PICTŌREA

UMA FERRAMENTA DE ENSINO PORTĀTIL PARA TRATAMENTO DE IMAGENS

NEUCIMAR JERÔNIMO LEITE

Dissertação apresentada ao Curso de MESTRADO EM ENGENHARIA ELÉTRICA da Universidade Federal da Paraíba, em cumprimento às exigências para a obtenção do Grau de Mestre.

ĀREA DE CONCENTRAÇÃO: PROCESSAMENTO DA INFORMAÇÃO

Prof. Arnaldo de Albuquerque Araújo, D.Sc
Orientador

Prof. João Marques de Carvalho, Ph.D
Co-Orientador

CAMPINA GRANDE
Setembro, 1989

AGRADECIMENTOS

Aos Orientadores
pela forma séria com que conduziram
o presente trabalho

Aos Amigos
que no laço indizível da amizade
participaram desses momentos de estudo

A minha grande Família
pelo estímulo e compreensão

L533f Leite, Neucimar Jerônimo.
Pictórea : uma ferramenta de ensino portátil para tratamento de imagens / Neucimar Jerônimo Leite. - Campina Grande, 1989.
205 f.

Dissertação (Mestrado em Engenharia Elétrica) - Universidade Federal da Paraíba, Centro de Ciências e Tecnologia, 1989.
"Orientação : Prof. Dr. Arnaldo de Albuquerque Araújo, Prof. Dr. João Marques de Carvalho".
Referências.

1. Processamento Digital de Imagens. 2. Software - Sistema. 3. Microcomputadores. 4. Dissertação - Engenharia Elétrica. I. Araújo, Arnaldo de Albuquerque. II. Carvalho, João Marques de. III. Universidade Federal da Paraíba - Campina Grande (PB). IV. Título

CDU 004.932(043)

PICTOREA: UMA FERRAMENTA PORTÁTIL PARA ENSINO DE TRATAMENTO DE IMAGENS

NEUCIMAR JERÔNIMO LEITE

DISSERTAÇÃO APROVADA EM

ARNALDO DE ALBUQUERQUE ARAÚJO, D.Sc., UFPB
Orientador

JOÃO MARQUES DE CARVLHO, Ph.D., UFPB
Componente da Banca

PAULO ROBERTO CAMPOS DE ARAÚJO, Mestre, UFPB
Componente da Banca

ASCENDINO FLÁVIO DIAS E SILVA, Dr.Ing., UFPE
Componente da Banca

CAMPINA GRANDE - PB
OUTUBRO - 1989

RESUMO

Este trabalho descreve o desenvolvimento de um pacote de software modular para simulação de um sistema de processamento digital de imagens (PDI). Projetado para trabalhar em máquinas do tipo IBM-PC e compatíveis, PICTÓREA consiste, inicialmente, de 10 módulos básicos que compreendem funções de processamento e visualização de imagens codificadas em 32 níveis de cinza, de tamanho 64x64 pixels.

O referido sistema pode ser utilizado em microcomputadores que possuam cartões gráficos do tipo CGA (versão 4 cores) ou do tipo EGA (versão 16 cores). Estas duas versões operam de modo semelhante, exceto no que diz respeito ao conjunto das cores apresentadas no vídeo.

A meus Pais

Cirilo e Maria Jerônimo

ÍNDICE

Página

Página

ÍNDICE DAS FIGURAS

Página

LISTA DE ABREVIAÇÕES

PDI	-	Processamento Digital de Imagens
IBM	-	"International Business Machines"
PC	-	"Personal Computer"
GDT	-	"Graphics Development Toolkit"
MI	-	Memória de Imagem
CGA	-	"Color/Graphics Monitor Adapter"
EGA	-	"Enhaced Graphics Adapter"
VDI	-	"Virtual Device Interface"
E/S	-	Entrada e Saída
SPDI	-	Sistema de Processamento Digital de Imagens
CRT	-	"Cathode-ray Tube"
NDC	-	"Normalized Device Coordinates"
MIN	-	Mínimo
MAX	-	Máximo
MDA	-	Média de Diferenças Absolutas
EMQ	-	Erro Médio Quadrático
FDC	-	Função de Distribuição Cumulativa
SVSV	-	Suavização com Vizinhança Selecionada por Variância
SSDA	-	Suavização com Vizinhança Selecionada pela Soma de Diferenças Absolutas
SDA	-	Soma de Diferenças Absolutas

1. INTRODUÇÃO

Processamento Digital de Imagens é uma das áreas onde os esforços em pesquisas e estudos têm crescido consideravelmente. Nestes últimos 25 anos, inúmeras técnicas de processamento de imagens foram elaborados concomitantemente com o rápido progresso da tecnologia de hardware [1]. Muitas destas técnicas são desenvolvidas para aplicações científicas. Por exemplo, as aplicações industriais e científicas abrangem: análise térmica, inspeção de peças, contagem de partículas, automação e visão de robôs, análise de cromossomos, raios-x industriais etc. Em aplicações militares e de segurança podemos citar: visão noturna, mapeamento e classificação de terrenos, deteção de alvos e rastreamento. Na medicina, a tomografia computadorizada, ressonância magnética nuclear, ultrassonografia, técnicas para interpretação de raios-x. Na geografia, podemos citar, dentre outras, a interpretação de imagens provenientes de satélites para o estudo da poluição em determinada área.

Processamento Digital de Imagens (PDI), pode ser entendido como a manipulação de imagens por computador, com o objetivo de extrair informações dessas imagens ou transformá-las de modo a facilitar nossa interpretação [2].

Muitas técnicas e ferramentas matemáticas, aplicadas ao processamento de imagens, foram desenvolvidas para atender a três problemas básicos [3].

1. *Digitalização e codificação de imagens*: conversão de imagens do plano contínuo para o discreto (digitalização), e compressão do resultado, de maneira a preservar o espaço de armazenamento ou a capacidade do canal

de transmissão.

2. *Realce e restauração de imagens:* recuperação de imagens que sofreram algum processo de degradação.

3. *Segmentação e descrição de imagens:* conversão de imagens em mapas simplificados; medida das propriedades das imagens ou partes destas; classificação ou descrição de imagens, em termos de suas componentes ou propriedades.

Uma imagem monocromática pode ser representada por uma função f(x,y) onde f é uma medida do nível de cinza ou brilho (intensidade) no ponto (x,y) [4]. Estas imagens são mapeadas numa matriz bidimensional finita e armazenadas, geralmente, no computador. Cada elemento de imagem é chamado de pixel (abreviação do inglês para "picture element").

Os componentes básicos que formam um sistema de PDI podem ser divididos em três categorias: digitalizador, computador e unidade de visualização. PICTÓREA, uma ferramenta de ensino de processamento digital de imagens [5,6], apresenta-se como alternativa, quando não se dispõe de um sistema deste tipo, e se deseja desenvolver trabalhos de laboratório na referida área. PICTÓREA oferece, assim, a professores e alunos, a possibilidade de se implementar, com dados reais, os diversos conceitos e algoritmos de PDI abordados em sala de aula.

O sistema baseia-se em sugestões dadas por Gonzalez e Wintz, em seu livro "Digital Image Processing" [7] e por Araújo, em seu trabalho sobre filtros espaciais [4]. Este último sugere a criação de um programa para a simulação de um sistema de PDI a ser utilizado em máquinas do tipo IBM-PC. Gonzalez e Wintz fornecem uma subrotina para a impressão de imagens com 32 níveis de cinza, além de um conjunto de imagens, já codificadas, que ser

vem de base para a realização dos experimentos a serem executados, à medida que o aluno avance na leitura do texto didático. Partindo desta idéia, resolvemos estender e flexibilizar as possibilidades de visualização e de processamento, dando origem ao sistema apresentado a seguir.

Os capítulos deste trabalho estão organizados da seguinte forma. O capítulo 2 dá uma idéia geral sobre a estrutura e o conjunto das funções contidas em PICTÓREA. O capítulo 3 fornece maiores detalhes sobre o pacote gráfico utilizado pelo sistema (GDT), e descreve o MENU DE ABERTURA presente após a inicialização de PICTÓREA. Finalmente, os capítulos 4, 5 e 6 abordam, detalhadamente, o conjunto das funções de processamento e visualização mencionadas no capítulo 2.

2. DESCRIÇÃO DO SISTEMA

A possibilidade de visualizar imagens e modificações sofridas pelas mesmas, após a aplicação de algoritmos de processamento digital de imagens é um dos fatores que motivam o interesse de alunos por esta área. Para visualizar imagens digitalizadas é necessário que se disponha de um sistema de tratamento de imagens composto, pelo menos, de computador e unidade de visualização de imagens. Mesmo quando se dispõe de tal sistema, nem sempre é possível alocá-lo para que alunos executem tarefas de laboratório acopladas a uma disciplina de PDI.

PICTÓREA apresenta-se como solução prática para contornar este problema. Desenvolvido para trabalhar em máquinas do tipo IBM-PC, PICTÓREA consiste, basicamente, de um arquivo de imagens de tamanho 64x64 pixels e escala de cinza com 32 níveis, e 3 memórias de imagens (MI1, MI2 e MI3) armazenadas na memória de trabalho do computador. Imagens armazenadas em MI1 e MI2 podem ser processadas isoladamente ou podem sofrer processamento que envolva ambas. O resultado do processamento é, em geral, armazenado em MI3. Como unidade de visualização de imagens, o sistema utiliza os cartões gráficos de 4 (CGA) ou 16 cores (EGA) disponíveis em máquinas do tipo IBM-PC, para dar o "display" de imagens com redução da escala de cinza para 4 ou 16 níveis, dependendo do cartão gráfico empregado. Imagens com 32 níveis de cinza são fornecidas ao usuário através de um método de impressão por superposição de caracteres [7], conforme veremos no capítulo 4.

2.1. Concepção do sistema

Os seguintes tópicos foram abordados na fase inicial de concepção do

sistema:

- coleção e seleção de alguns algoritmos de PDI conhecidos da literatura;

- escolha da linguagem de programação: FORTRAN 77;

- manipulação de dados: o programa consiste de subrotinas, onde campos são transferidos através de listas de parâmetros;

- padrão de documentação: além da descrição dos parâmetros necessários aos algoritmos, subrotinas de auxílio são disponíveis. Os programas são documentados na língua portuguesa;

- opção por um sistema modular baseado em menus, com apresentação "linha por linha" da tela e com diálogos do tipo "perguntas e respostas " [8].

A biblioteca apresenta-se dividida em quatro planos básicos (Fig. 2.1). No primeiro plano estão as rotinas que implementam os diversos algoritmos da maneira mais independente possível. No segundo plano estão as rotinas comuns às do exterior como, por exemplo, as rotinas de convolução. No plano seguinte estão as rotinas de auxílio, que contêm informações sobre a utilização do sistema. No plano mais interno, associado ao sistema operacional, encontra-se o VDI ("Virtual Device Interface") [9], que gerencia o controle de dispositivos de entrada e saída, independentemente do hardware utilizado.

