & \text{conectados e pertencem a } \mathbb{R}; \\ [\text{Falso}] & \text{caso contrário;} \end{cases}$$

Um par de pixels P_1 e P_2 são ditos conectados em \mathbb{R} se:

- 1 - Existir uma seqüência (P_1, \dots, P_2) , tal que os pixels adjacentes na seqüência sejam conectados e
- 2- Todos os pixels da seqüência estejam em \mathbb{R} .

Na implementação do pacote de segmentação, foi sempre analisada a conectividade dos quatro vizinhos, como pode ser visto na fig. 6.4.

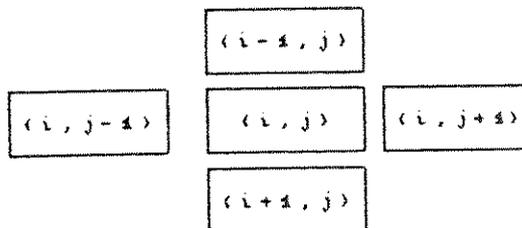


fig.6.4 - Os quatro vizinhos conectados do pixel (i, j) serão testados para inclusão ou não na região.

6.2.1) CRESCIMENTO DE REGIÕES

Na segmentação por crescimento de regiões, vale-se da propriedade de uniformidade de regiões, isto é, os pixels para serem incluídos na região devem satisfazer a algum predicado de homogeneidade H . A escolha do predicado é um dos parâmetros de entrada do algoritmo. A escolha adequada deste predicado permite a correta segmentação da imagem.

O processo de segmentação tem início com a escolha de um pixel representativo para a região de interesse. Este pixel servirá como ponto inicial ou semente para o processo de crescimento. O valor do predicado H para a semente servirá como parâmetro de comparação para inclusão ou não de novos pixels à região.

A partir da escolha da semente, é criada uma lista, cujo primeiro elemento é a própria semente. Retira-se cada elemento da lista e testam-se seus quatro vizinhos (veja fig. 6.4) para inclusão ou não na região, já que estes quatro vizinhos satisfazem o predicado C de conectividade apresentado na eq. 6.11. Um pixel é incluído na região, caso o predicado H de homogeneidade seja satisfeito (exemplos de predicados usados serão dados a seguir). No caso de um pixel ser incluído na região, seus parâmetros são também incluídos na lista, para que seus vizinhos sejam também testados. Os pixels já visitados são marcados, para que não sejam testados novamente. O processo continua, até que a lista fique

vazia. O seguinte algoritmo resume a segmentação por crescimento de regiões:

PROCEDIMENTO CRESCIMENTO_DE_REGIÕES;

INÍCIO

Escolha um pixel que servirá como semente;

Marque a semente com o atributo já-testado;

Inclua a semente na região;

Inclua a semente na lista; (primeiro elemento)

SE lista não estiver vazia FAÇA:

Retire o primeiro elemento da lista;

PARA cada um de seus quatro vizinhos FAÇA:

SE o atributo do vizinho for não-testado FAÇA:

Marque o vizinho com o atributo já-testado;

SE H for verdadeiro FAÇA:

inclua o vizinho na lista;

inclua o vizinho na região;

FIM;

FIM;

FIM;

FIM;

FIM;

A seguir, serão listados alguns exemplos de possíveis predicados $H(R)$.

1 - H será verdadeiro, caso a diferença entre o nível de cinza do pixel analisado e o nível de cinza da semente (ou a média de uma vizinhança em torno da semente) não seja superior à amplitude de variação escolhida para a região.

2 - H será verdadeiro, caso a condição 1 seja verdadeira e um thresholding não tenha sido alcançado.

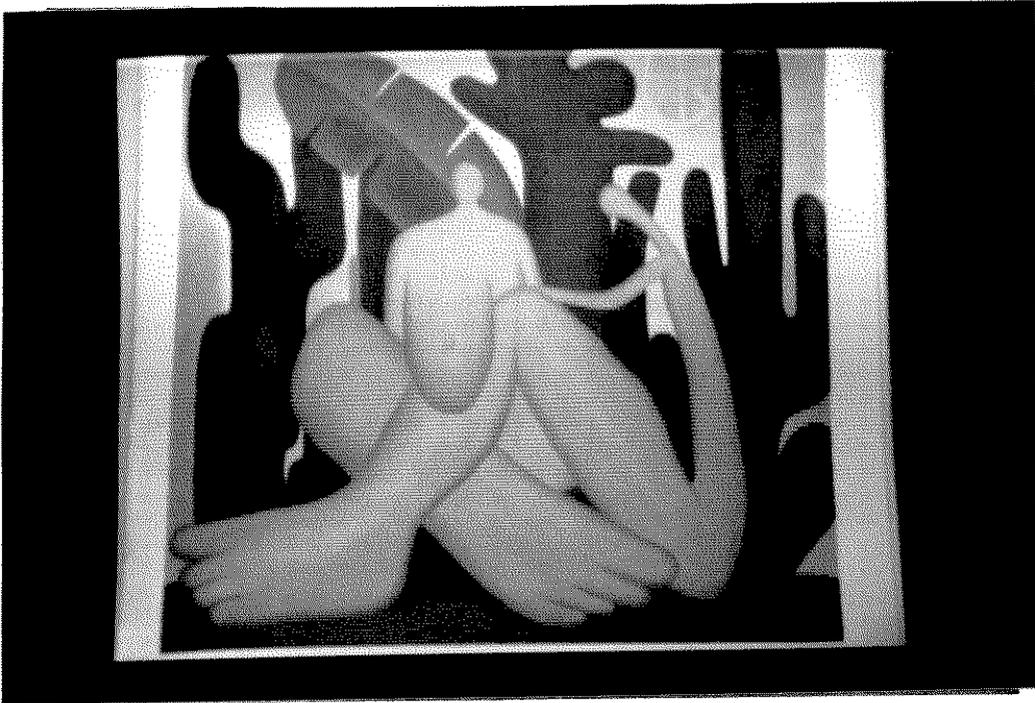
3 - H será verdadeiro, caso a condição 1 seja verdadeira e um contorno falso (previamente editado) não tenha sido encontrado.

4 - H será verdadeiro, caso a condição 1 seja verdadeira e uma borda não tenha sido previamente detectada.

5 - H será verdadeiro, caso a diferença entre o nível de cinza da semente e a média acumulada dos pixels já incluídos na região não seja superior à amplitude de variação escolhida para a região.

6 - H será verdadeiro, caso alguma medida empírica de estatística de segunda ordem (contraste, entropia, uniformidade, homogeneidade etc), referente a alguma matriz de co-ocorrência escolhida, calculada para uma vizinhança em torno do pixel analisado, esteja dentro de um intervalo esperado. O limite esperado depende da mesma medida estatística calculada em uma região em torno da semente.

a)



b)

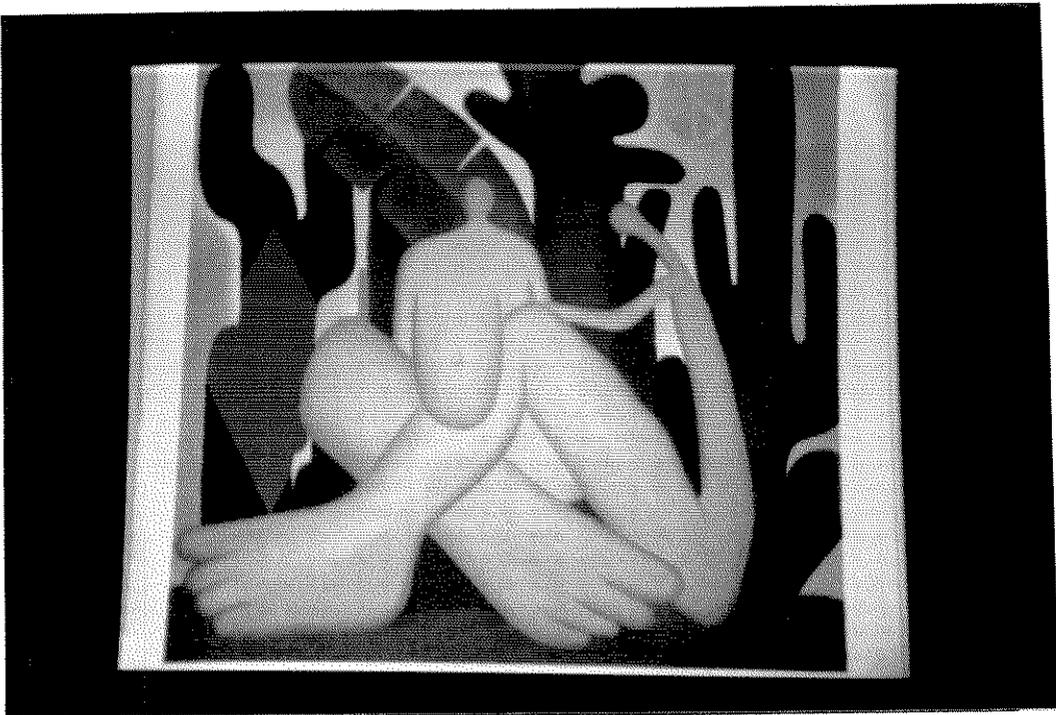


Foto 6.1 - Tarsila do Amaral, "Antropofagia", óleo sobre tela.
Como critério de parada, foi usado o predicado 5 apresentado. a) Notar semente em estágio inicial de crescimento. b) Semente em estágio avançado de crescimento.

Pode-se montar outros predicados ou usar uma combinação destes citados. A escolha do melhor predicado está relacionada com o tipo de região que se quer segmentar.

6.2.2) SUBDIVISÃO DE ÁREA

A segmentação por crescimento de regiões é um procedimento do pixel para a região, enquanto que na subdivisão de área parte-se da imagem para o pixel. O algoritmo subdivide regiões não-homogêneas em regiões menores, de forma que o teste de homogeneidade seja reaplicado a estas regiões

Inicialmente escolhe-se uma amostra significativa da região que se quer segmentar (uma amostra 16×16 por exemplo) e calcula-se uma medida estatística E_0 que caracterize esta amostra. Pode-se, a partir deste ponto, iniciar o processo de subdivisão de área, que é um algoritmo recursivo, que subdividirá as regiões, em busca de sub-regiões de interesse. A subdivisão é feita pelo processo clássico de quadtree.

O algoritmo é composto de duas fases distintas: a subdivisão e a união. Inicialmente na fase de subdivisão, o processo é chamado tendo como parâmetro inicial a primeira partição da imagem, ou melhor, toda a tela. A partição é então testada quanto a sua homogeneidade, com o auxílio de um predicado H . Se a partição não for uniforme (predicado H for falso), o programa, que é um algoritmo recursivo, é chamado outras quatro

vezes, tendo como parâmetros cada uma das quatro subdivisões da partição (veja fig. 6.5). No caso da partição ser uniforme, é calculada a medida estatística E_p que caracterize a partição. Se E_p e E_a (da amostra) satisfizerem o predicado S_p de semelhança, a região será dita de interesse, pois é estatisticamente semelhante à amostra. As partições de interesse são marcadas, para que sejam analisadas na fase de união. Após a possível marcação, o programa retorna.

Um exemplo implementado, para a avaliação da medida estatística E_a em uma amostra característica da imagem, é o cálculo da média μ_a e do desvio padrão δ_a desta amostra. Também, para cada partição, deve-se calcular a medida estatística E_p , isto é, os parâmetros μ_a e δ_a da partição. O predicado H é avaliado da seguinte forma:

(eq. 6.12)

$$H = \begin{cases} [\text{Verdadeiro}] & \text{se 95\% dos pixels da partição} \\ & \text{estiverem no intervalo } \mu_a \pm \delta_a. \\ [\text{Falso}] & \text{caso contrário.} \end{cases}$$

A escolha da porcentagem 95% foi arbitrária. O seu valor pode ser alterado, pois é uma das entradas do programa. O predicado S é avaliado da seguinte forma:

(eq. 6.13)

$$S[E_a, E_p] = \begin{cases} [\text{Verdadeiro}] & \text{se } \mu_p \text{ estiver no intervalo} \\ & [\mu_a - \delta_a, \mu_a + \delta_a]. \\ [\text{Falso}] & \text{caso contrário.} \end{cases}$$

Outros predicados H e S podem ser criados, a partir do cálculo de outras medidas estatísticas. A escolha dos melhores predicados está associada ao tipo de região que se quer segmentar.