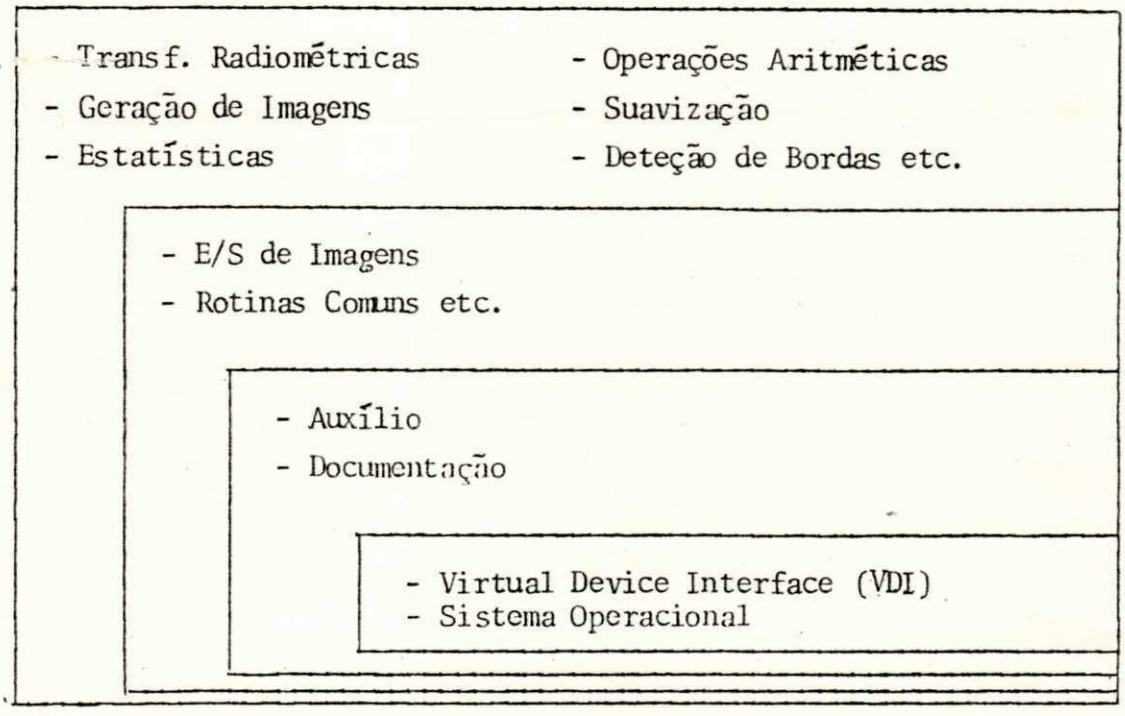


Figura 2.1. Estrutura do software

O controlador VDI, que integra o pacote gráfico da IBM, o GDT ("Graphics Development Toolkit"), define um protocolo ou linguagem de comunicação que consiste de funções pré-definidas, métodos de acesso e convenções de parâmetros que possibilitam um software com as características acima mencionadas. A Figura 2.2. apresenta a ligação entre PICTÓREA e os diversos dispositivos de entrada e saída. O capítulo 3 fornece maiores detalhes sobre o GDT.

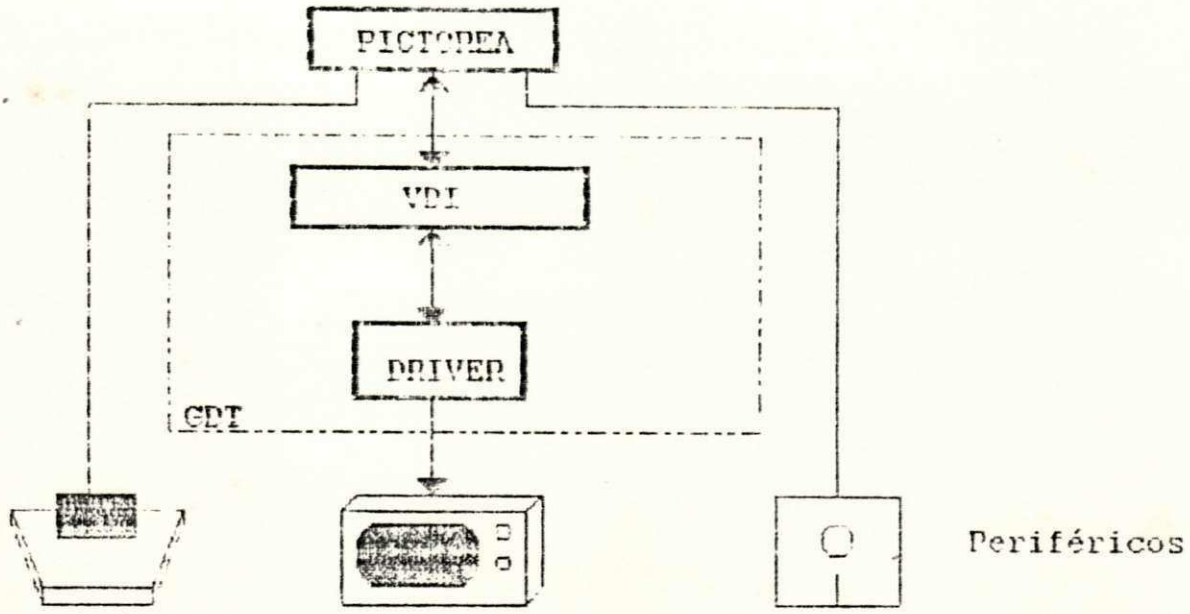


Figura 2.2. Controle dos dispositivos de E/S pelo VDI

2.2. Funções contidas no sistema

Ao executar o programa PICTÓREA.EXE, o usuário dispõe, inicialmente, de um MENU DE ABERTURA que agrupa as seguintes funções:

1 - PROGRAMA SPDI

2 - PROGRAMA DO USUÁRIO

3 - DIRETÓRIO DE IMAGENS

4 - AUXÍLIO

5 - ENCERRA

A função PROGRAMA SPDI (programa de Sistema de Processamento Digital de Imagens), compreende o conjunto de algoritmos que realizam as funções de processamento e de visualização de imagens propriamente ditas. O PROGRAMA DO USUÁRIO permite que o usuário, através de algumas regras básicas,

execute o seu programa fonte e, em seguida, utilize os recursos oferecidos pela função PROGRAMA SPDI. O DIRETÓRIO DE IMAGENS fornece o diretório das imagens presentes no acionador de disco indicado pelo usuário. A função AUXÍLIO contém informações gerais sobre PICTÓREA e a função ENCERRA conclui a execução do programa, transferindo o controle ao sistema operacional.

Ao acessar PROGRAMA SPDI, 10 módulos de processamento e visualização são apresentados ao usuário. Estes módulos são:

MENU PRINCIPAL

Módulos:

1. Entrada/saída de imagens
2. Display de imagens
3. Histogramas e estatísticas
4. Aritmética de imagens
5. Transformações radiométricas
6. Filtros espaciais
7. Geração de imagens
8. Geração de ruídos
9. Auxílio
10. Fim

As funções de um módulo que contêm outro conjunto de funções são apresentadas como submódulo I deste módulo. Da mesma forma, uma função deste submódulo, contendo outras funções, é apresentada como submódulo II.

Descreveremos, a seguir, o conjunto das funções contidas em cada um desses módulos. Os capítulos 4, 5 e 6 descrevem detalhadamente estas funções.

Módulo 1: ENTRADA/SAÍDA DE IMAGENS

Funções:

F1 Carrega imagens

F2 Armazena imagem

F3 Imprime imagem

F4 Transfere imagens entre MI's

F9 Auxílio

F10 Retorna

Módulo 2: DISPLAY DE IMAGENS

Este módulo é responsável pela visualização das imagens nas MI's (memórias de imagens). As mesmas são apresentadas com redução da escala de cinza de 32 níveis para 4 níveis (PICTÓREA: versão 4 cores) ou para 16 níveis (PICTÓREA: versão 16 cores).

DISPLAY DE IMAGENS está presente em vários outros módulos do sistema a fim de agilizar a possibilidade de visualização das MI's.

Módulo 3: HISTOGRAMAS E ESTATÍSTICAS

Funções:

F1 Display de histograma

F2 Display de perfil de linha ou coluna

F3 Estatísticas locais

F4 Verifica e/ou modifica pixels

F9 Auxílio

F10 Retorna

Módulo 4: ARITMÉTICA DE IMAGENS

Funções:

F1 Adição de MI1 e MI2

F2 Subtração de 2 imagens

F3 Diferença absoluta entre MI1 e MI2

F4 Média da diferença absoluta e erro médio quadrático entre 2 imagens

F5 Efeito zoom

F6 Redução de imagem

F7 Display de imagens

F9 Auxílio

F10 Retorna

Módulo 5: TRANSFORMAÇÕES RADIOMÉTRICAS

Funções:

F1 Equalização histogrâmica

F2 Mapeamento dos níveis de cinza

F3 Display de imagens

F9 Auxílio

F10 Retorna

Submódulo 1:/TRANSFORMAÇÕES RADIOMÉTRICAS/MAPEAMENTO DOS NÍVEIS DE CINZA

Funções:

F1 Fatiamento em dois níveis

F2 Compressão

F3 Compressão/expansão monotônica

F4 Fatiamento por plano com fundo

F5 Aumento linear do contraste

F6 Fatiamento por plano

F7 Inversão da escala de cinza

F8 Dente de serra 3-ciclos

F10 Retorna

OUTRA TECLA Menu principal

Módulo 6: FILTROS ESPACIAIS

Funções:

F1 Convolução

F2 Deteção de bordas

F3 Suavização

F4 Display de imagens

F9 Auxílio

F10 Retorna

Submódulo 1:/FILTROS ESPACIAIS/DETEÇÃO DE BORDAS

Funções:

F1 Gradiente de Roberts

F2 Operador de Sobel

F3 Operador de Prewitt

F4 Operadores direcionais

F10 Retorna

OUTRA TECLA Menu principal

Submódulo 11:/FILTROS ESPACIAIS/DETEÇÃO DE BORDAS/GRADIENTE DE ROBERTS

Funções:

F1 Aplicação direta do gradiente

F2 Gradiente com fundo definido

F3 Imagem gradiente binária

F10 Retorna

OUTRA TECLA Menu principal

Submódulo 11:/FILTROS ESPACIAIS/DETEÇÃO DE BORDAS/OPERADORES DIRECIONAIS

Funções:

F1 Máscaras direcionais de Prewitt

F2 Máscaras direcionais de Kirsch

F3 Máscaras simples de 3 níveis

F4 Máscaras simples de 5 níveis

F10 Retorna

OUTRA TECLA Menu principal

Submódulo 1:/FILTROS ESPACIAIS/SUAVIZAÇÃO

Funções:

F1 Filtro da média

F2 Filtros da ordem

F3 Suavização com vizinhança selecionada por variância

F4 Suavização com vizinhança selecionada por soma de diferenças absolutas

F5 Filtro da média com os k-vizinhos mais próximos

F6 Filtro sigma

F10 Retorna

OUTRA TECLA Menu principal

Módulo 7: GERAÇÃO DE IMAGENS

Funções:

F1 Círculo

F2 Retângulo

F3 Quatro quadrados

F4 Tabuleiro

F5 Display de imagens

F9 Auxílio

F10 Retorna

Módulo 8: GERAÇÃO DE RUÍDOS

Funções:

F1 Ruído com distribuição uniforme

F2 Ruído com distribuição gaussiana

F3 Display de imagens

F9 Auxílio

F10 Retorna

Módulo 9: AUXÍLIO

O módulo AUXÍLIO contém informações básicas sobre a utilização de PICTOREA. Cada um dos módulos do MENU PRINCIPAL dispõe de uma função AUXÍLIO dedicada a este módulo, onde algumas informações sobre os algoritmos e parâmetros necessários ao seu funcionamento são fornecidas ao usuário.

Módulo 10: FIM

Este módulo encerra a execução do PROGRAMA SPDI, transferindo o controle ao MENU DE ABERTURA do sistema.

PICTÓREA contém funções de apoio, tais como: F10-Retorna, que põe o menu imediatamente anterior à disposição do usuário; OUTRA TECLA (presente nos submódulos I e II), que indica que o usuário deve pressionar qualquer tecla para retornar ao MENU PRINCIPAL, e a função ESCAPE, que anula a ocorrência de uma função no momento da sua chamada.

O módulo AUXÍLIO, bem como as funções de apoio, aparecem, como veremos, em vários outros módulos e submódulos do sistema. Como medida de simplificação, os mesmos não mais serão mencionados no decorrer dos capítulos seguintes.

Dois algoritmos implementando as transformadas rápidas de Fourier e de Walsh são fornecidos ao usuário juntamente com PICTÓREA. Estes algoritmos podem ser acessados através da função PROGRAMA DO USUÁRIO do MENU DE ABERTURA do sistema.

2.3. Conclusão

PICTÓREA tentou agrupar, dentro dos seus diversos módulos, um conjunto de algoritmos que possa dar ao usuário uma visão do que vem a ser processamento digital de imagens.

A redução dos níveis de cinza da imagem de 32 para apenas 4 cores, na tela, é estabelecido pela capacidade do cartão gráfico CGA, comumente disponível nas máquinas do tipo IBM-PC. PICTÓREA propõe, para cartões do tipo EGA, uma outra versão que permite o "display" simultâneo de até 16 cores, aumentando, assim, a possibilidade de resolução do sistema.

3. O PACOTE GRÁFICO

O GDT ("Graphics Development Toolkit") é o pacote gráfico da IBM utilizado no desenvolvimento de PICTÓREA. Este sistema permite a elaboração de programas independentes dos dispositivos gráficos (estações de trabalho) utilizados. Ele contém uma longa lista de funções gráficas e de textos que são explicitadas, de uma única maneira, no controle dos diversos dispositivos de entrada e saída.

O GDT consiste de um Controlador de Interface de Dispositivos Virtuais - VDI ("Virtual Device Interface"); um conjunto de dispositivos "drivers" de entrada e saída; bibliotecas de "linkagem" referentes às funções gráficas e funções de texto disponíveis, e um manual contendo informações sobre cada uma destas funções, de acordo com a linguagem de programação específica.

3.1. O controlador de interface de dispositivos virtuais - VDI

O VDI define uma linguagem comum ou protocolo que permite a comunicação entre um programa, e os diversos dispositivos "drivers" associados às suas estações de trabalho. Este protocolo consiste de um conjunto de funções e parâmetros que possibilitam a realização do software com as características desejadas.

Os dispositivos "drivers" comunicam-se diretamente com o VDI e as estações de trabalho. Estes dispositivos, quando referenciados em um programa qualquer, interagem diretamente com o sistema operacional, de uma maneira transparente ao usuário. Cada um dos dispositivos de entrada e saída é controlado por um "driver" que tem como objetivo traduzir a informação pas

sada por um programa a uma respectiva estação de trabalho.

3.1.1. Modos de operação

Uma estação de trabalho pode operar em MODO GRÁFICO ou em MODO CURSOR. Apenas um deles pode ser ativado em determinado instante.

O MODO GRÁFICO permite:

- Executar funções gráficas de entrada e saída
- Obter informações sobre a execução de determinada função (atributos)
- Utilizar primitivas gráficas para desenhos
- Utilizar textos alfanuméricos
- Controlar textos gráficos.

Nenhuma função do modo cursor deve ser referenciada no modo gráfico.

O MODO CURSOR só é aplicado em dispositivos do tipo CRT. Este modo permite:

- Apagar uma página completa, linha ou parte de uma linha
- Posicionar a saída de determinado caracter na tela
- Atribuir determinadas características, tais como: modo piscante, sublinhagem, vídeo reverso etc.

Nenhuma função gráfica deve ser referenciada no modo cursor.

3.2. O driver gráfico VDIDY004.SYS

O dispositivo gráfico utilizado na elaboração do programa PICTÓREA (versão 4 cores), é denominado IBM COLOR/GRAPHICS MONITOR ADAPTER - MEDIUM RESOLUTION 4 COLOR (VDIDY004.SYS). É um dispositivo associado a uma estação de trabalho de nome lógico DISPLAY e que representa o controlador de

um monitor de vídeo de resolução 320x200.

Este "driver", quando no modo gráfico, pode mostrar, simultaneamente, o conjunto das seguintes cores ("default"):

ÍNDICE DE CORES	CORES
0	preto ("background")
1	branco
2	magenta
3	cyan

O VDIDY004.SYS fornece, ainda, quatro paletas de cores diferentes as quais podem ser selecionadas pelo usuário. Estas paletas são:

ÍNDICE DE CORES	PALETA 1	PALETA 2	PALETA 3	PALETA 4
1	marrom	amarelo claro	cinza	branco
2	vermelho	vermelho claro	magenta	magenta claro
3	verde	verde claro	cyan	cyan claro

PICTÓREA (versão 4 cores) utiliza as cores "default", mostradas anteriormente, na limiarização dos pixels da imagem.

No modo gráfico, o sistema trabalha em coordenadas NDC ("Normalized Device Coordinates"), onde a relação entre as dimensões horizontal e vertical da imagem ("aspect ratio") é preservada.

A quantidade de memória necessária para alocar o sistema VDI é igual a 32K octetos sendo necessários mais 31K, aproximadamente, para os controla

dores gráficos.

3.3. O menu de abertura do sistema

Ao carregar o programa PICTÓREA, o usuário dispõe de um MENU DE ABERTURA contendo as seguintes opções:

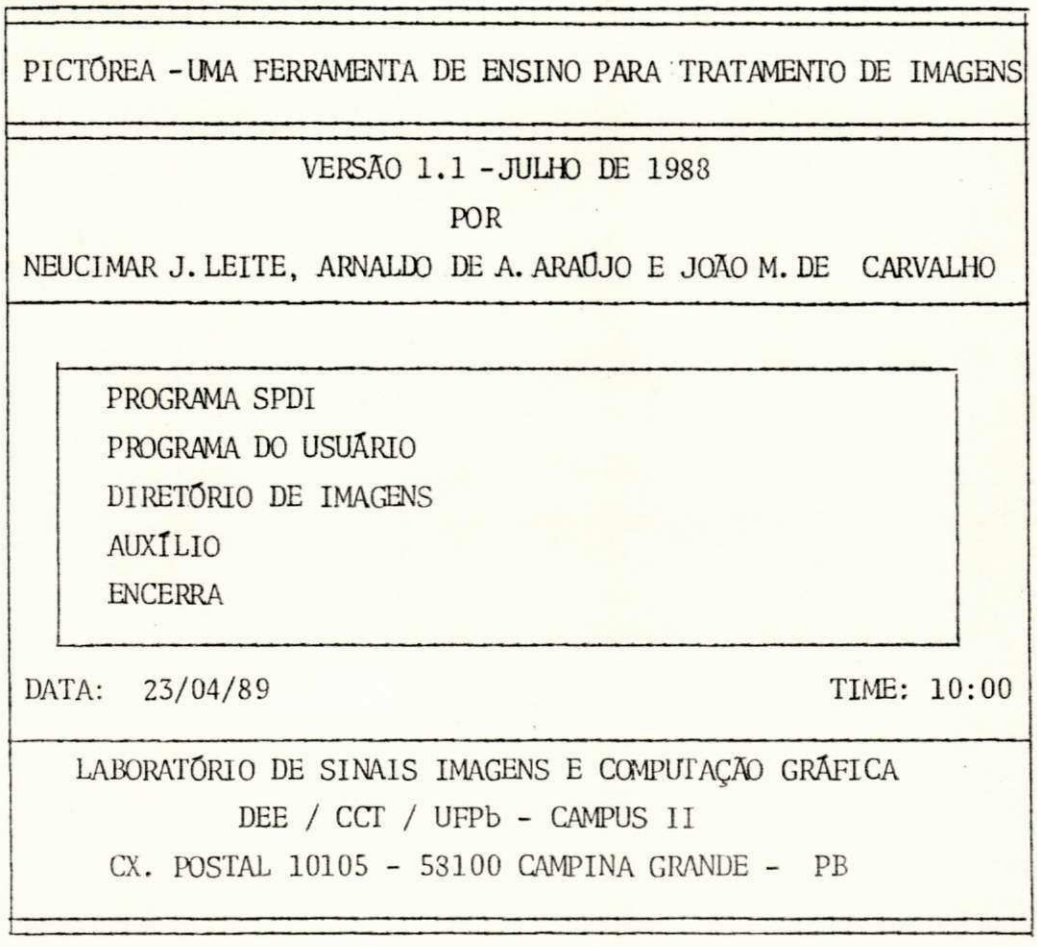


Figura 3.1. Menu de abertura

Este programa foi elaborado em linguagem PASCAL, dado a sua facilidade em acessar arquivos executáveis externos a um programa (Apêndice 2).

A escolha de uma das funções do MENU DE ABERTURA é feita através das teclas de movimento de cursor para cima <↑> ou para baixo <↓> seguidas do

comando <ENTER>.

3.3.1. Programa SPDI

Ao selecionar esta função, o sistema carrega o programa SPDI.EXE, o qual acessa todos os módulos de visualização e processamento de imagens disponíveis. Os capítulos 4 e 5 apresentam detalhadamente o conjunto das funções contidas nos 10 módulos do PROGRAMA SPDI.

3.3.2. Programa do usuário

PICTÓREA permite uma expansão da sua estrutura, oferecendo uma maneira flexível de interação sistema-usuário. Na escolha desta função, o usuário pode ter o seu próprio programa acoplado ao aplicativo. Para a linguagem FORTRAN 77, o mesmo dispõe de uma biblioteca, PICTÓREA.LIB, que o auxilia no carregamento e no armazenamento da imagem a ser processada no seu programa.

3.3.3. Diretório de imagens

Esta função mostra, na tela, todos os arquivos com extensão IMG. Estes arquivos constituem os arquivos de imagens fornecidos ao usuário, juntamente com o sistema PICTÓREA. Cada um deles contém uma imagem 64x64 codificada em 32 níveis de cinza, onde o "0" corresponde ao preto e o "31" ao branco.

3.3.4. Auxílio

Contém algumas informações à respeito das funções presentes no MENU DE ABERTURA.

3.3.5. Encerra

Finaliza a execução do programa PICTÓREA, transferindo o controle ao sistema operacional.

Explicações a nível de utilização de PICTÓREA e outras informações sobre o seu conteúdo estão registradas no MANUAL DO USUÁRIO.

3.4. Conclusão

Embora PICTÓREA utilize, na sua segunda versão, o controlador gráfico de 16 cores VDIDY00D.SYS (IMB Enhanced Graphics Adapter-Medium Resolution 16 Color), limitamo-nos, neste capítulo, exclusivamente à descrição do VDIDY004.SYS. Esses dois controladores possuem as mesmas características, exceto no que se refere ao possível conjunto das cores mostradas na tela.

O quadro abaixo contém as cores "default" empregadas na versão em 16 cores de PICTÓREA.

ÍNDICE DE CORES	CORES
0	preto
1	branco
2	vermelho
3	verde
4	azul
5	amarelo
6	cyan
7	magente
8	marrom
9	cinza claro

10	cinza escuro
11	azul claro
12	verde claro
13	cyan claro
14	vermelho claro
15	magenta claro

4. PROGRAMA SPDI - 1ª PARTE

A função PROGRAMA SPDI, contida no MENU DE ABERTURA, executa as funções de visualização e processamento digital de imagens do sistema.

Os módulos do MENU PRINCIPAL, do PROGRAMA SPDI, são os seguintes:

MENU PRINCIPAL

1. Entrada/saída de imagens
2. Display de imagens
3. Histogramas e estatísticas
4. Aritmética de imagens
5. Transformações radiométricas
6. Filtros espaciais
7. Geração de imagens
8. Geração de ruídos.

Por questões de organização, os módulos de 1 a 5 serão apresentados neste capítulo; o módulo 6, no capítulo 5, e os módulos 7 e 8, no capítulo subsequente.

4.1. Módulo: Entrada/saída de imagens

Este módulo contém as seguintes funções:

ENTRADA/SAÍDA DE IMAGENS

F1 Carrega imagem

F2 Armazena imagem

F3 Imprime imagem

F4 Transfere imagens entre MI's

F9 Auxílio

F10 Retorna

F1 realiza a transferência de um arquivo de imagem, em disco, para uma área de memória do computador; F2 executa o processo inverso, transferindo um arquivo de imagem da memória RAM para um arquivo em disco, e F3 imprime imagens com 32 níveis de cinza através da superposição dos seguintes caracteres [7].

```
MMMMMMHHHHHHXHXOZWMNOS=I*++=:-.-
WWWWWW###*++-----      =  -   -
####OO+-
OOO
+
```

Figura 4.1. Caracteres usados para a impressão dos 32 níveis de cinza da imagem

Os caracteres de uma coluna, quando superpostos, produzem os níveis de cinza mostrados abaixo:

Figura 4.2. Níveis de cinza gerados a partir da superposição dos caracteres da Figura 4.1.

Para a impressão das imagens, o usuário pode escolher uma dentre as 4 funções de translação da escala de cinza:

ESCALA	FUNÇÃO
Linear	$f(NC) = NC$
Raiz-quadrática	$f(NC) = 1/31*(NC)^2$
Logarítmica	$f(NC) = (31*\log_{10}(NC+1))/\log_{10}32$
Absorção	$f(NC) = 31*e^{-((1-NC/31)*\log_{10}31)}$

Tabela 4.1. Funções de impressão

NC representa a intensidade do pixel no ponto x,y da imagem. O Apêndice 1 contém imagens impressas de acordo com as funções apresentadas na Tabela 4.1.