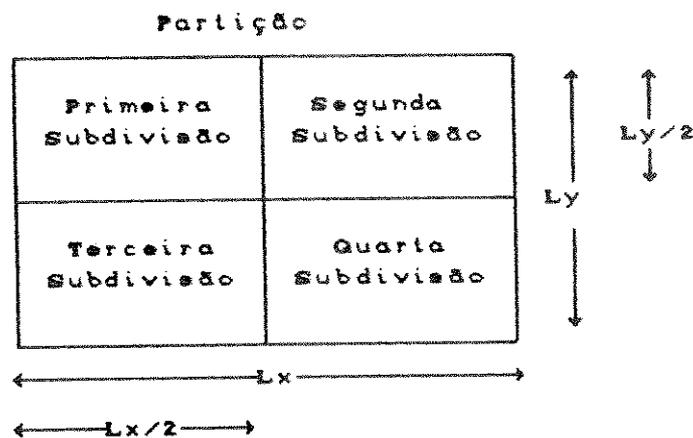


fig.6.5 - Ilustração de uma partição com suas quatro subdivisões.

O programa subdividirá as partições, em busca de regiões uniformes, até chegar ao nível de um pixel - caso seja necessário. Neste ponto, têm-se várias partições de interesse, dos mais variados tamanhos, armazenadas na estrutura de dados na forma de quadtree. No exemplo da fig.6.5., se deseja segmentar a região

hachurada. Para isto, são separadas 16 partições e marcadas 5 partições de interesse

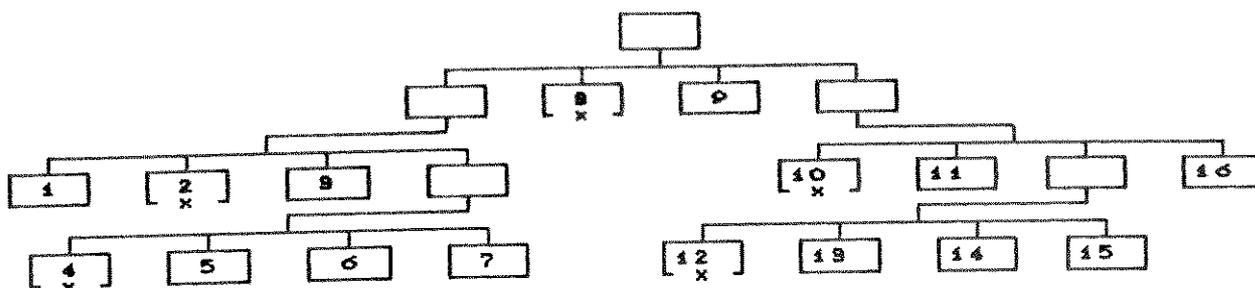
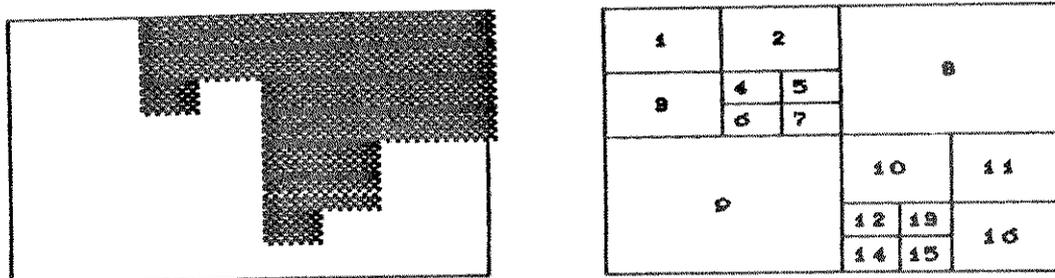


fig.6.6 - Ilustração da subdivisão das partições. As partições de interesse são marcadas na quadtree.

Na fase de união, percorre-se a quadtree, de forma a se aglutinar as regiões de interesse, que satisfaçam ao critério de continuidade com a semente escolhida, isto é, exista pelo menos um pixel, um em cada região, para o qual o predicado C de continuidade com a semente seja verdadeiro. O algoritmo abaixo ilustra o processo de subdivisão de área:

PROCEDIMENTO PROGRAMA_PRINCIPAL;

- 1- Escolha uma amostra significativa para a região;
- 2- Calcule E_a da amostra;
- 3- SUBDIVISÃO (Toda_a_imagem);
- 4- Aglomere as regiões de interesse que satisfaçam o predicado C; (fase de união)

FIM;

PROCEDIMENTO SUBDIVISÃO (Partição);

SE H(Partição) for Verdadeiro FAÇA:

INÍCIO

1- Calcule E_p da partição;

SE S(E_p, E_a) for Verdadeiro

2- marque a partição com o atributo
[de-interesse];

SENÃO

3- marque a partição com o atributo
[eliminada]

FIM;

SENÃO

INÍCIO

SUBDIVISÃO (Primeira_Subdivisão);

SUBDIVISÃO (Segunda_Subdivisão);

SUBDIVISÃO (Terceira_Subdivisão);

SUBDIVISÃO (Quarta_Subdivisão);

FIM;

FIM;

7) COLORIZAÇÃO DE IMAGENS

No processo de colorização, tem-se por objetivo a atribuição de um mapeamento adequado de nível de cinza para cor, a cada segmento da imagem monocromática. Foram implementados no pacote, vários algoritmos de colorização, de modo que é possível a escolha interativa do mapeamento mais adequado a cada segmento.

Todos os algoritmos que serão detalhados a seguir, foram implementados no sistema, de forma a poderem ser aplicados à imagem inteira ou a um segmento específico de qualquer tamanho. A imagem final colorizada pode ser do tipo comprimida com partição da look-up-table ou especificada através de três imagens com as componentes primárias da soma: $R(x,y)$, $G(x,y)$ e $B(x,y)$, conforme discutido no capítulo 4. Caso a segunda solução seja escolhida, deve-se, ao final da colorização, usar um algoritmo de escolha da LUT ótima, de modo a se poder visualizar a imagem colorizada no equipamento disponível (arquitetura integrada).

A colorização de uma imagem no equipamento disponível, como foi discutido em 4.1, envolve sempre modificações na look-up-table. A LUT possui 256 entradas que servem de endereço para o vetor (R,G,B) da cor especificada. De forma a se padronizar a nomenclatura nos algoritmos que serão apresentados a seguir, usar-se-á o termo $LUT(i)$ para representar o vetor (R,G,B) armazenado na entrada i da LUT (i assume valores no intervalo $[0,255]$). Os termos $LUT_R(i)$, $LUT_G(i)$ e $LUT_B(i)$ se referem aos elementos

individuais constituintes do vetor (R,G,B), respectivamente.

7.1) COLORIZAÇÃO POR PLANOS DE CORTE

A técnica de colorização por planos de corte é, conceitualmente, semelhante à técnica de thresholding discutida no item 6.2.1. A idéia de planos de corte deriva da interpretação geométrica da imagem $f(x,y)$, como sendo uma função bidimensional de níveis de cinza [Gonzales/87]. Supondo que os níveis de cinza possam assumir valores no intervalo $[0,N]$, pode-se definir um plano de corte no nível de cinza n_1 , tal que $n_1 \in [0,N]$. Definem-se, assim, duas regiões distintas: do nível de cinza 0 ao n_1 e do n_1 ao N . O método se resume a atribuir uma cor a cada uma das regiões definidas. Um pixel receberá uma cor, de acordo com sua posição relativa ao plano de corte (acima ou abaixo).

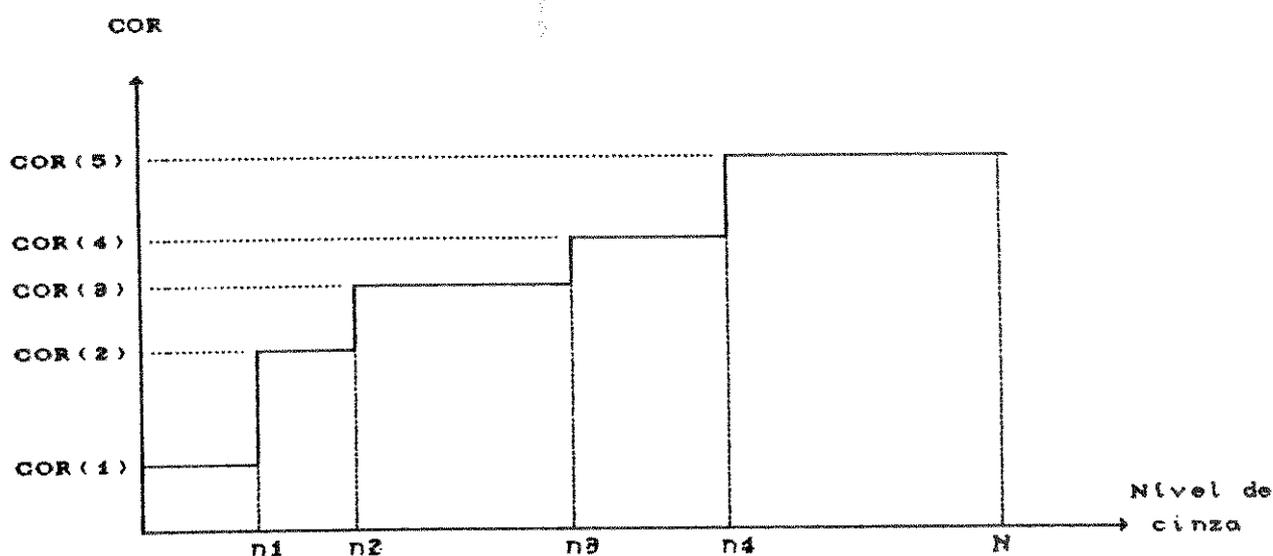
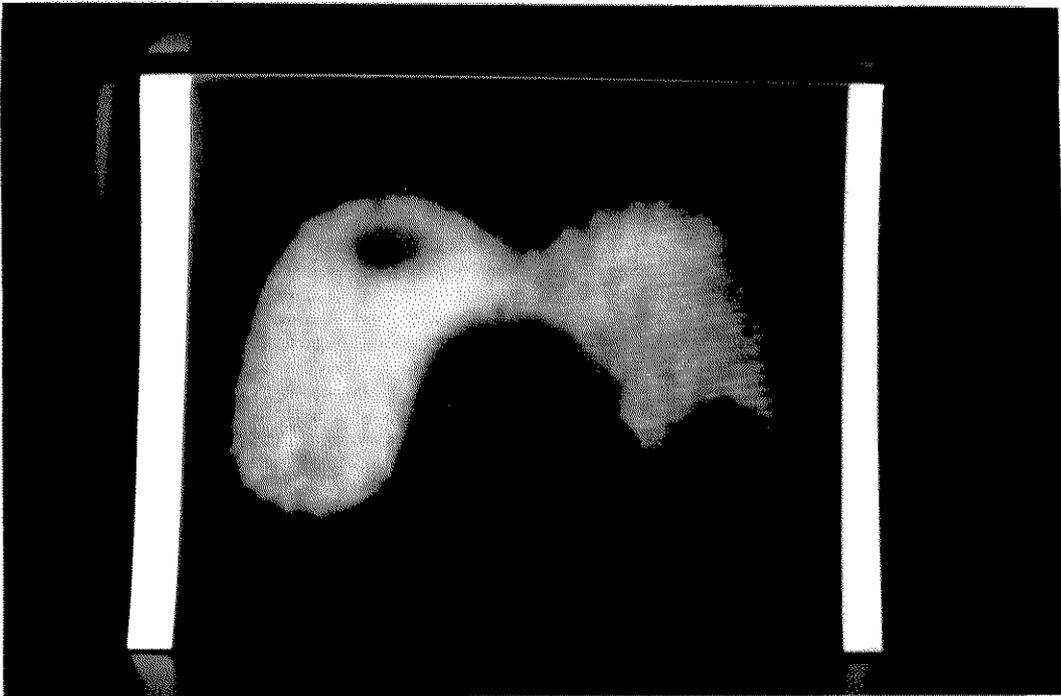


fig. 7.1 - Ilustração do processo de colorização com o uso de 4 planos de corte, nos níveis de cinza n_1 , n_2 , n_3 e n_4 .