A função F4, deste módulo, transfere imagens entre as memórias de imagens MI1, MI2 e MI3, permitindo que processos iterativos sejam aplicados sem que a imagem processada tenha que ser armazenada em disquete e carregada de volta ao sistema.

4.1.1. Módulo: Display de imagens

Este módulo é responsável pela visualização das imagens armazenadas nas MI's. Para o "display" destas imagens, utiliza-se o cartão gráfico disponível nas máquinas do tipo IBM-PC ou compatíveis. Estas imagens são apresentadas com redução da escala de cinza de 32 níveis para apenas 4 níveis ou cores (PICTÓREA: versão 4 cores) ou para 16 cores (PICTÓREA: versão 16 cores). Pode-se obter o "display" de até 2 imagens numa mesma tela sendo possível, por exemplo, uma comparação visual entre uma imagem original e uma imagem processada. Cada uma delas pode ser acompanhada de um tí

tulo de até 14 dígitos alfanuméricos que pode servir para identificá-las.

Para agilizar o processo de visualização, DISPLAY DE IMAGENS está presente nos módulos: ARITMÉTICA DE IMAGENS, TRANSFORMAÇÕES RADIOMÉTRICAS, FILTROS ESPACIAIS, GERAÇÃO DE IMAGENS e GERAÇÃO DE RUÍDOS. Este módulo não mais será mencionado nos próximos itens.

4.1.2. Módulo: Histogramas e estatísticas

Este módulo permite um estudo qualitativo das imagens, através de parâmetros estatísticos fornecidos ao usuário. As seguintes funções estão disponíveis:

HISTOGRAMAS E ESTATÍSTICAS

F1 Display de histograma global

F2 Display de perfil de linha ou coluna

F3 Estatísticas locais

F4 Verifica e/ou modifica pixels

F9 Auxílio

F10 Retorna

A função F1 apresenta o histograma global da imagem, dando informações sobre a distribuição da intensidade dos pixels na imagem digitalizada (Fig. 4.3). O histograma é representado pelo gráfico dos níveis de cinza, presentes na imagem, em função da quantidade dos elementos pertencentes a cada um desses níveis (freqüência de ocorrência). O conteúdo de um histograma apresenta uma descrição geral do aspecto de uma cena. Ele especifica o número de pixels para cada nível de cinza, sem indicar a sua localização na imagem. O histograma é único para determinada cena; no entanto, cenas

diferentes podem ter o mesmo histograma.

F2 apresenta, graficamente, a intensidade dos elementos contidos numa linha ou coluna de uma imagem. O eixo das abcissas contém o número de linhas ou colunas, e o eixo das ordenadas, os respectivos níveis de cinza. A Figura 4.4 mostra o perfil da linha 32 de uma das imagens disponíveis no sistema.

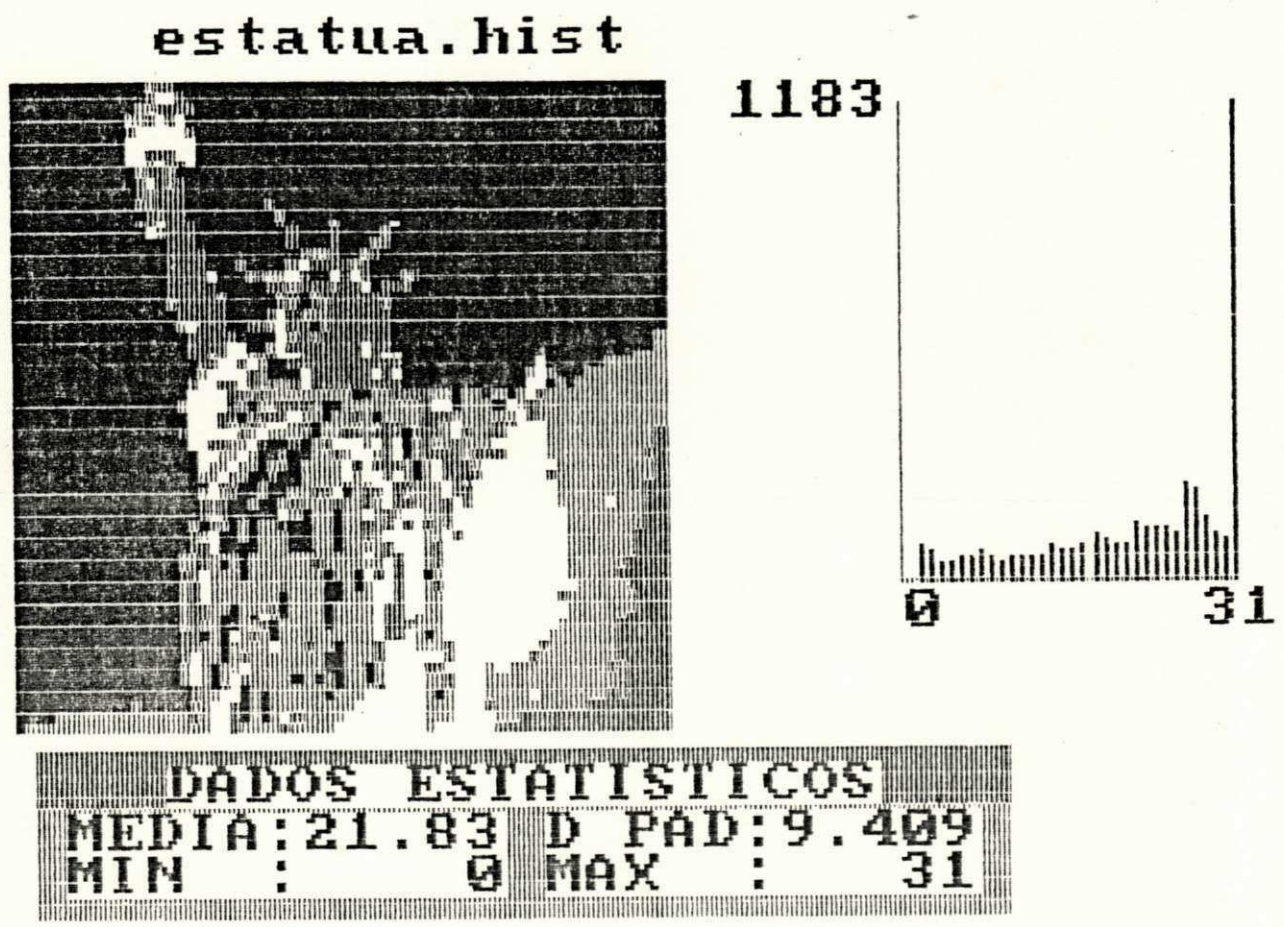


Figura 4.3. Histograma geral de uma imagem

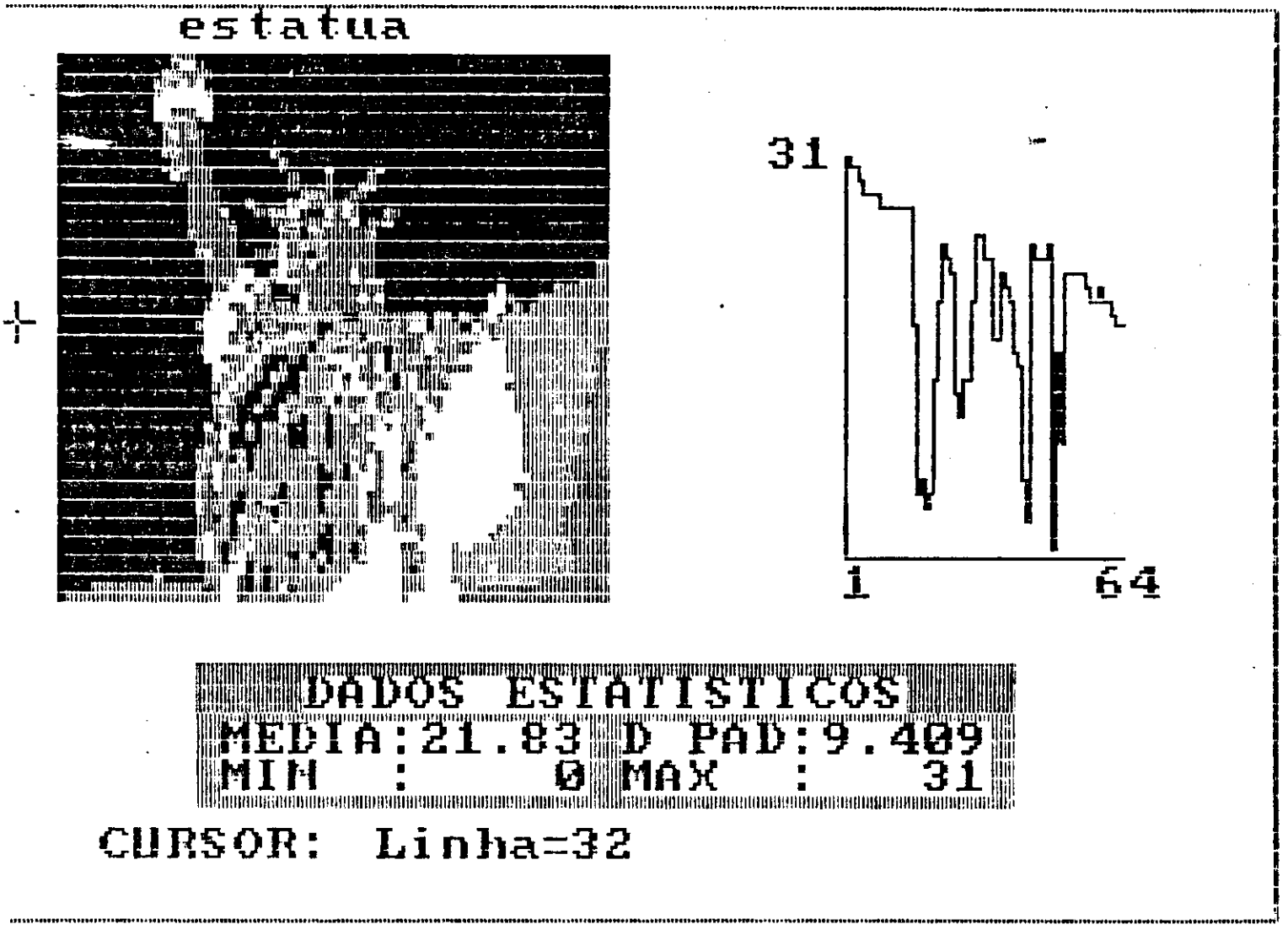


Figura 4.4. Perfil de linha de uma imagem

A função F3 deste módulo fornece alguns dados estatísticos da cena, tais como: média, desvio-padrão, valores mínimo (MIN) e máximo (MAX) dos pixels contidos numa janela móvel e de tamanho variável (4 x 4, 8 x 8 ou 16 x16), além de apresentar o seu histograma local (Fig. 4.5).

F4 permite a visualização dos níveis de cinza presentes na cena dos elementos contidos numa janela móvel, de tamanho 5x5 (Fig. 4.6), bem como a modificação destes elementos. Esta função permite que o usuário altere todo o conteúdo de uma memória de imagem, podendo mesmo criar uma nova cena.

4.1.4. Módulo: Aritmética de imagens

Este módulo realiza as seguintes operações em imagens:

ARITMÉTICA DE IMAGENS

F1 Adição de MI1 e MI2

F2 Subtração de 2 imagens

F3 Diferença absoluta entre MI1 e MI2

F4 Média da diferença absoluta e erro médio quadrático entre 2 imagens

F5 Efeito zoom

F6 Redução de imagem

F7 Display de imagens

F9 Auxílio

F10 Retorna

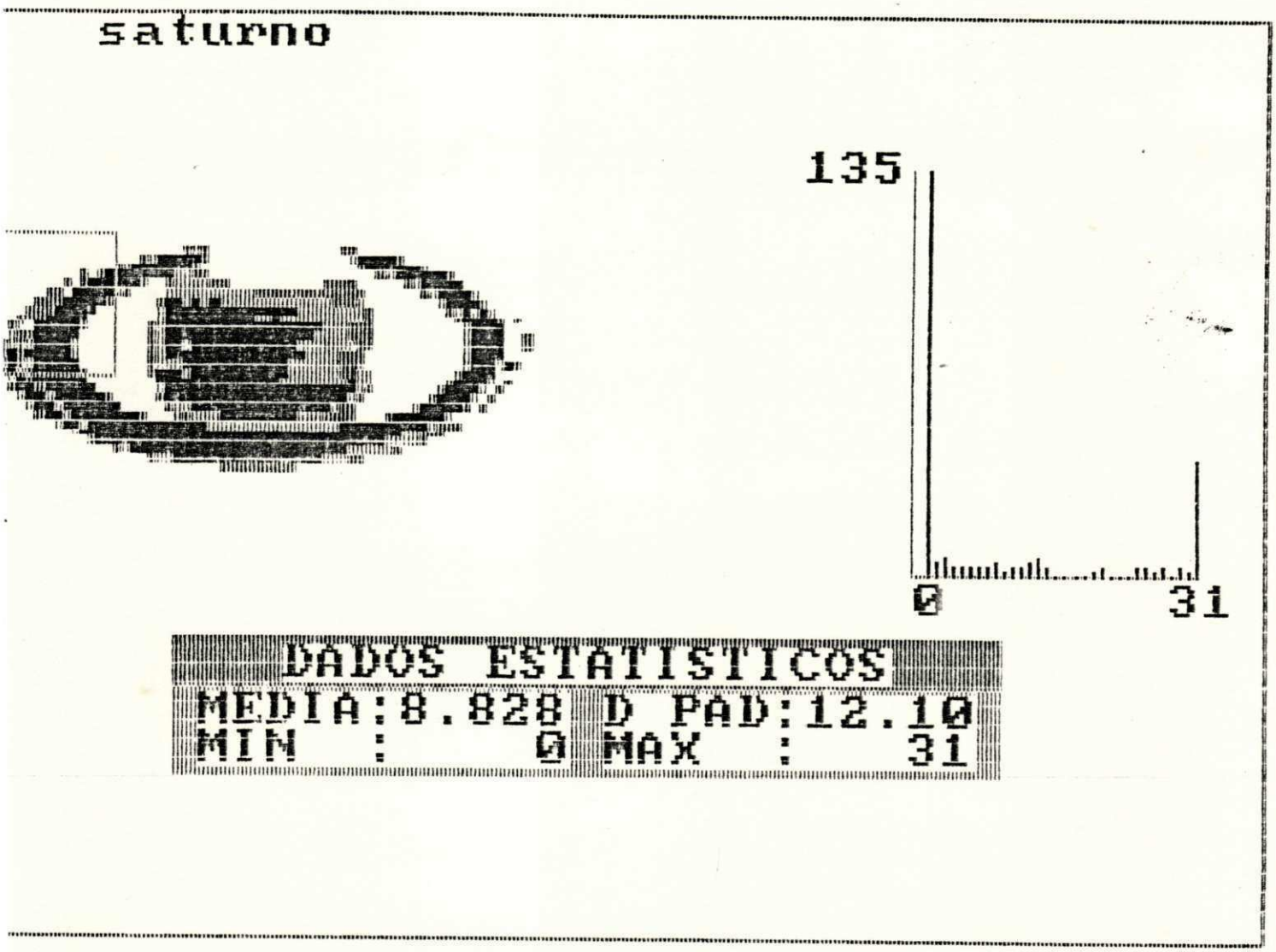


Figura 4.5. Histograma local de uma janela 16x16

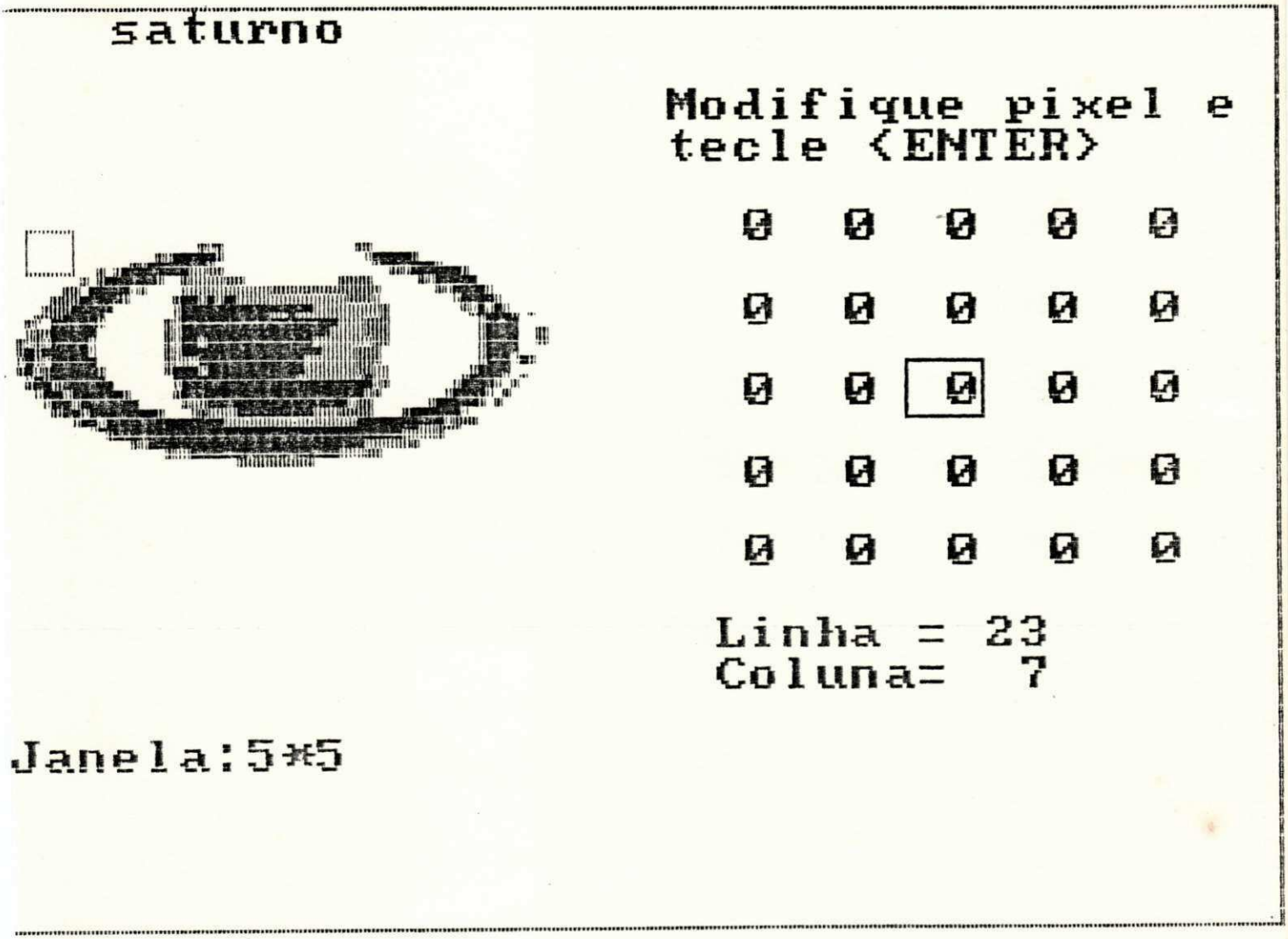


Figura 4.6. Visualização dos níveis de cinza de uma área da imagem

As funções F1 e F2 deste módulo executam as operações pontuais de adição e subtração, respectivamente, armazenando o resultado em MI3. A adição pode ser utilizada, por esemplo, para sobrepor o conteúdo de uma imagem sobre outra, enquanto a subtração pode servir para detectar mudanças entre duas imagens de uma mesma cena.

A diferença absoluta entre MI1 e MI2, função F3, apresenta-se como uma medida visual da fidelidade de uma imagem processada em relação à imagem original. A média da diferença absoluta (MDA) e o erro médio quadrático (EMQ), entre 2 imagens, apresentam-se como parâmetros qualitativos desta fidelidade.

Os valores do EMQ e da MDA são dados por:

$$EMQ = \frac{1}{N^2} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} [g(x,y)-f(x,y)]^2 \qquad \text{Eq. 4.1}$$

$$MDA = \frac{1}{N^2} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} |g(x,y)-f(x,y)| \qquad \text{Eq. 4.2}$$

onde g(x,y) é a imagem processada, f(x,y) é a imagem original e N = 64.

A função F5 realiza uma ampliação de um segmento da imagem selecionado por uma janela móvel de dimensão 16x16 pixels, dando origem ao que chamamos efeito "zoom". Dois fatores possíveis de ampliação: 2x e 4x. Esta operação resulta no aumento ou ampliação da área da janela, de acordo com o fator definido pelo usuário. O resultado do processamento é armazenado em MI3. A Figura 4.7 mostra o "zoom" (fator igual a 4) da área indicada na imagem.

A função F6 realiza uma redução de 2x ou de 4x sobre toda a imagem 64x64, resultando em imagens de tamanho 32x32 ou 16x16 pixels, respectiva

mente (Fig. 4.8).

4.1.4. Módulo: Transformações radiométricas

Este módulo do programa apresenta um conjunto de funções que modificam o valor dos níveis de cinza dos pixels da imagem. Estas transformações são do tipo pontual, em que o valor do pixel de saída depende apenas do valor do pixel de entrada correspondente, e cujo objetivo é enfatizar alguma característica de interesse de uma cena (REALCE).

Este bloco compreende as seguintes opções:

TRANSFORMAÇÕES RADIOMÉTRICAS

F1 Equalização histogrâmica

F2 Mapeamento dos níveis de cinza

F3 Display de imagens

F9 Auxílio

F10 Retorna

Apesar da sua simplicidade, as operações pontuais abrangem importantes técnicas de processamento de imagens.