a)



b)

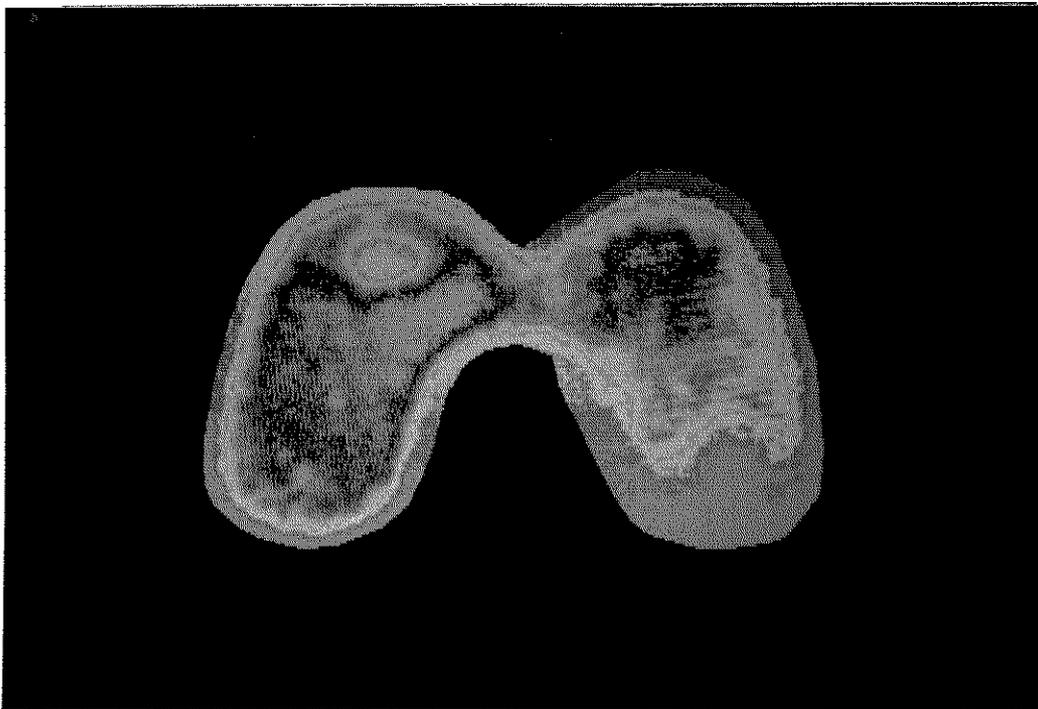


foto 7.1) - a) Imagem radiológica de mapeamento de tireóide, já processada por um algoritmo de expansão de histograma. b) Mesma imagem colorizada com sete planos de corte. Notar a escala de cores utilizada: vermelho/quente, violeta/frio.

O método pode ser generalizado, permitindo-se a definição de L planos de corte, por conseguinte, L+1 regiões (veja fig. 7.1). Atribui-se, da mesma forma, uma cor a cada região. No fragmento de algoritmo apresentado a seguir, que ilustra a colorização por planos de corte, é suposto que o usuário já tenha escolhido L planos de corte, cujos respectivos níveis de cinza estão armazenados no vetor `Nível_Plano`, nas entradas de 1 a L. O usuário também deverá ter escolhido L+1 cores, que serão atribuídas às L+1 regiões definidas pelos L planos. As cores estão armazenadas no vetor `Cores`, nos índices de 1 a L+1.

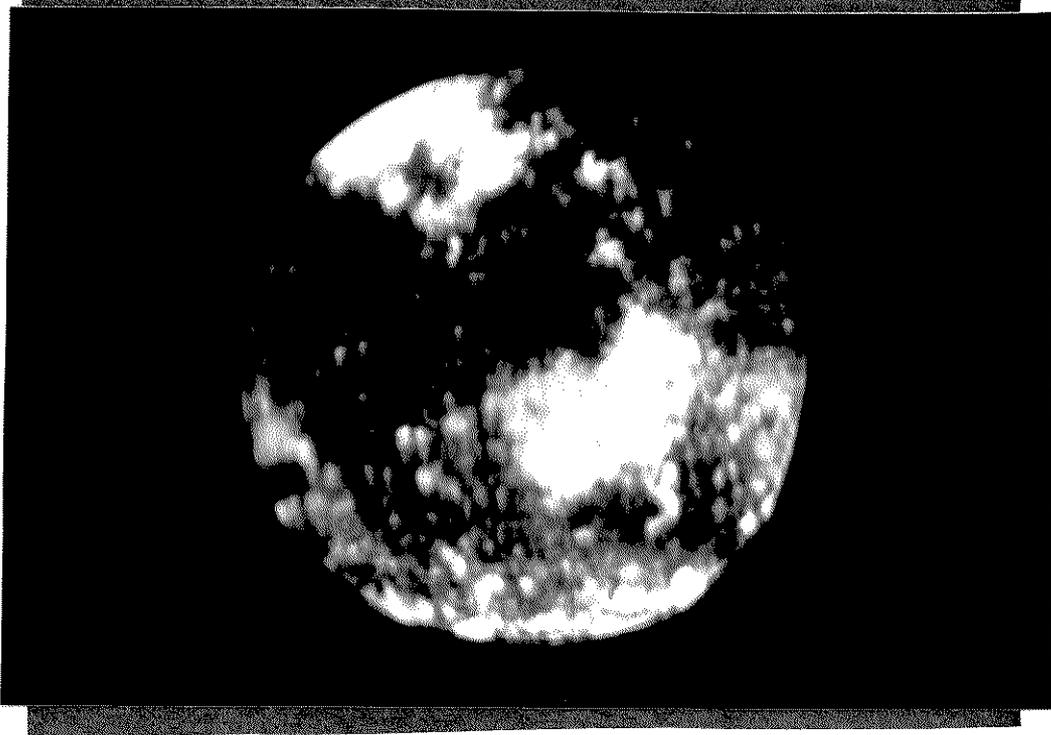
```

Nível_Plano(0) := 0;
Nível_Plano(L+1) := N+1;
PARA plano = 0 , L FAÇA:
    PARA i = Nível_Plano(plano) , Nível_Plano(plano+1) FAÇA:
        LUT(i) = COR(i);

```

Este tipo de colorização é muito interessante, quando usada em imagens cuja noção de profundidade está comprometida pelo baixo contraste dos níveis de cinza. Os planos de corte, com as respectivas regiões por eles definidas, apresentam um efeito semelhante a curvas de nível em uma projeção de um objeto tridimensional. Com isto, a noção de profundidade fica valorizada e o perfil do objeto torna-se mais aparente.

a)



b)

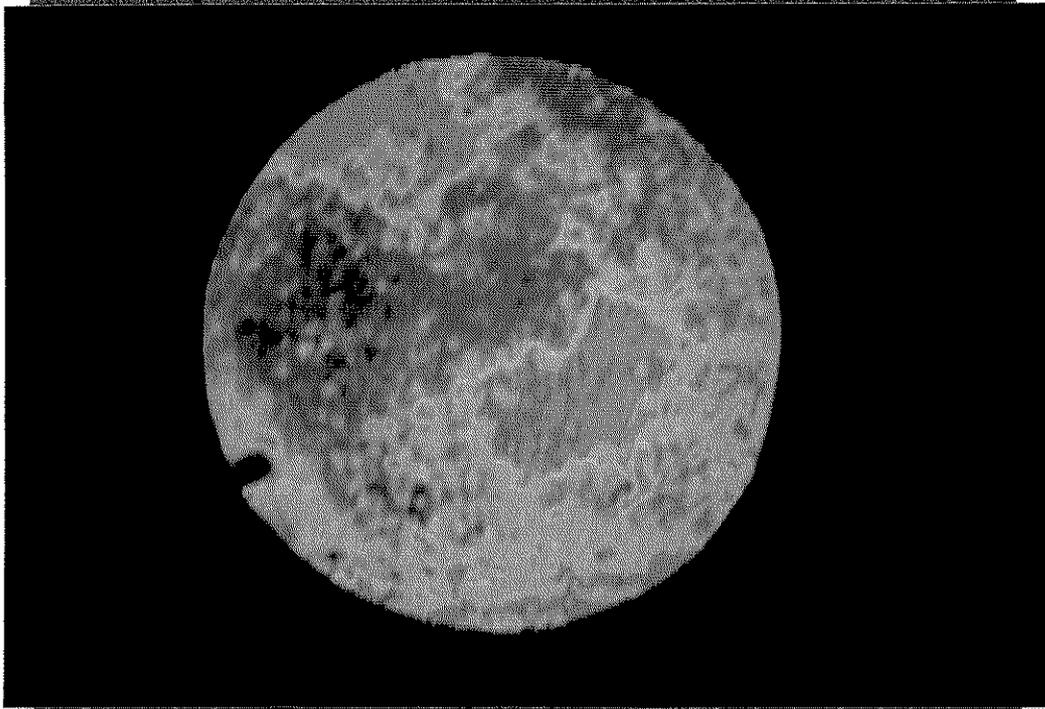


foto 7.2 - a) Tomografia computadorizada de um testemunho de rocha-reservatório. b) Mesma imagem colorizada com 14 planos de corte.

Para que o efeito da valorização de profundidade seja melhor alcançado, é interessante o uso de cores com algum contraste em regiões adjacentes. Um dado prático seria o uso de cores espaçadas, no mínimo, de 30° na variável matiz (H) do modelo HSV. Este espaçamento já apresenta um contraste perceptível. Caso se alterem simultaneamente as variáveis saturação (S) e luminância (V), é possível o uso de um passo menor no matiz ($20 - 25^\circ$).

O uso de cores em seqüência, espaçadas no matiz por algum passo constante, é extremamente interessante, pois reproduz a escala tipo arco-íris, que é muito intuitiva. No mais, a associação dos tons de vermelho (H em torno de zero) aos pixels escuros e dos tons de violeta (H em torno de 300) aos pixels claros, recupera a noção de vermelho/quente e violeta/frio.

7.2) COLORIZAÇÃO POR FUNÇÕES

Nesta técnica de colorização de imagens, utilizam-se três funções $f_R(n)$, $f_G(n)$ e $f_B(n)$, que serão aplicadas ao nível de cinza n da imagem monocromática $f(x,y)$ [Schowengerdt/83]. O conceito envolvido nestas transformações é semelhante ao apresentado no mapeamento de níveis de cinza discutido no item 5.2.2. A diferença reside no fato de que o valor calculado para as funções $f_R(n)$, $f_G(n)$ e $f_B(n)$, para o nível de cinza n , serão armazenados na entrada n da look-up-table. O seguinte algoritmo ilustra este tipo de colorização:

```

PARA  Nível_de_Cinza = 0 , N  FAÇA:
    LUT_R(Nível_de_Cinza) := fR(Nível_de_Cinza);
    LUT_G(Nível_de_Cinza) := fG(Nível_de_Cinza);
    LUT_B(Nível_de_Cinza) := fB(Nível_de_Cinza);

FIM;

```

Pode-se usar qualquer função para se efetuar as transformações $f_R(n)$, $f_G(n)$ e $f_B(n)$. A única restrição está na necessidade de se ter um mapeamento consistente. Para um domínio no intervalo $[0,255]$ (níveis de cinza possíveis no equipamento), o contra-domínio deve estar no intervalo $[0,63]$, isto é, um valor que possa ser armazenado na look-up-table. Em alguns casos, quando a colorização está restrita a um segmento da imagem, o domínio da função será um sub-intervalo do domínio apresentado.

As funções $f_R(n)$, $f_G(n)$ e $f_B(n)$ podem ser funções distintas ou, o que é mais comum, a mesma função transladada. A translação é necessária, pois caso se use a mesma função nos três canais (sem translação), a imagem final continuará sendo monocromática.

Nas figuras 7.2-a, 7.2-b e 7.2-c, são apresentadas as transformações mais utilizadas. A transformação com o uso da função módulo do seno, como pode ser observado na fig. 7.2-a, apresenta pequena variação na vizinhança dos máximos locais e variações rápidas nos mínimos. Com a colocação adequada dos

máximos e mínimos, ajustando-se para isto, tanto a frequência dos senos quanto suas fases iniciais, pode-se conseguir grande variedade de resultados. A transformação de fig. 7.2-b produz uma escala de cores familiar ao usuário, a dizer: os pixels escuros apresentarão um tom de vermelho, os médios um tom de verde e os claros um tom de azul. A maior vantagem da transformação da figura 7.2-c reside no fato de que as cores obtidas estão normalizadas em relação à luminância, isto é:

$$\text{Vermelho} + \text{Verde} + \text{Azul} = \text{Constante} \quad (\text{eq. 7.1})$$

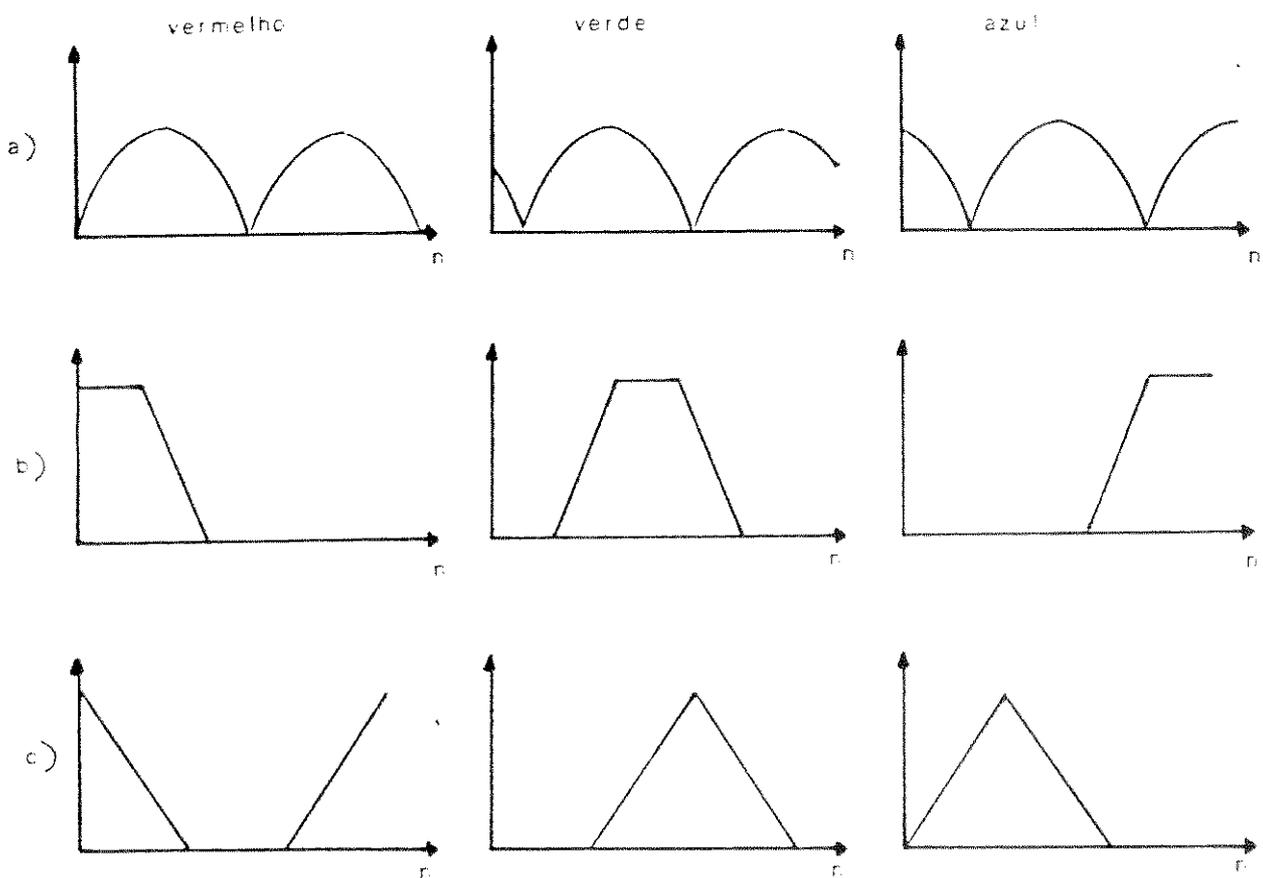


fig. 7.2 - (a) Função módulo de seno. (b) Função com escala cromática (c) Função com luminância normalizada

a)



b)

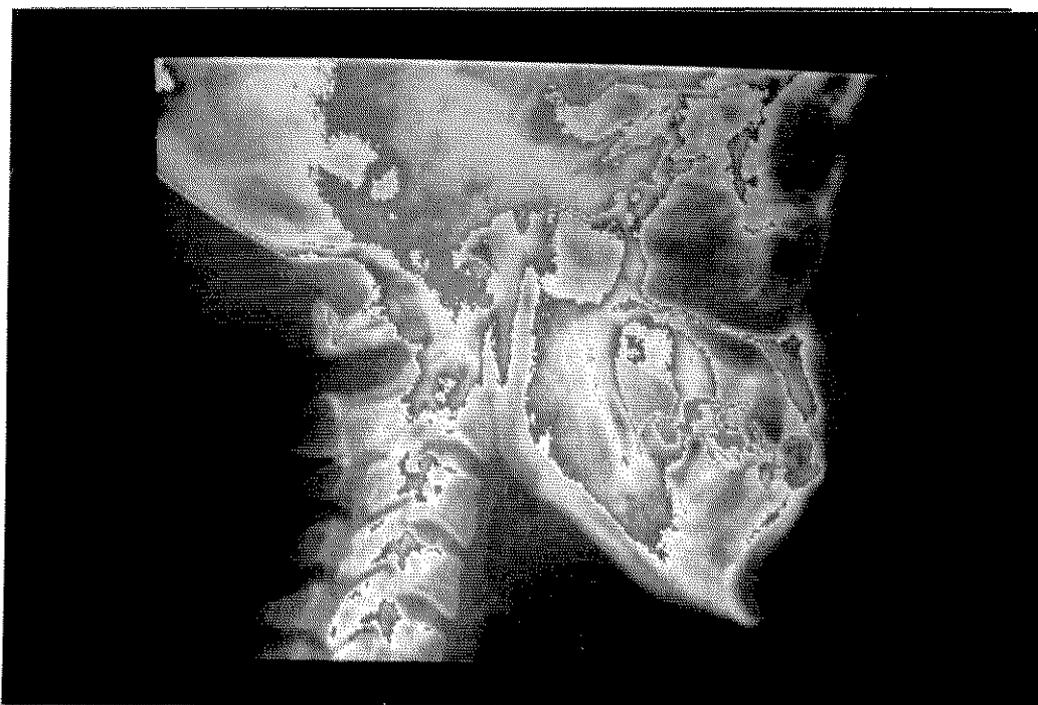


foto 7.3 - a) Radiografia de crânio e coluna cervical, incidência de perfil b) Mesma imagem colorizada com 1 plano de corte. Ao fundo, aplicou-se uma função módulo de seno. À segunda região aplicou-se uma função de luminância normalizada.

a)



b)

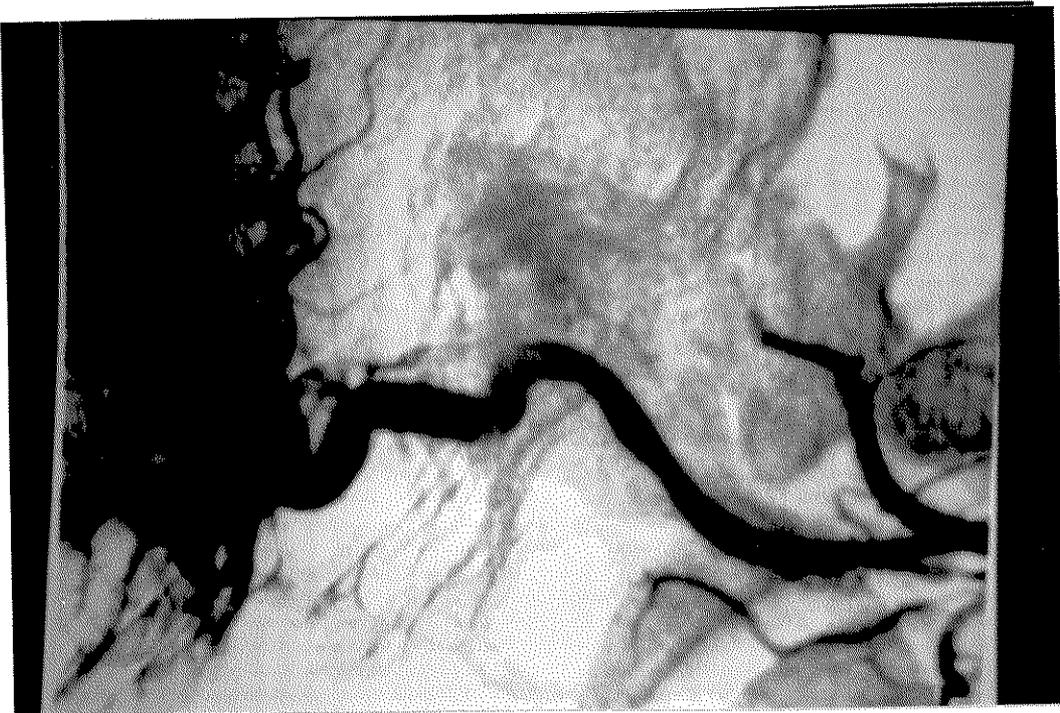


foto 7.4 - a) Ecografia da crossa da artéria aorta, aorta torácica e aorta abdominal. b) Mesma imagem colorizada por uma função de escala cromática.

a)



b)

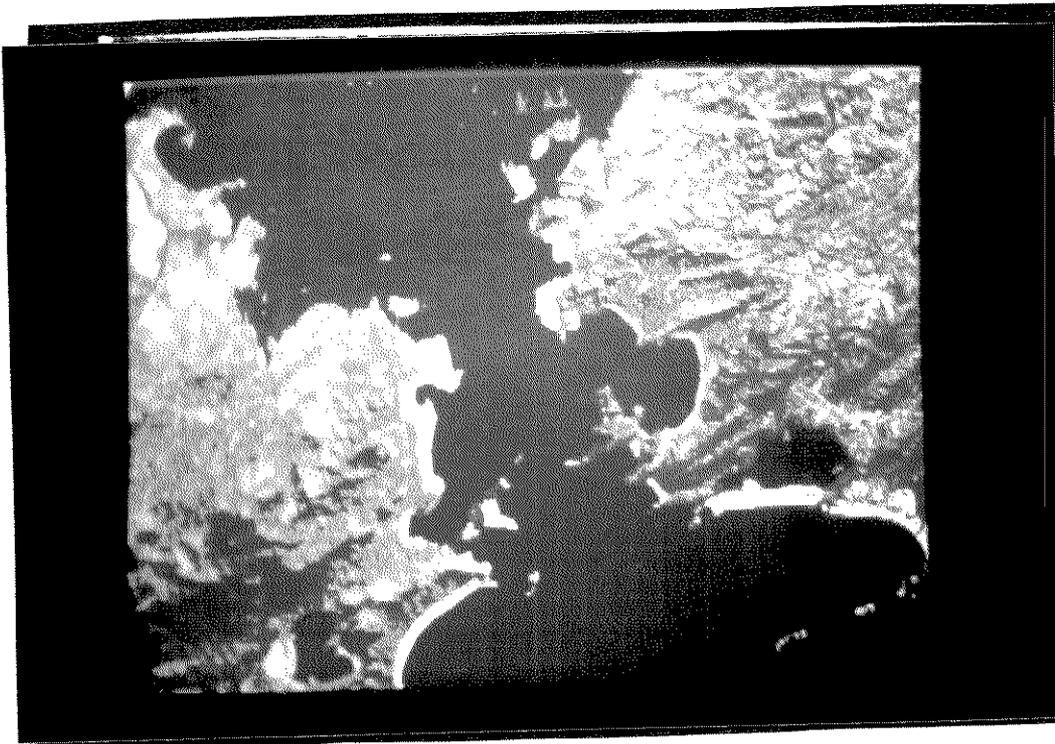


foto 7.5 - a) Imagem aeroespacial da baía da Guanabara. b) Mesma imagem colorizada com dois segmentos. O primeiro segmento, que contém o continente, foi colorizado por uma função seno. O segundo, onde estão as águas, por uma interpolação de luminância.