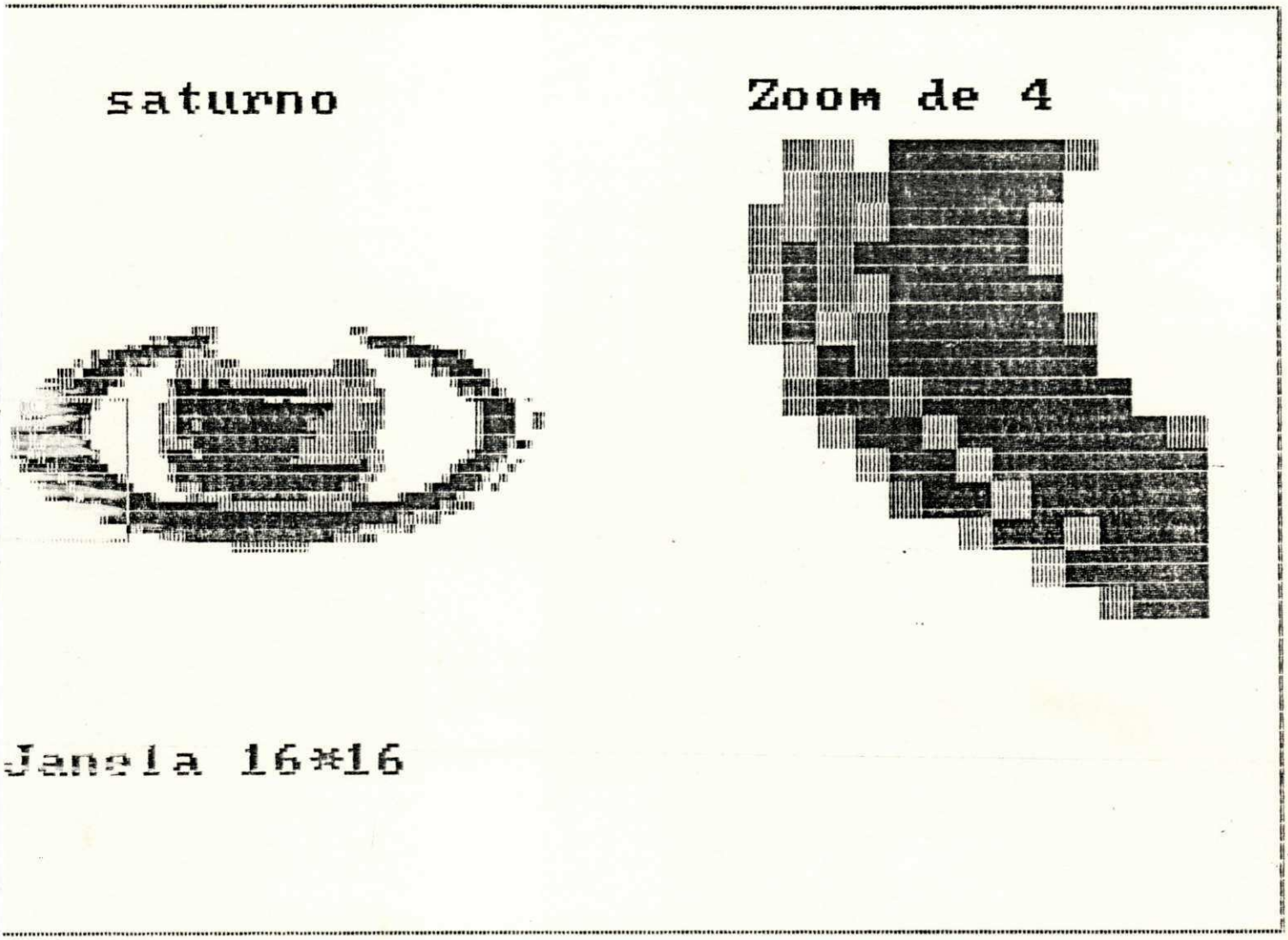


Figura 4.7. Zoom de uma área de imagem

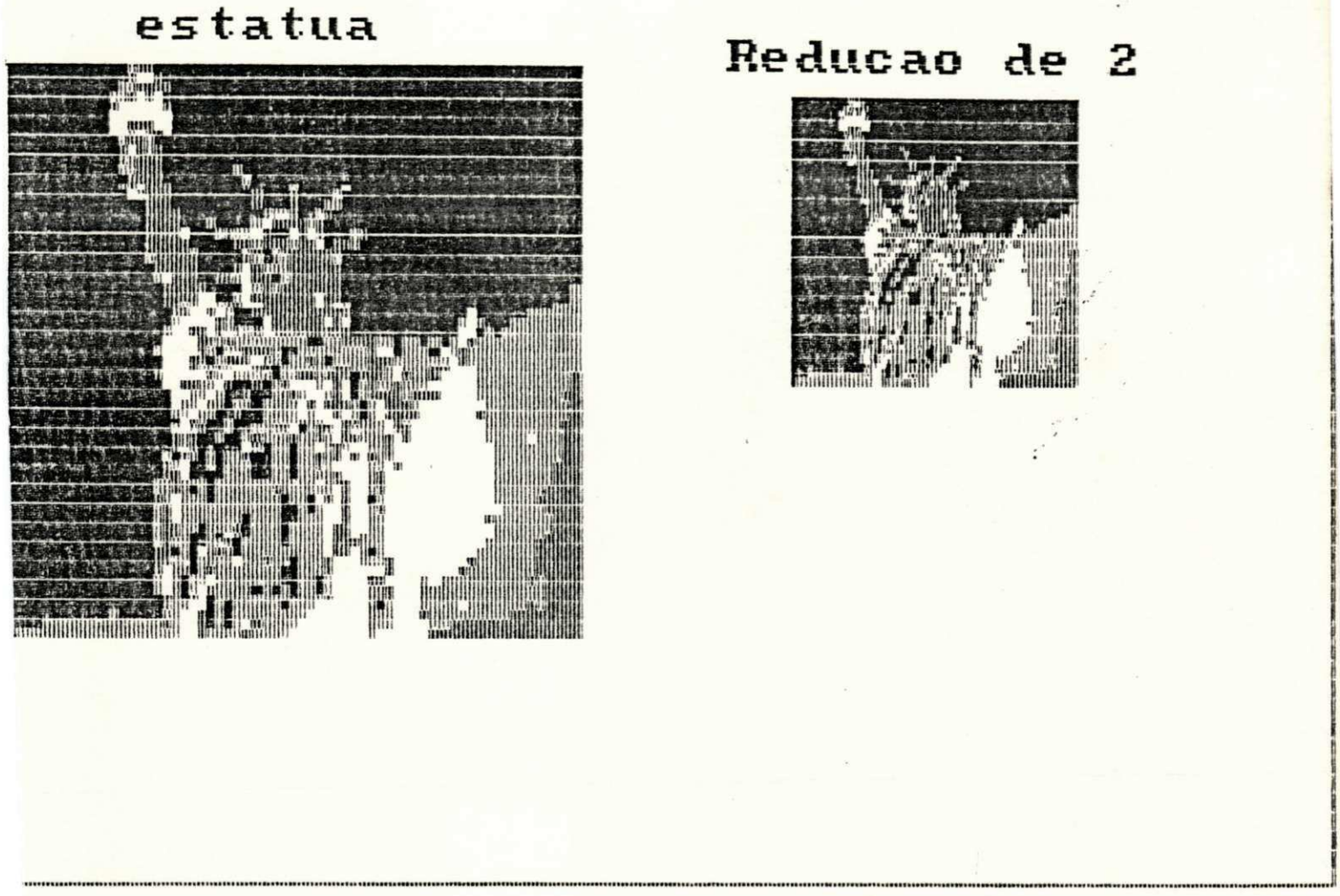


Figura 4.8. Redução da imagem

Quando uma operação pontual qualquer gera uma imagem de saída a partir de uma imagem de entrada, deve existir uma correspondência entre os pontos nas duas imagens, onde cada pixel da imagem de saída está associado a um pixel da imagem de entrada. As transformações radiométricas, vistas neste bloco, são aquelas em que os níveis de cinza das imagens são modificados, sem que ocorra uma alteração da geometria espacial.

A função F1 executa a transformação histogrâmica dando origem a uma segunda imagem com uma distribuição mais uniforme dos níveis de cinza [7].

Considerando uma variável **r** como sendo o nível de cinza dos pixels de uma imagem a ser processada, e assumindo que os seus valores são normalizados, então:

$$0 \leq r \leq 1 \qquad \text{Eq. 4.3}$$

onde **r** = 0 representa o preto e r = 1, o branco, na escala de cinza.

Para qualquer **r** no intervalo [0,1] desejamos obter a seguinte transformação:

$$s = T(r) \qquad \text{Eq. 4.4}$$

a qual produz um nível de cinza **s** para cada valor de **r** na imagem original. Esta função deve satisfazer as seguintes condições:

1. T(r) está associado a um único valor e é monotonicamente crescente no intervalo $0 \leq r \leq 1$
2. $0 \leq T(r) \leq 1$ para $0 \leq r \leq 1$

Desta forma, preservamos o intervalo de mapeamento entre **r** e **s**, e a ordem da escala de cinza, que vai do preto (nível 0) ao branco (nível 1).

A função de transformação inversa é dada por

$$r = T^{-1}(s) \quad 0 \leq s \leq 1 \qquad \text{Eq. 4.5}$$

com T^{-1} satisfazendo as condições 1 e 2.

Assumindo, por um momento, que os níveis de cinza da imagem são variáveis contínuas no intervalo normalizado [0,1] os níveis de cinza das duas imagens, original e transformada, podem ser caracterizados por suas funções densidade de probabilidade $p_r(r)$ e $p_s(s)$, respectivamente.

Se $p_r(r)$ e $T(r)$ são conhecidos e $T^{-1}(s)$ satisfaz a condição 1, então a função densidade de probabilidade dos níveis de cinza transformados é dada por:

$$p_s(s) = [p_r(r)dr/ds]_{r=T^{-1}(s)} \qquad \text{Eq. 4.6}$$

A equalização histogrâmica realiza a transformação da imagem, controlando a função densidade de probabilidade dos níveis de cinza através da função de transformação $T(r)$.

Considere a seguinte função de distribuição cumulativa (FDC) de r:

$$s = T(r) = \int_0^r p_r(w)dw \qquad 0 \leq r \leq 1 \qquad \text{Eq. 4.7}$$

Esta função satisfaz as condições 1 e 2 vistas anteriormente, onde a FDC cresce monotonicamente de 0 a 1 em função de **r**.

Da equação acima podemos obter:

$$ds/dr = p_r(r) \qquad \text{Eq. 4.8}$$

Substituindo este valor na Eq. 4.6 temos:

$$p_s(s) = [\, p_r(r)/p_r(r) \,]_{r = T^{-1}(s)}$$

$$p_s(s) = [\, 1 \,]_{r = T^{-1}(s)}$$

$$p_s(s) = 1 \qquad 0 \leq s \leq 1 \qquad \text{Eq. 4.9}$$

ou seja, $p_s(s)$ apresenta uma densidade uniforme no intervalo **s**. Isto significa que se usarmos uma transformação T(r) igual à distribuição cumulativa de **r**, teremos uma imagem cujos níveis de cinza apresentam densidade uniforme.

No plano discreto, a probabilidade de ocorrência do k-ésimo nível de cinza é dado por:

$$p_r(r_k) = n_k/n \qquad 0 \leq r_k \leq 1 \qquad k = 0,1 \ldots N-1 \qquad \text{Eq. 4.10}$$

onde n é o número de níveis de cinza, n_k é o número de vezes que este nível aparece na imagem e n é o número total de pixels da imagem.

A fórmula discreta para a função de transformação (Eq. 4.7) é:

$$s_k = T(r_k) = \sum_{j=0}^{k} nj/n \qquad 0 \leq r_k \leq 1 \qquad k = 0,1 \ldots N-1$$

$$s_k = \sum_{j=0}^{k} p_r(r_j) \qquad \text{Eq. 4.11}$$

A equalização histogrâmica pode ser obtida diretamente através da Eq. 4.11. A Figura 4.9 mostra uma imagem com seu histograma uniformizado pelo processo descrito acima.