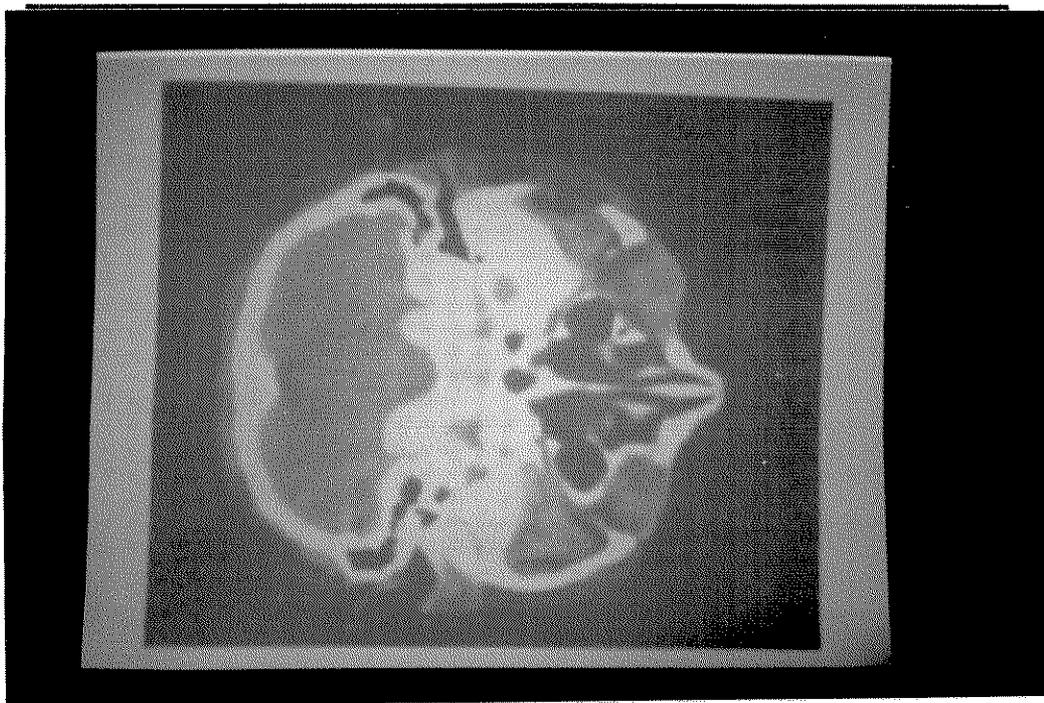
A colorização por transformações é um processo rápido, podendo ser aplicado imediatamente a qualquer imagem. No sistema implementado, tem-se disponível um menu de funções pré-definidas. Uma vez escolhida uma função base, deve-se definir as translações relativas, a serem aplicadas à função base, para se obter as funções $f_R(n)$, $f_G(n)$ e $f_B(n)$. O processo é iterativo, pois pode-se aplicar um número qualquer de funções (bastando que se escolha no menu), até que se encontre a que apresente os melhores resultados na colorização. Outra aplicação interessante de colorização por funções é na padronização de colorizações. Caso se queira ter uma correspondência biunívoca entre nível de cinza e cor, pode-se definir uma transformação padrão a ser aplicada a cada tipo de imagem.

7.3) COLORIZAÇÃO POR INTERPOLAÇÕES NOS NÍVEIS DE CINZA

7.3.1) INTERPOLAÇÃO LINEAR

No processo de interpolação dos níveis de cinza, faz-se uma correspondência linear entre nível de cinza e uma das componentes fundamentais da cor: matiz, saturação e luminância. Escolhe-se qual componente será interpolada e fixam-se as outras duas em um valor desejado. É comum, também, pedir-se os valores inicial e final da componente a ser interpolada, de modo a se ter bem definidos os limites de interpolação.

a)



b)

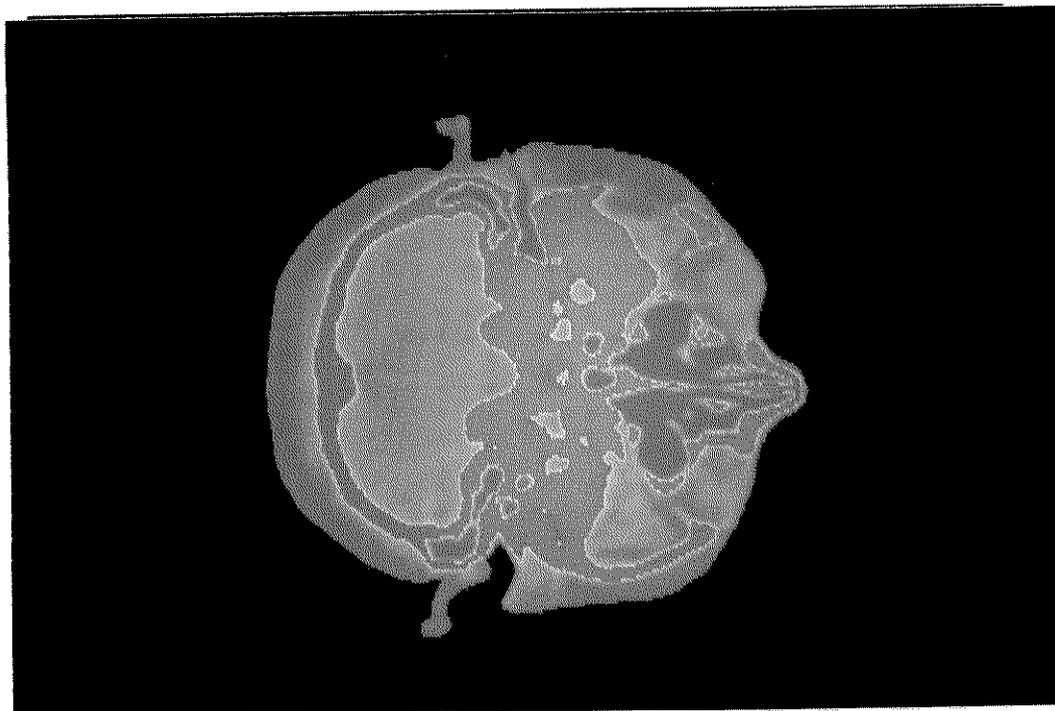


foto 7.6 - a) Tomografia computadorizada do crânio. b) Mesma imagem colorizada com dois planos de corte. A região central foi colorizada com uma interpolação de saturação. Às demais regiões, foram atribuídas cores únicas

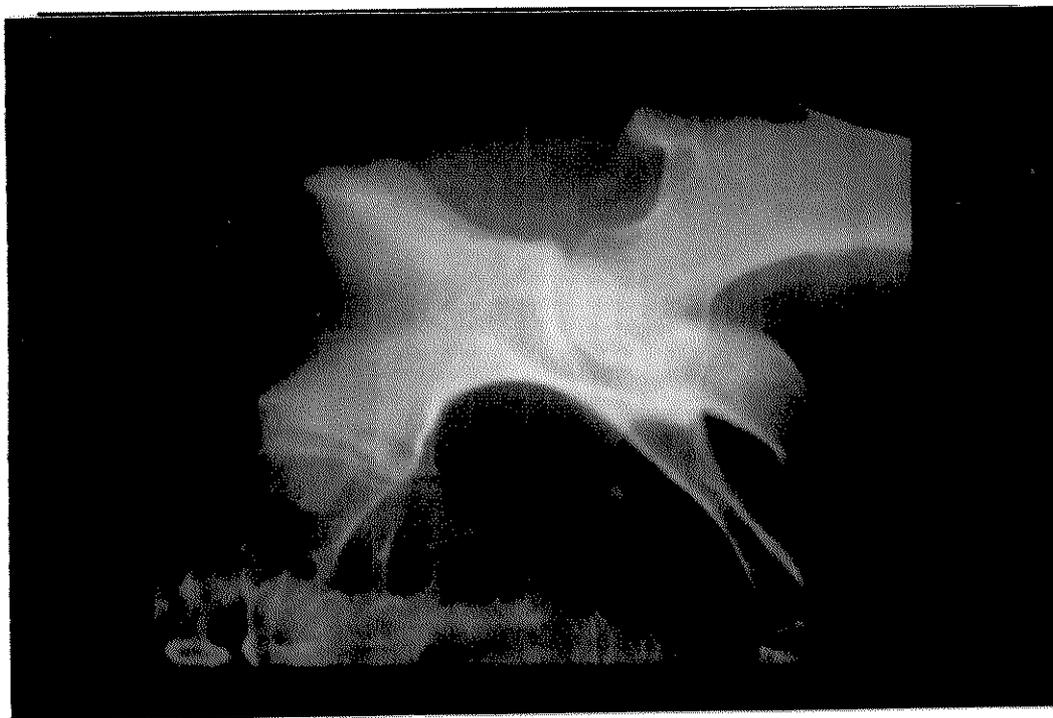


foto 7.7 - Imagem mostrada na foto 5.1 colorizada com uma interpolação de matiz.

Na interpolação de matiz, pode-se conseguir interessantes variações espectrais, como uma gradação tipo arco-íris. Esta é uma colorização muito útil para vários tipos de aplicação, pois esta escala é muito intuitiva para o usuário (vermelho/quente, azul/frio!). No caso da interpolação de saturação, a escala tenderá a ir da cor pura (nos pixels escuros) ao branco (nos pixels mais claros), isto é, obter-se-á um degradê da cor básica escolhida (matiz e luminância). Este tipo de interpolação é muito útil quando se deseja colorizar um objeto onde haja, em alguns pixels, excesso de luminância ou brilho intenso. Este é o caso de uma esfera metálica, ou um capô de um automóvel, refletindo uma fonte de luz, ou mesmo nuvens brancas

contrastando com um céu azulado. No caso da interpolação de luminância, simplesmente se escolhe uma cromaticidade (matiz + saturação) e faz-se uma correspondência linear entre o nível de cinza do pixel e a luminância da cor. Este tipo de interpolação reproduz com boa fidelidade a noção de profundidade e sombreamento, que se encontrava na cena monocromática.

7.3.2) INTERPOLAÇÃO CILÍNDRICA

Com o intuito de se obter um grau ainda maior de realismo na colorização de imagens monocromáticas, desenvolveu-se a técnica de interpolação cilíndrica. Neste algoritmo, se faz a interpolação não só de uma, mas das três componentes fundamentais simultaneamente. Consegue-se, com isto, a reprodução de padrões complexos de cor. Estes padrões ocorrem na maioria das cenas reais. No exemplo de um pôr-do-sol, tem-se um padrão que vai do azul ao amarelo, passando pelo vermelho. O azul tem algum branco diluído e pouca luminância, enquanto o amarelo é um pouco menos saturado, com muito brilho. Consegue-se, também, evitar o aparecimento de contornos em preto, que são comuns em interpolações lineares de luminância. O contorno em preto retira o realismo da imagem, tornando-a mais parecida com um desenho. No mais, melhora-se muito o sombreamento dos objetos da imagem, reproduzindo-se, com mais fidelidade, a noção de profundidade [Fonseca/90].

a)



b)



foto 7.8 - a) Lasar Segall, "Retrato de Mário de Andrade", óleo sobre tela. b) Mesma imagem colorizada por interpolação cilíndrica. Foram definidos 29 segmentos, em cada um foi utilizada uma interpolação adequada.

a)



b)

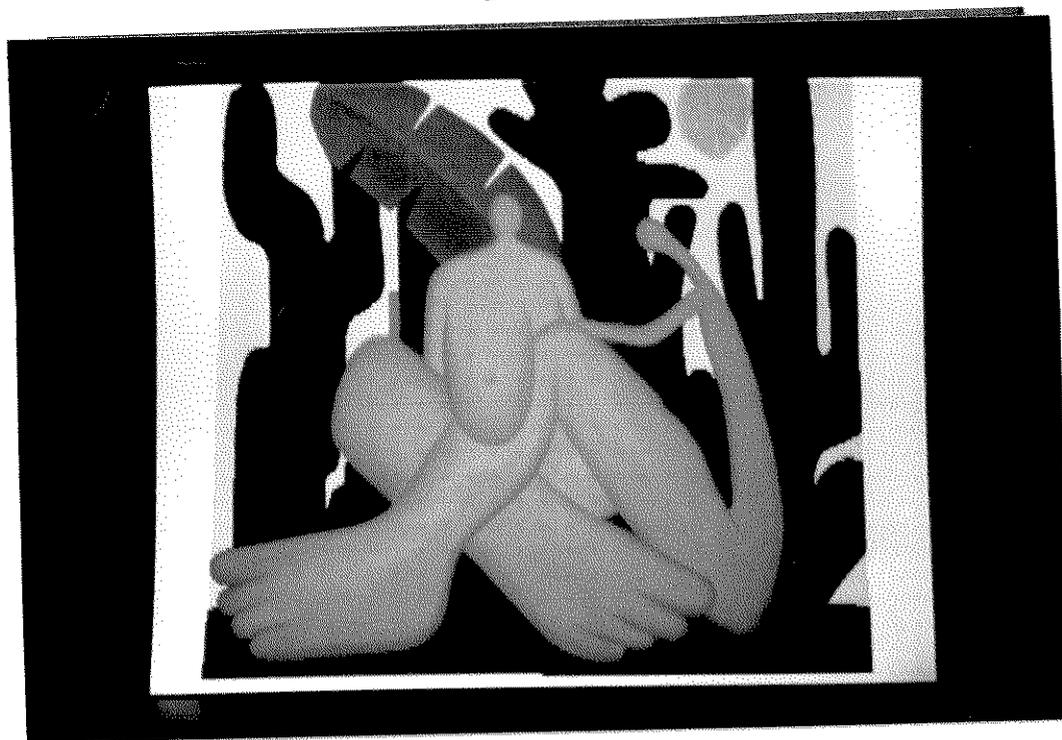


foto 7.9 - a) Tarsila do Amaral, "Antropofagia", óleo sobre tela.
b) Mesma imagem colorizada por interpolação cilíndrica. Foram
definidos 10 segmentos. Em cada um foi utilizada uma interpolação
adequada.

O algoritmo apresentado abaixo resume o processo de colorização por interpolações, já que a interpolação cilíndrica pode ser vista como sendo uma generalização dos três casos previamente apresentados. Neste algoritmo, é suposto que se deseja colorizar um segmento, que ocupa entradas na look-up-table do nível de cinza $N_{inicial}$ ao nível N_{final} , e que o usuário selecionou os limites de interpolação adequados ($H_{inicial}$, H_{final} , $S_{inicial}$, S_{final} , $V_{inicial}$, V_{final}).

PARA $n = N_{inicial}$, N_{final} FAÇA:

$$H = H_{inicial} + (H_{final} - H_{inicial}) * (n - N_{inicial}) / (N_{final} - N_{inicial});$$

$$S = S_{inicial} + (S_{final} - S_{inicial}) * (n - N_{inicial}) / (N_{final} - N_{inicial});$$

$$V = V_{inicial} + (V_{final} - V_{inicial}) * (n - N_{inicial}) / (N_{final} - N_{inicial});$$

HSVRGB (H,S,V,R,G,B); (converte especificação hsv para rgb)

LUT_R(n) := R;

LUT_G(n) := G;

LUT_B(n) := B;

FIM;

As interpolações de componentes fundamentais (matiz, saturação e luminância) é o método utilizado na colorização de fotos monocromáticas, nas quais se quer reproduzir, com a maior fidelidade possível, as cores que estão presentes na imagem real.

As fotos são digitalizadas, processadas e segmentadas. Então, aplica-se uma interpolação cilíndrica a cada segmento da imagem

7.4) COLORIZAÇÃO POR FALSA COR

Na colorização por falsa-cor, selecionam-se três bandas espectrais da mesma imagem, obtidas por um satélite com varredura multi-espectral, de forma a utilizá-las como se fossem as matrizes $R(x,y)$, $G(x,y)$ e $B(x,y)$ em um sistema de arquitetura segregada. A fig. 7.3 ilustra este método de colorização:

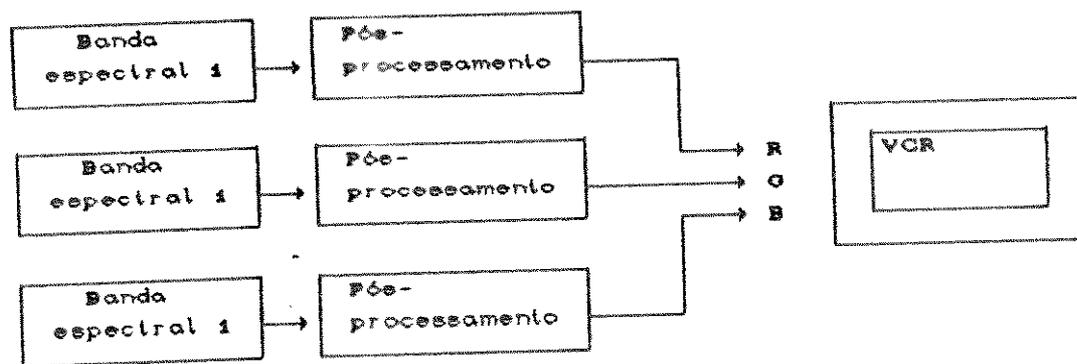


fig. 7.3 - Ilustração do processo de colorização por falsa-cor

O pós-processamento, indicado na fig. 7.3, pode ser um mapeamento de níveis de cinza, equalização de histograma ou qualquer outro algoritmo de melhoramento que se mostre adequado. O uso de algoritmos de melhoramento, em cada uma das três componentes, permite uma variação na natureza da colorização por falsa-cor obtida. Caso este pós-processamento seja feito de uma maneira padronizada, o usuário do sistema de colorização poderá

analisar grande número de imagens, usando um conhecimento adquirido sobre o tipo de colorização utilizado. Um exemplo de padronização seria o uso da banda espectral infravermelha na componente $R(x,y)$ da soma. Neste caso, a vegetação aparecerá, normalmente, com uma coloração avermelhada.

7.5) COLORIZAÇÃO POR CONTEÚDO EM FREQUÊNCIA

Uma outra abordagem semelhante à exposta em 7.4 seria a colorização por conteúdo em frequência. Parte-se de uma imagem monocromática $f(x,y)$, à qual serão aplicados três filtros digitais: um passa-altas, um passa-banda e um passa-baixas. Os resultados da convolução, com estes três filtros, devem ser armazenados nas matrizes $R(x,y)$, $G(x,y)$ e $B(x,y)$ de um sistema segregado. Caso se queira visualizar a imagem colorizada no equipamento disponível em laboratório (tipo integrado), deve-se usar o algoritmo de escolha da look-up-table ótima. A fig. 7.4 ilustra este processo de colorização por conteúdo em frequência. Da mesma forma que na colorização por falsa-cor, a escolha de um pós-processamento adequado permite variações na colorização final. Mudanças na frequência de corte w_{c1} e w_{c2} também causam uma variação apreciável na colorização obtida.

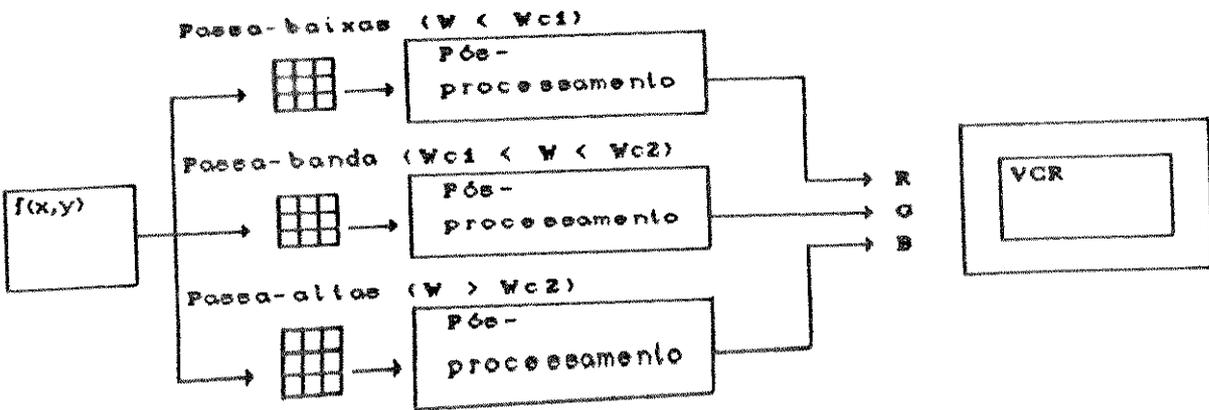


fig. 7.4 - ilustração do processo de colorização por conteúdo em frequência.

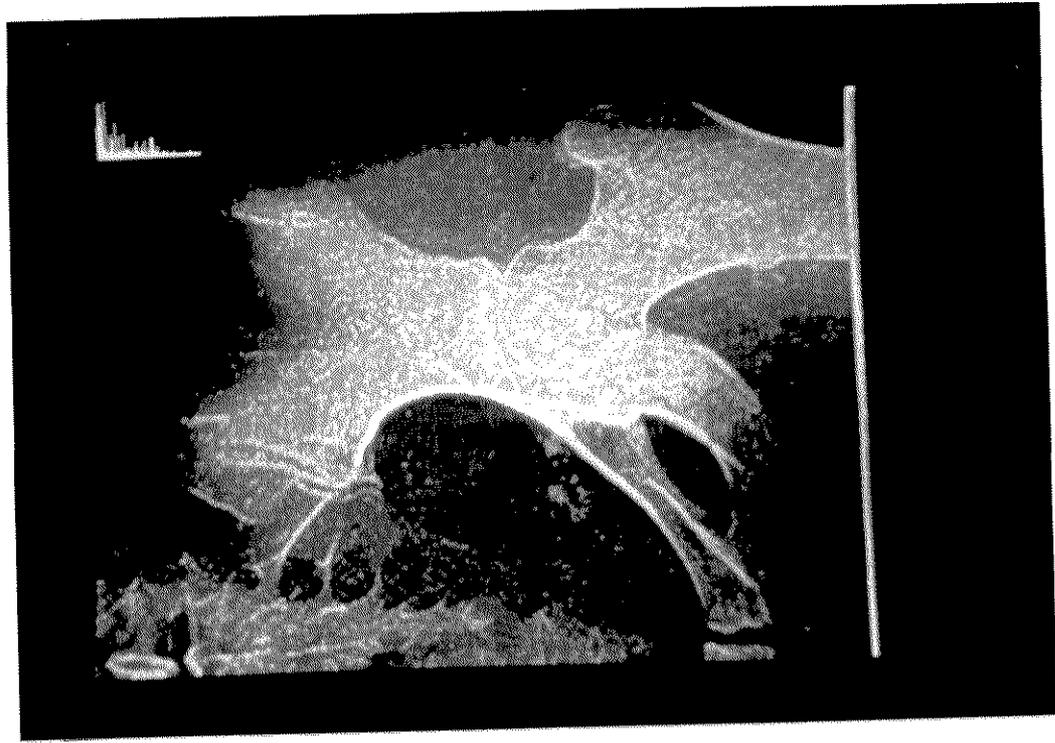


foto 7.10 - Imagem mostrada na foto 5.1 colorizada por conteúdo em frequência.