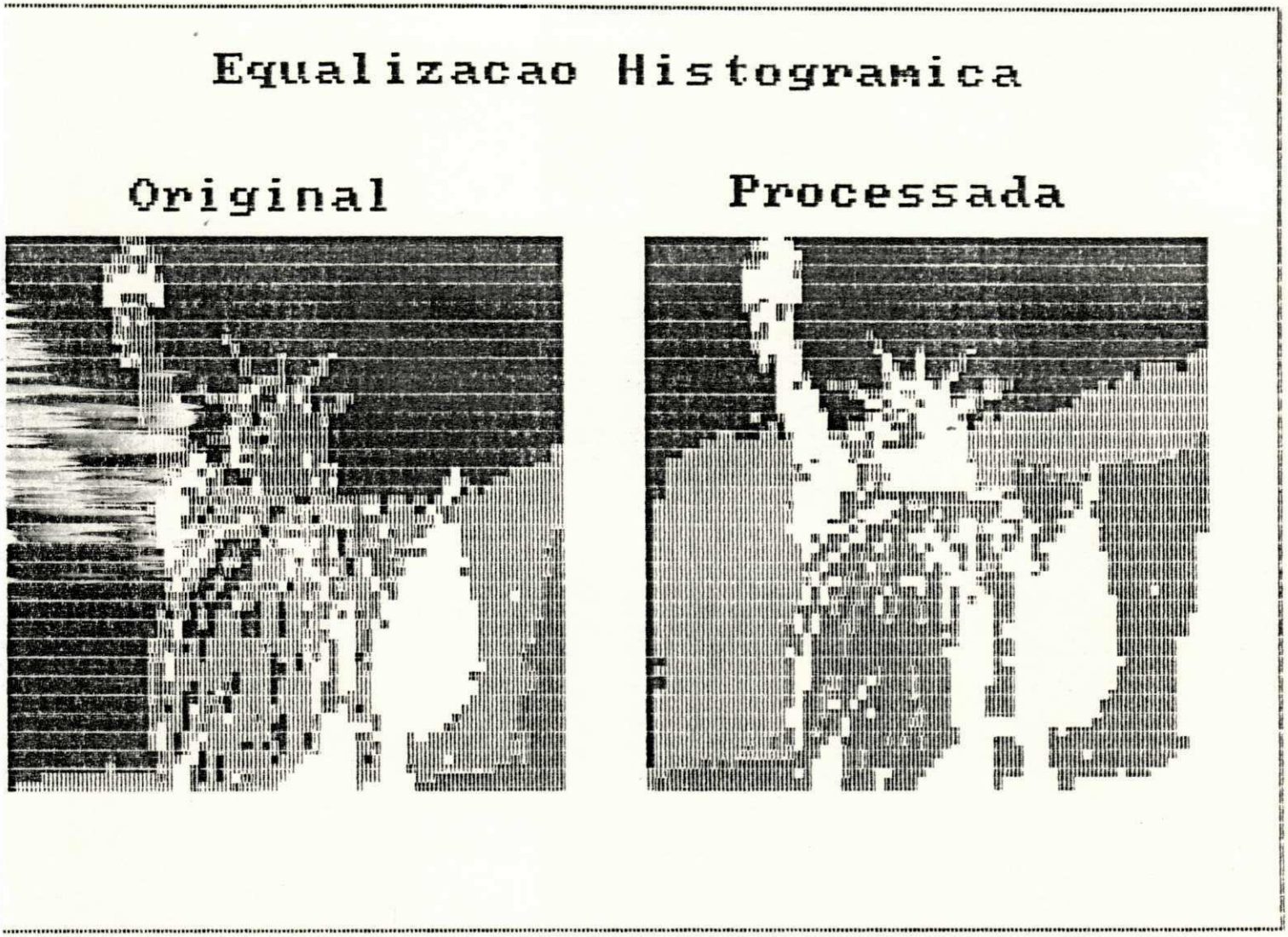


Figura 4.9. Equalização histogrâmica de uma imagem

4.1.4.1. Submódulo I:/Transformações radiométricas/mapeamento dos níveis de cinza

O item F2, do módulo TRANSFORMAÇÕES RADIOMÉTRICAS, oferece um conjunto de funções de mapeamento que realçam ou não determinadas características da imagem, dependendo da função e dos parâmetros especificados. As funções contidas neste submódulo são:

MAPEAMENTO DOS NÍVEIS DE CINZA

F1 Fatiamento em dois níveis

F2 Compressão

F3 Compressão/expansão monotônica

F4 Fatiamento por plano com fundo

F5 Aumento linear do contraste

F6 Fatiamento por plano

F7 Inversão da escala de cinza

F8 Dente de serra 3-ciclos

F9 Retorna

OUTRA TECLA Menu principal

Num mapeamento direto dos níveis de cinza, o valor dos pixels de entrada (r) varia de 0 a L-1, e o de saída (s), de 0 a M-1. Em geral, M é igual a L (número de níveis de cinza da imagem) [2]. A função de mapeamento pode ser dada por uma expressão aritmética, como as utilizadas neste trabalho, e que servem para descrever o conjunto das funções abordadas acima (M=L=32). São elas:

1. Fatiamento em dois níveis (Fig. 4.10)

$$s = 0 \quad \text{se} \quad r \leq a \qquad \text{Eq. 4.12}$$
$$s = M-1 \quad \text{se} \quad r > a$$

onde a é o nível de cinza indicado pelo usuário e que define a faixa de fatiamento. A função acima executa uma limiarização ou classificação, dando origem a uma imagem de saída com apenas dois níveis: 0 e M-1.

2. Compressão (Fig. 4.11)

$$s = a + r(b-a)/(L-1) \qquad \text{Eq. 4.13}$$

onde **a** e **b** representam o intervalo de compressão definido pelo usuário. Esta função "comprime" os níveis de cinza de 0 a L-1 para o intervalo [a,b].

3. Compressão/expansão monotônica (Fig. 4.12)

$$s = r/2 \quad \text{se } r \leq 10 \qquad \text{Eq. 4.14}$$
$$s = (2r - 15) \quad \text{se } 11 \leq r \leq 21$$
$$s = r/2 + 15 \quad \text{se } 22 \leq r \leq 31$$

A função acima divide o intervalo L-1 em três regiões, realizando compressão (nos intervalos 1 e 3) e expansão (intervalo 2) sobre a imagem de entrada.