A colorização final obtida, por este método, está fortemente relacionada com o conteúdo em frequência da imagem. No sistema apresentado na fig. 7.4, as regiões de mudança abrupta de nível de cinza (alta frequência) serão colorizadas com azul, as regiões relativamente constantes (baixa frequência) serão colorizadas em vermelho e as regiões intermediárias em verde.

7.6) COLORIZAÇÃO POR ROTAÇÃO DA LOOK-UP-TABLE

No processo de colorização por rotação da look-up-table, deve-se, primeiro, inicializar a look-up-table, de forma que entradas vizinhas apresentem grande contraste de cores. Assim, o vermelho deverá ser armazenado próximo do verde, o azul do amarelo etc. A aplicação desta look-up-table à imagem monocromática resultará em uma colorização que dará destaque às pequenas diferenças de nível de cinza. Como as cores vizinhas da look-up-table possuem bastante contraste, qualquer diferença de nível de cinza, mesmo que inicialmente fosse imperceptível, será valorizada na imagem final colorizada.

Caso a colorização obtida não seja satisfatória, pode-se fazer uma pequena rotação na look-up-table (fig. 7.5). Qualquer rotação na look-up-table, mesmo que seja de apenas uma entrada, promoverá uma grande mudança visual na imagem colorizada. Pode-se continuar o processo de rotação, até que a região de interesse apresente a colorização desejada. O algoritmo abaixo ilustra a rotação da look-up-table de N entradas.

```

SE (rotação_para_cima) FAÇA:
    Temp := LUT(N);
    PARA i = N , 2 FAÇA:
        LUT(i) := LUT(i-1);
    LUT(1) := Temp;
FIM;
SENÃO
    Temp := LUT(1);
    PARA i = 1 , N-1 FAÇA:
        LUT(i) := LUT(i+1);
    LUT(N) := Temp;
FIM;

```

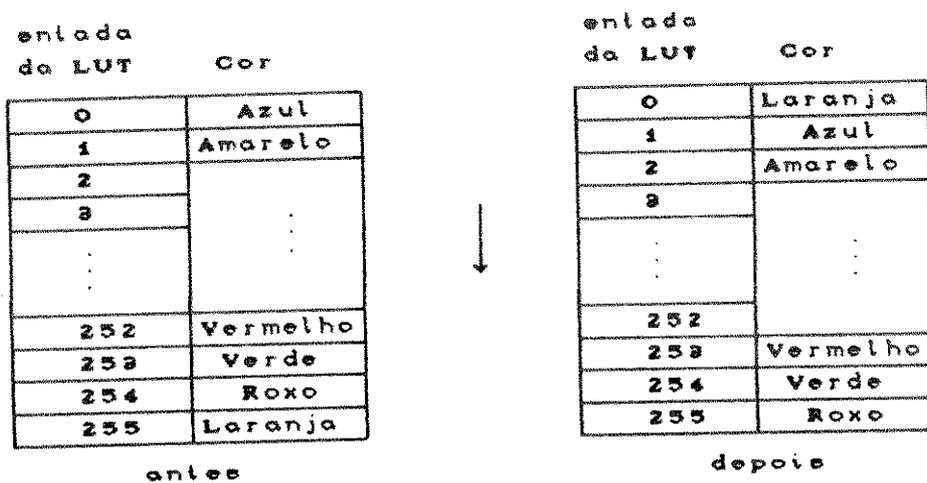
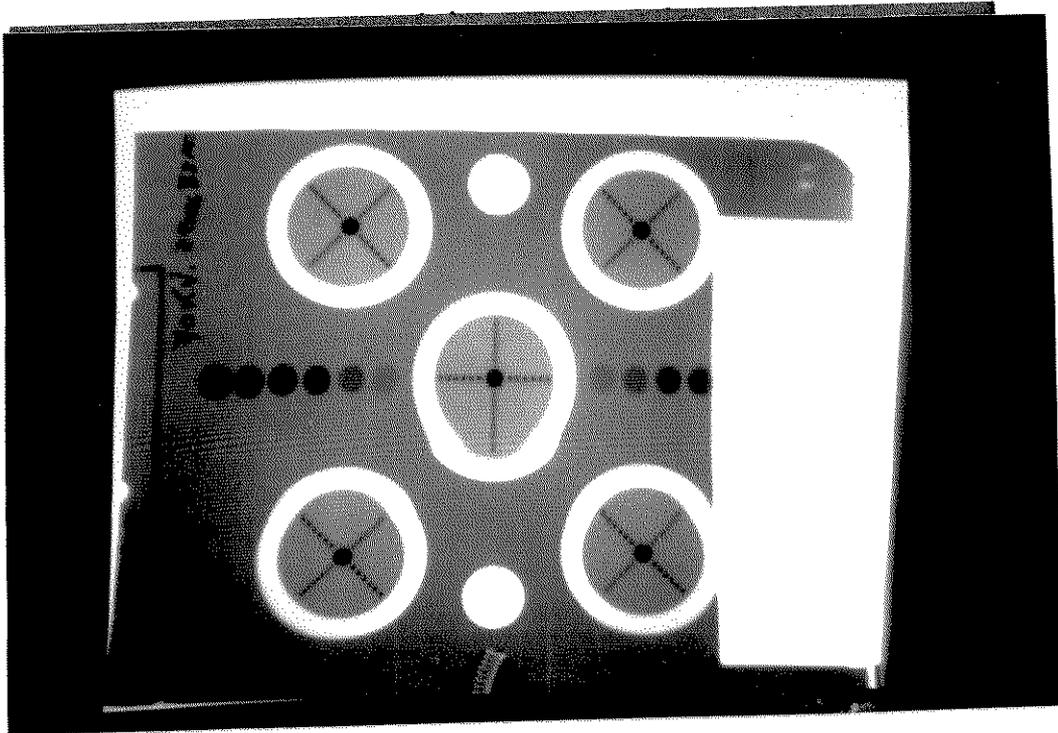


fig.7.5 - Ilustração do processo de rotação da look-up-table.

a)



b)

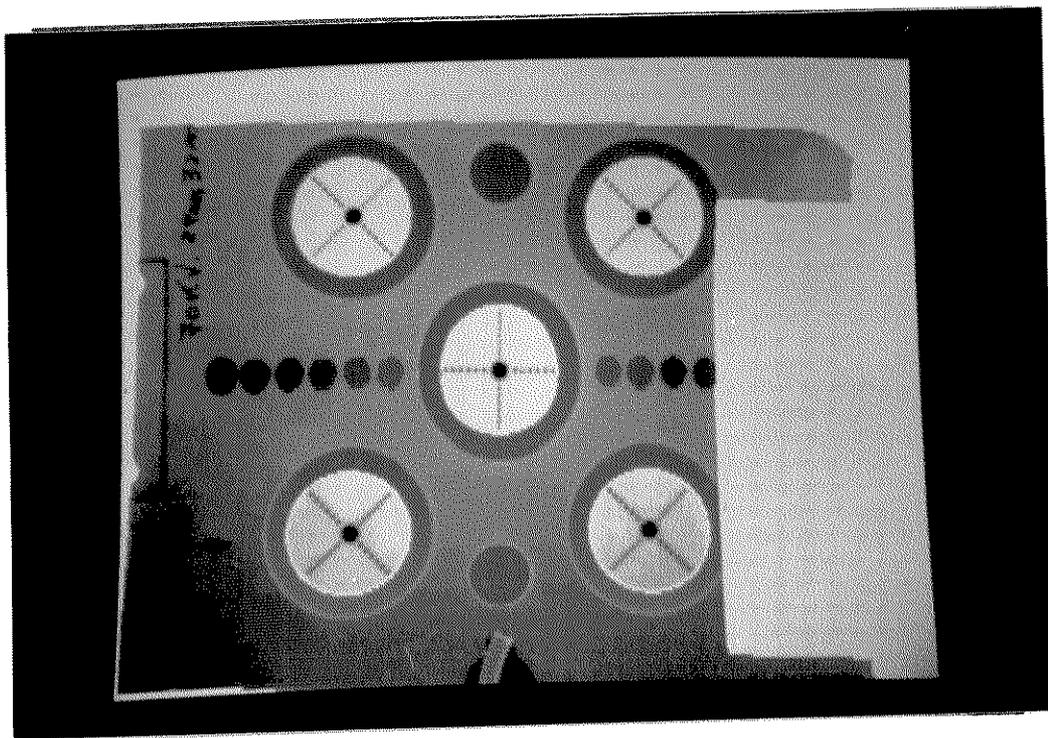


foto 7.11 - a) Radiografia de um calibrador padrão tirada com tensão de 70KV. b) Mesma imagem colorizada com dois segmentos. Aos orifícios foi aplicado um algoritmo de rotação de LUT. Ao fundo foi aplicada uma interpolação cilíndrica.

8)

CONCLUSÕES

1) O modelo HSV é muito interativo, pois permite que o usuário se familiarize rapidamente com o modo de obtenção de uma cor no equipamento. O algoritmo de conversão implementada (HSV para RGB e vice-versa) usa a escala de cores modificada, o que mostra ótimos resultados práticos, a medida que os tons de vermelho e verde são mais valorizados, em detrimento dos tons de azul.

2) O uso do método de compressão da imagem e partição da LUT é muito adequado quando se tem um equipamento de arquitetura integrada. A colorização com este método é rápida e interativa, o que não acontece, caso se opte pela geração direta das três imagens $R(x,y)$, $G(x,y)$ e $B(x,y)$. Neste caso, a cada modificação na colorização, deve-se aplicar um algoritmo de escolha da LUT ótima para a visualização da imagem colorizada no equipamento disponível, o que é um processo muito demorado.

3) Caso se tenha disponível um equipamento de arquitetura segregada, nenhum dos métodos de colorização desenvolvidos ficam invalidados, pois o pacote permite que se opte pela geração direta das três imagens $R(x,y)$, $G(x,y)$ e $B(x,y)$. Neste caso, não mais é necessário o uso do algoritmo de escolha da LUT ótima.

4) O pacote básico de processamento digital de imagens

implementado é utilizado, em todos os exemplos apresentados, como ferramenta auxiliar aos processos de segmentação e colorização. Com o uso dos algoritmos deste pacote, as deficiências do digitalizador podem, a medida do possível, ser contornadas.

5) Dentre os algoritmos de agudização de imagens implementados, o Laplaciano apresenta resultados quase sempre satisfatórios. A sua aplicação é sempre mais rápida que a do operador de Sobel, não exigindo, ainda, a demorada fase de projeto dos filtros passa-altas. O resultado da aplicação dos algoritmos de agudização devem ser sobrepostos à imagem inicial, de modo a se facilitar o processo de segmentação.

6) Dentre os algoritmos de suavização de imagens, o filtro do valor mediano apresenta os melhores resultados. Consegue-se um ótimo grau de suavização, com significativa redução de ruído, sem que se borre a imagem, isto é, sem que se perca a definição das bordas e contornos. Quando se usa um filtro passa-baixas ou um filtro do valor médio, se obtém sempre maior grau de borramento para um mesmo grau de suavização obtido por um filtro de valor mediano.

7) Os algoritmos de manipulação de histograma são fundamentais quando se deseja uniformizar os dados de entrada do sistema de colorização. As imagens, que são digitalizadas nas mais diversas condições de iluminação, não apresentam um histograma de nível de cinza padronizado. O uso de um algoritmos de manipulação de histograma adequado permite a definição de um mapeamento

consistente de nível de cinza para cor.

8) A segmentação baseada em textura é uma ferramenta muito importante para a segmentação de imagens digitais. No caso do crescimento de regiões, a escolha de um predicado de parada adequado permite que se segmente da melhor maneira possível a imagem. O uso de predicados mais complexos, que envolvam medidas estatísticas de segunda ordem, permite a segmentação de uma enorme variedade de objetos.

9) A subdivisão de área, embora permita menor flexibilidade na escolha dos predicados de parada, se mostra muito interessante quando os pixel do objeto a ser segmentado se confundem com os do fundo.

10) O algoritmo de thresholding é uma ferramenta rápida de segmentação. No entanto, só é eficiente quando o objeto apresenta um forte contraste com o fundo.

11) A detecção de bordas, do mesmo modo que os operadores de agudização, são uma ferramenta auxiliar ao processo de segmentação. As bordas detetadas devem ser somadas à imagem inicial, de modo a se ter facilitado o processo de segmentação orientado para regiões.

12) A edição direta na imagem é uma ferramenta indispensável quando se deseja colorizar fotos ou filmes monocromáticos. Neste tipo de aplicação, normalmente se tem que

colorizar vários segmentos, com as mais diversas características. O uso de "tablet" facilita um demasia a edição direta na imagem, permitindo que se faça a partição mais adequada da imagem.

13) A edição direta na imagem sempre provoca o aparecimento de bordas e contornos muito agudos, o que tira a naturalidade da cena que se quer colorizar. Os algoritmos de suavização aplicados à cada uma das componentes da imagem, ($R(x,y)$, $G(x,y)$ e $B(x,y)$) reduzem bastante este problema.

14) Em todas as imagens nas quais se aplica algoritmos de colorização, consegue-se ressaltar características que estavam mascaradas por um baixo contraste de nível de cinza na imagem monocromática. As imagens se tornam mais fáceis de serem analisadas e interpretadas, com pode ser observado nos exemplos mostrados no capítulo 7, principalmente no que se refere a ecografias, tomografias, radiografias, fotos aeroespaciais etc.

15) A colorização por planos de corte se mostra muito interessante quando se tem por objetivo valorizar as características de profundidade da imagem original. Este é o caso da radiografia de mapeamento de tireóide mostrada na foto 7.1. O uso de sete planos de corte ressaltou a noção tridimensional da radiografia. O uso da escala intuitiva vermelho/quente, violeta/frio pode facilitar a interpretação de um especialista. No caso da tomografia de testemunho de rocha-reservatório, mostrada na foto 7.2, o uso de planos de corte permitiu que se separasse, com facilidade, os elementos água, óleo e lama de perfuração. A

água ficou colorizada com um tom azul-violeta, a lama com um tom amarelado e o óleo com os tons de vermelho.

16) A colorização por funções se mostra muito prática em aplicações nas quais se quer padronizar o tipo de colorização a se usado. O uso desta técnica é rápido e eficiente. Uma vez definida a função a ser aplicada a uma imagem, a colorização de outras imagens do mesmo tipo se torna uma tarefa quase instantânea. Esta técnica pode ser aplicada com sucesso em imagens médicas e fotos aeroespaciais, onde não se pode perder muito tempo na colorização de uma única foto. Na radiografia de crânio e coluna cervical, mostrada na foto 7.3, o uso da cor melhorou a definição de detalhes, principalmente nas vértebras da coluna. Na ecografia da artéria aorta, mostrada na foto 7.4, conseguiu-se uma boa separação das artérias do fundo. A foto aeroespacial, mostrada na foto 7.5, foi consideravelmente melhorada, conseguindo-se reduzir o ruído de fundo. Esta redução foi conseguida com o uso de um algoritmo de suavização de imagens. O uso de uma colorização por funções nesta imagem, no segmento que contém o continente, valorizou a visualização do relevo presente na região.

17) Com o uso de colorização por interpolações, consegue-se um bom grau de fidelidade em aplicações nas quais se quer reproduzir as cores que estavam presentes na cena real. De modo a se ter uma medida do grau de fidelidade das reproduções de cor, pode-se digitalizar fotos coloridas no sistema. Como a câmera CCD disponível é monocromática, a imagem digitalizada será também monocromática. Esta imagem digitalizada deve ser colorizada por

algoritmos de interpolação cilíndrica adequados, de forma a se reproduzir as cores presentes na foto original. Uma comparação da foto original com a imagem colorizada dá uma boa medida da acuidade da colorização. Esta foi a idéia por trás da colorização das fotos 7.8 e 7.9. As imagens colorizadas finais mostraram boa semelhança com os quadros originais. Os resultados obtidos nestas imagens demonstraram a potencialidade do sistema para o uso em colorizações de fotos e filmes monocromáticos.

18) As interpolações se mostraram também eficientes na colorização de imagens médicas, como pode ser visto nas fotos 7.6 e 7.7. A interpolação de saturação, usada no segmento central da tomografia da mostrada na foto 7.6, melhorou consideravelmente o sombreamento presente na imagem original. Do mesmo modo que na colorização por funções, as interpolações podem se padronizadas para a colorização de imagens médicas. Este é o caso da interpolação de matiz usada na radiografia da articulação coxo-femural mostrada na foto 7.7. Os detalhes da radiografia ficaram bem mais claros, ficando o contraste bem mais acentuado.

19) A colorização por conteúdo em frequência se mostra eficiente em imagens que não tenham contornos e bordas bem definidos. As regiões de transição, por estarem relacionadas com componentes de alta frequência da transformada de Fourier da imagem, apresentam, nas imagens colorizadas por este método, uma cor azulada, o que facilita a sua identificação. A radiografia mostrada na foto 7.10 ilustra este melhoramento da definição das bordas. O uso deste método de colorização no equipamento

disponível não permite o uso de compressão da imagem e partição da LUT, pois deve-se ter as três imagens, resultantes das aplicações dos três filtros digitais, bem definidas

20) A colorização por rotação da look-up-table pode ser usada em imagens que apresentem pequenas diferenças de nível de cinza. Na radiografia do calibrador padrão, mostrada na foto 7.11, conseguiu-se diferenciar com facilidade os níveis de cinza presentes nos orifícios de profundidade padronizada. Neste tipo de colorização, é necessário o uso de uma escala de cores de alto contraste. No exemplo mostrado, usou-se um espaçamento de 30° no matiz, o que se mostrou bastante adequado.

BIBLIOGRAFIA

- [Alvarez/89] Marcelo A. Alvarez, Montagem de uma Estação Digitalizadora e Processadora de Imagens, RT-DCA 005, Unicamp, 1989.
- [Asmus/82] John F. Asmus, Digital Image Processing in Art Conservation, BYTE, pp 151 a 166, março 1987.
- [Andrew/82] H. C. Andrew, A. G. Tescher e P. P. Kruger, Image Processing by Digital Computer, IEEE Spectrum, vol. 9, número 7, pp 20-30, julho 1982.
- [Bernstein/78] Ralph Bernstein, Digital Image Processing for Remote Sensing, IEEE Press, 1978.
- [Castanho/90] José E. C. Castanho, Uma Estação de Trabalho para Visão Computacional, Tese de Mestrado, Unicamp, 1990.
- [Dougherty/87] Edward R. Dougherty e Charles R. Giardina, Matrix Structured Image Processing, Prentice Hall, 1987.

- [Dudgeon/83] D. E. Dudgeon, R. M. Mersereau, Multidimensional Digital Signal Processing, Englewood Cliffs, Prentice Hall, 1983.
- [Ekstrom/84] Michael P. Ekstrom, Digital Image Processing Techniques, Academic Press Inc, 1984.
- [Encarnação/86] J. Encarnação e W. Strasser, Computer Graphics: Geraetetechnik, Programmierung und Anwendung Graphischer Systeme, Oldenbourg Verlag, 1986.
- [Foley/84] J. D. Foley e A. Van Dam, Fundamentals of Interactive Computer Graphics, Addison-Wesley Publishing Company, 1984.
- [Fonseca/90] Luciano E. N. da Fonseca, O Uso da Cor no Processamento Digital de Imagens, SIBGRAPI'90, anais, 1990.
- [Gonzalez/87] C. R. Gonzalez e P. Wintz, Digital Image Processing, Addison-Wesley Publishing Company, 1987.

- [Green/83] William B. Green, Digital Image Processing - A System Approach, Van Nostrand Reinhold Electrical, Computer Science and Engineering Series, 1983.
- [Hall/79] Ernest L. Hall, Computer Image Processing and Recognition, Academic Press, 1979.
- [Harrington/87] Steven Harrington, Computer Graphics: A Programming Approach, McGraw-Hill International Edition, 1987
- [Heckbert/82] Paul Heckbert, Color Image Quantization for Frame Buffer Display, Computer Graphics, Vol 16, número 3, pp 297 a 307, julho 1982.
- [Loureiro/89] Ricardo A. M. Loureiro, Detecção e Tratamento de Bordas de Elementos de uma Imagem, RI/DCA-003/89
- [Mascarenhas/89] Néelson D. A. Mascarenhas e Flávio R. D. Velasco, Processamento Digital de Imagens, EBAI, 1989.
- [Newman/79] William M. Newman e Robert F. Sproull, Principles of Interactive Computer Graphics, McGraw-Hill, 1979.

- [Oppenheim/75] Alan V. Oppenheim e Ronald W. Schafer, Digital Signal Processing, Prentice Hall, 1975.
- [Paiva/90] Rogério F. Paiva, Análise em Geologia e Engenharia de Reservatórios Usando Tomografia Computadorizada de Raio-x, Tese de Mestrado, Unicamp, 1990.
- [Pedrosa/82] Israel Pedrosa, Da Cor à Cor Inexistente, Editora Universidade de Brasília, 1982.
- [Pratt/78] William K. Pratt, Digital Image Processing, John Wiley and Sons, 1978.
- [Rabiner/75] L. R. Rabiner e B. Gold, Theory and Applications of Digital Signal Processing, Prentice Hall, 1975.
- [Rosenfeld/82] Azriel Rosenfeld e Avinash C. Kak, Digital Picture Processing, Vol. 1 e 2, Academic Press, 1982.
- [Schaale/89] Michael Schaale, Farbenlotto: Wiedergabe von Echtfarbbildern mit VGA-Karten, C'T - Magazin für Computertechnik, pp 166 a 178, dezembro 1989.

- [Schalkoff/89] Robert J. Schalkoff, Digital Image Processing and Computer Vision, John Wiley and Sons, 1989.
- [Schowengerdt/83] Robert A. Schowengerdt, Techniques for Image Processing and Classification in Remote Sensing, Academic Press, 1983.
- [Sheldon/89] Ken Sheldon, A Film of a Different Color, Byte, pp 164 a 165, março 1987.
- [Stevens/82] R. J. Stevens, A. F. Lehar e F. H. Preston, Manipulation and Presentation of Multidimensional Image Data Using Peano Scan, IEEE Transaction on Pattern Analysis and Machine Intelligence, vol PAMI-5, número 5, Setembro 1983.
- [Vilaca/89] Ari M. Vilaca e Clésio Luiz Tozzi, Síntese de Imagens Realísticas, uma Abordagem com Metodologia Ray-Tracing, Tese de mestrado, FEE Unicamp, Setembro 1989.
- [Wyszecky/82] G. Wyszecky e W. S. Stiles, Color Science, Concepts and Methods, Quantitative Data and Formulae, John Wiley and Sons, 1982.