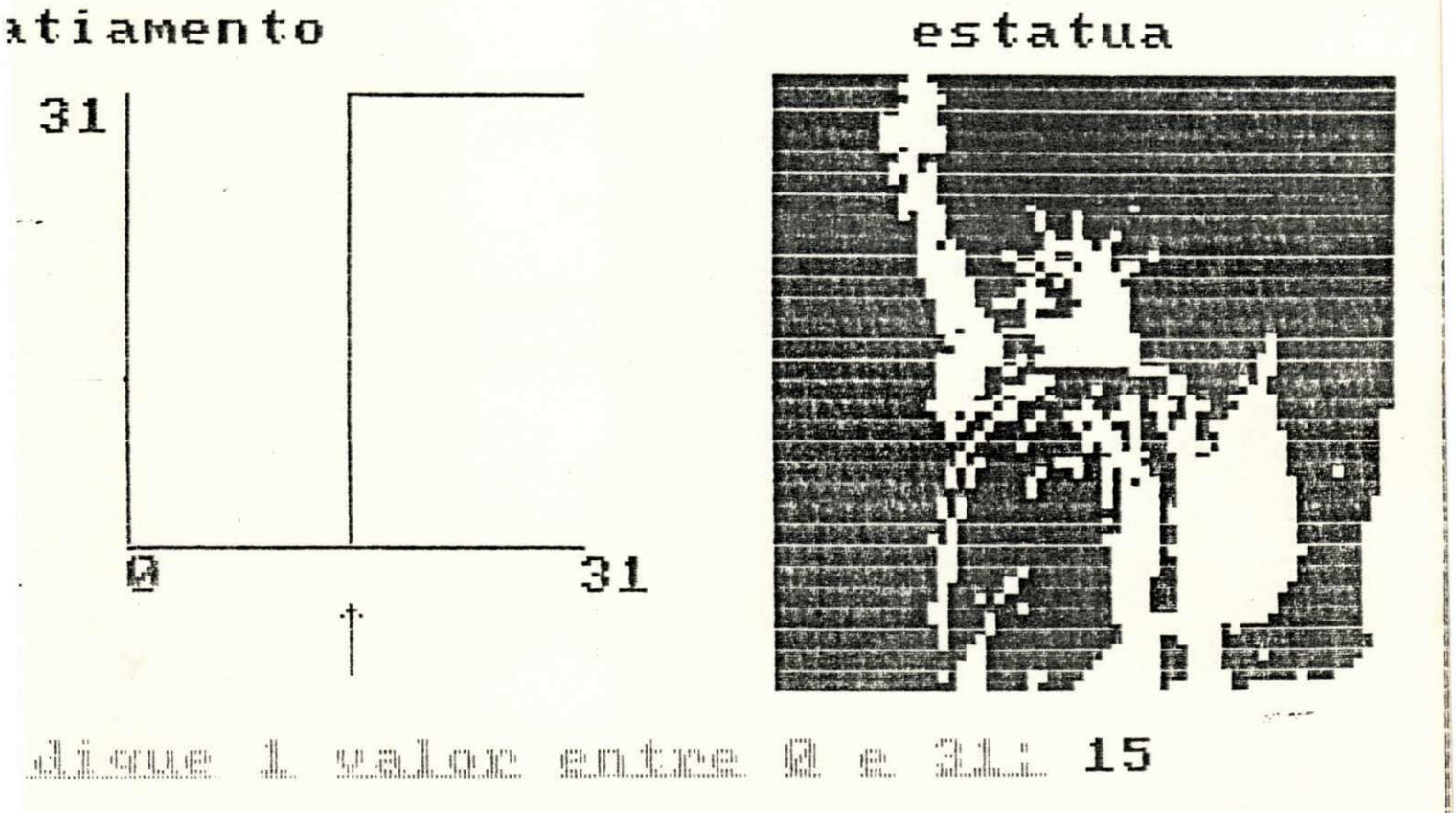


Figura 4.10. Fatiamento em dois níveis

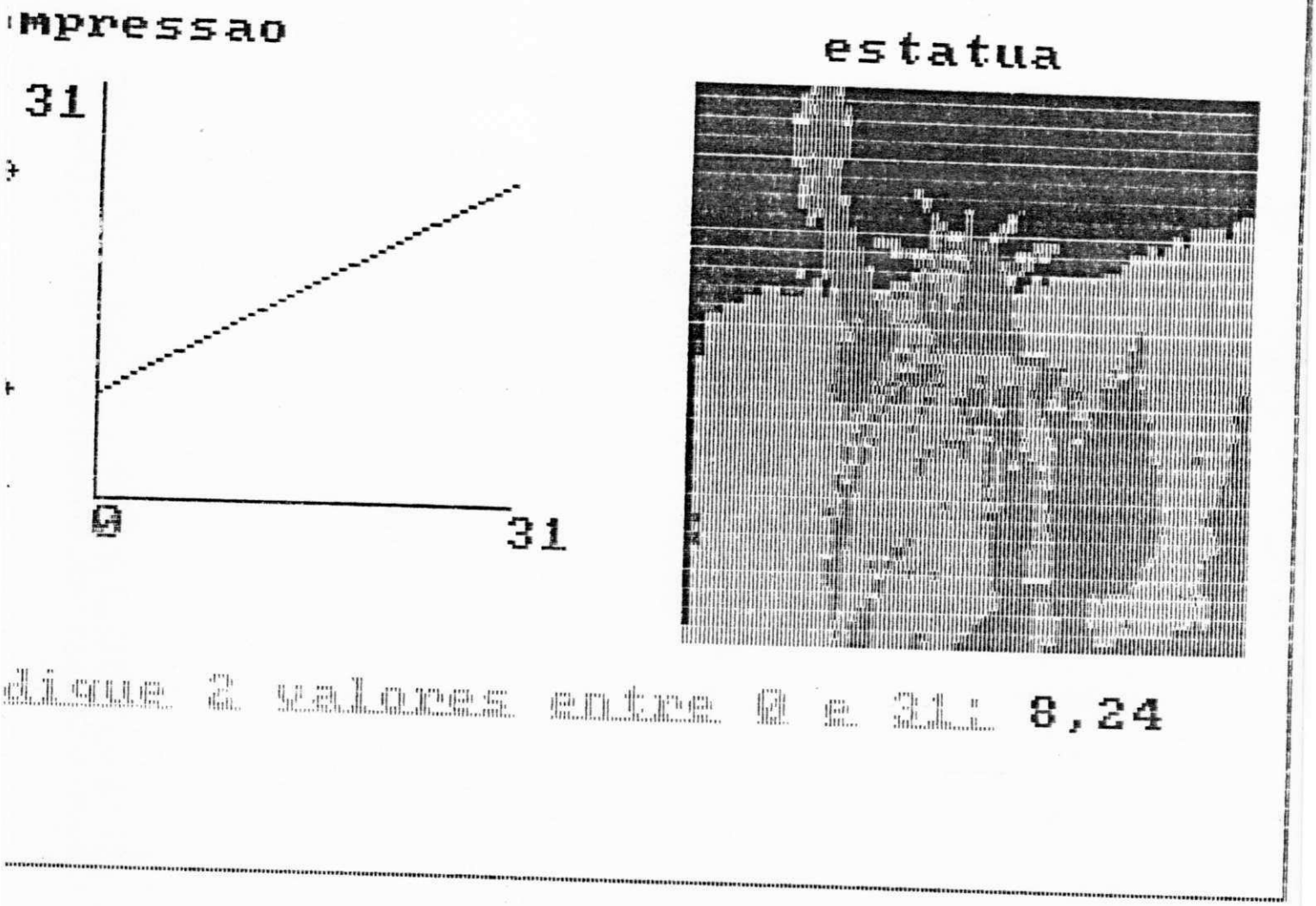


Figura 4.11. Compressão

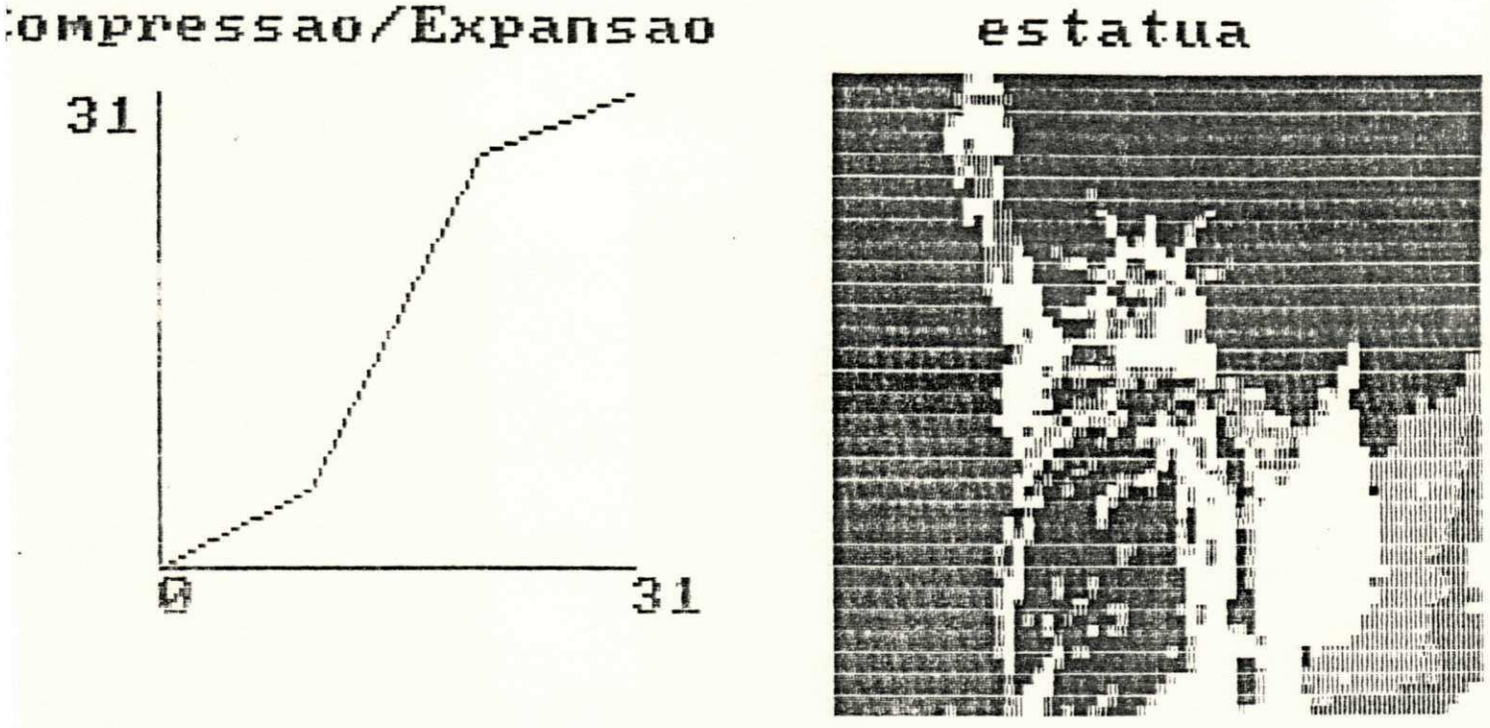


Figura 4.12. Compressão e expansão monotônica

4. Fatiamento por plano com fundo (Fig. 4.13)

$s = r$ se $r \leq a$ e $r \geq b$ Eq. 4.15

$s = M-1$ se $a < r < b$

onde **a** e **b** constituem o intervalo de fatiamento definido pelo usuário. Esta função mapeia os pontos da imagem contidos no intervalo [a,b] para o nível de cinza M-1, preservando os demais pixels fora deste intervalo.

5. Aumento linear do contraste (Fig. 4.14)

$s = 0$ se $r \leq a$ Eq. 4.16

$s = (M-1)(r-a=/(b-a)$ se $a <r< b$

$s = M-1$ se $r \geq b$

Esta função realça as variações dentro do intervalo [a,b], indicado pelo usuário, e satura em 0 os valores menores que **a**, e em M-1 os valores maiores que **b**.

6. Fatiamento por plano (Fig. 4.15)

$s = 0$ se $r \leq a$ e $r \geq b$ Eq. 4.17

$s = M-1$ se $a < r < b$

Esta função realça os pontos da imagem contidos no intervalo [a,b] indicado pelo usuário. A imagem resultante tem apenas dois níveis de cinza: 0 e M-1.

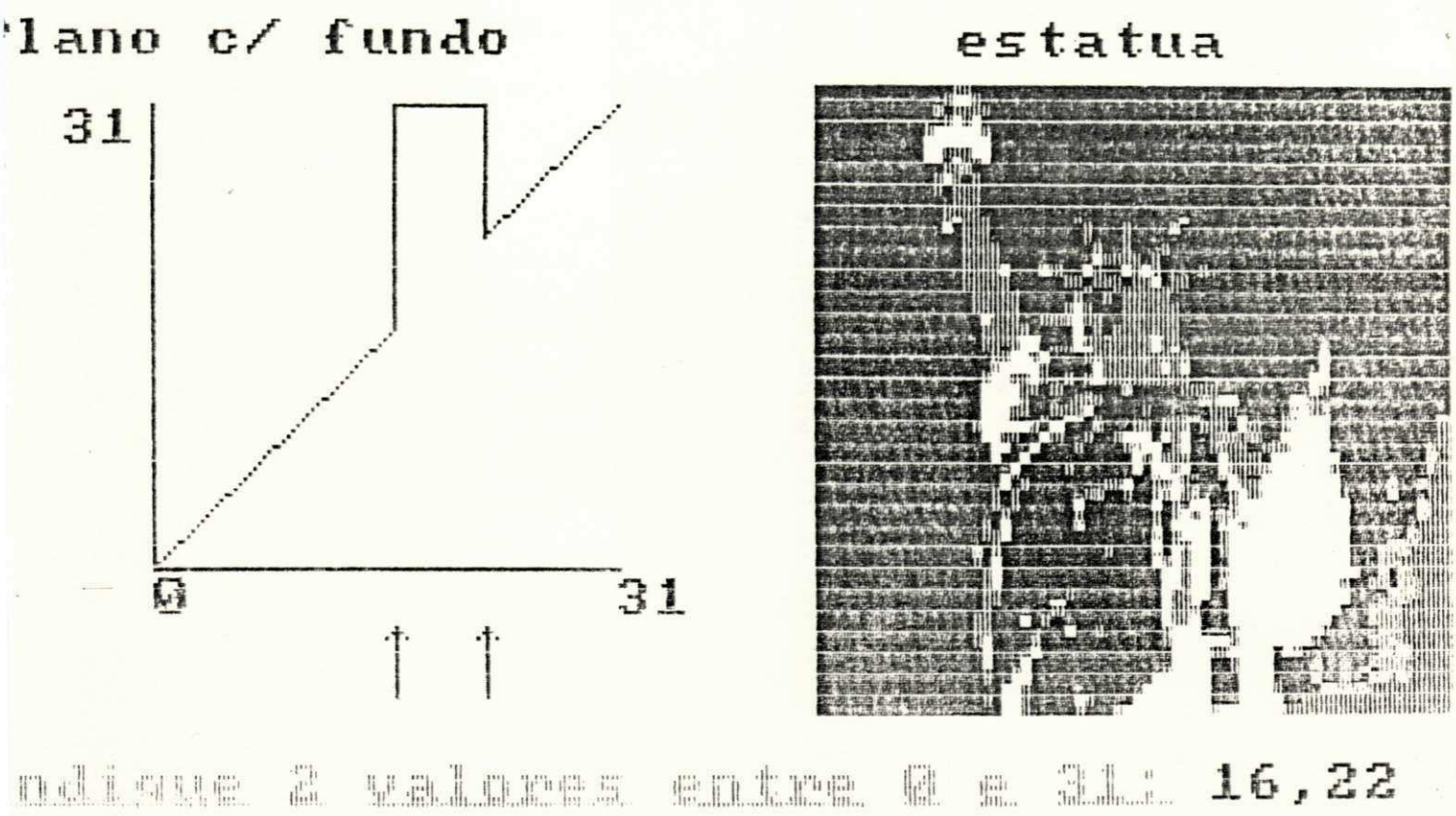


Figura 4.13. Fatiamento por plano com fundo

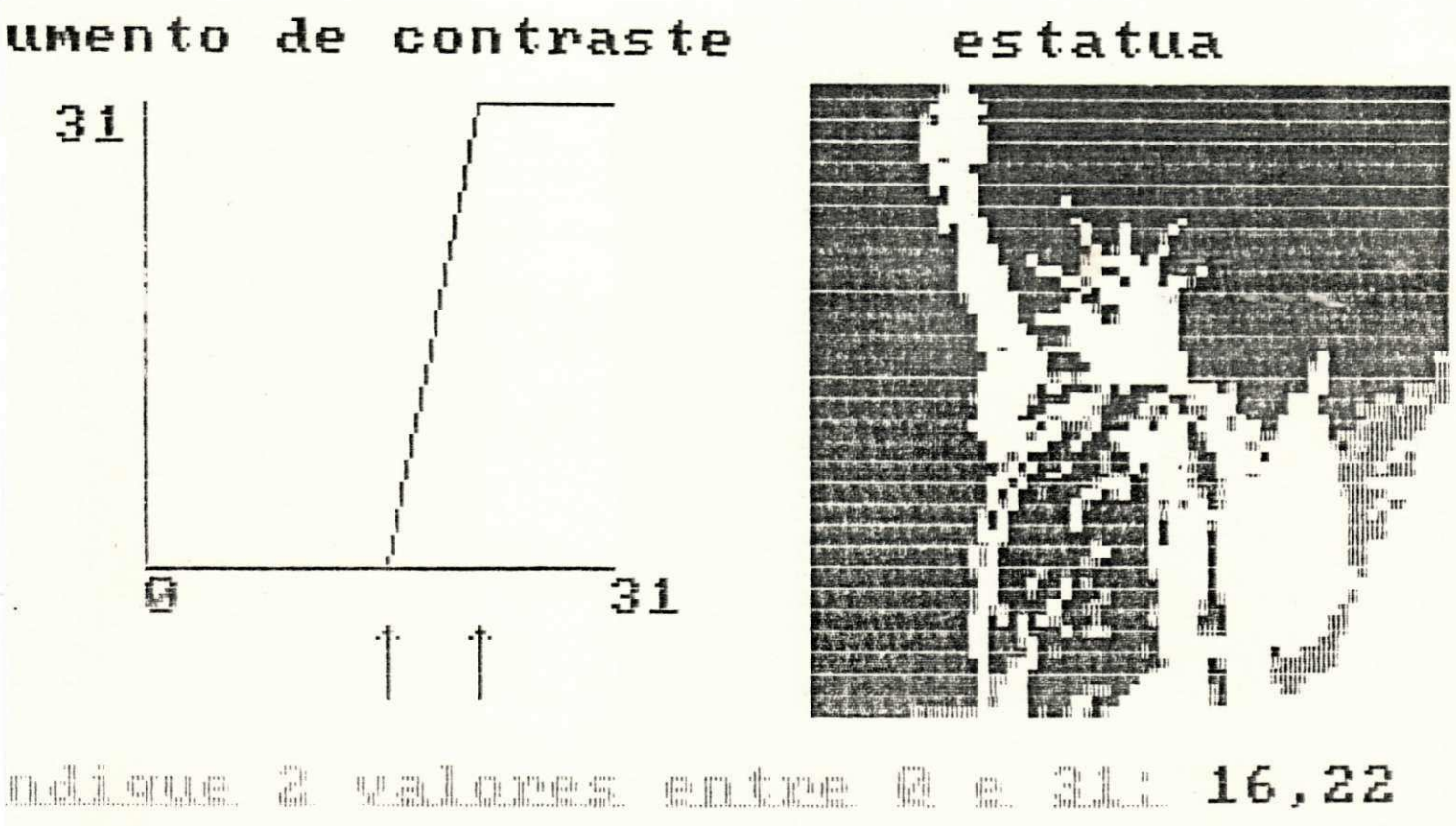


Figura 4.14. Aumento linear do contraste

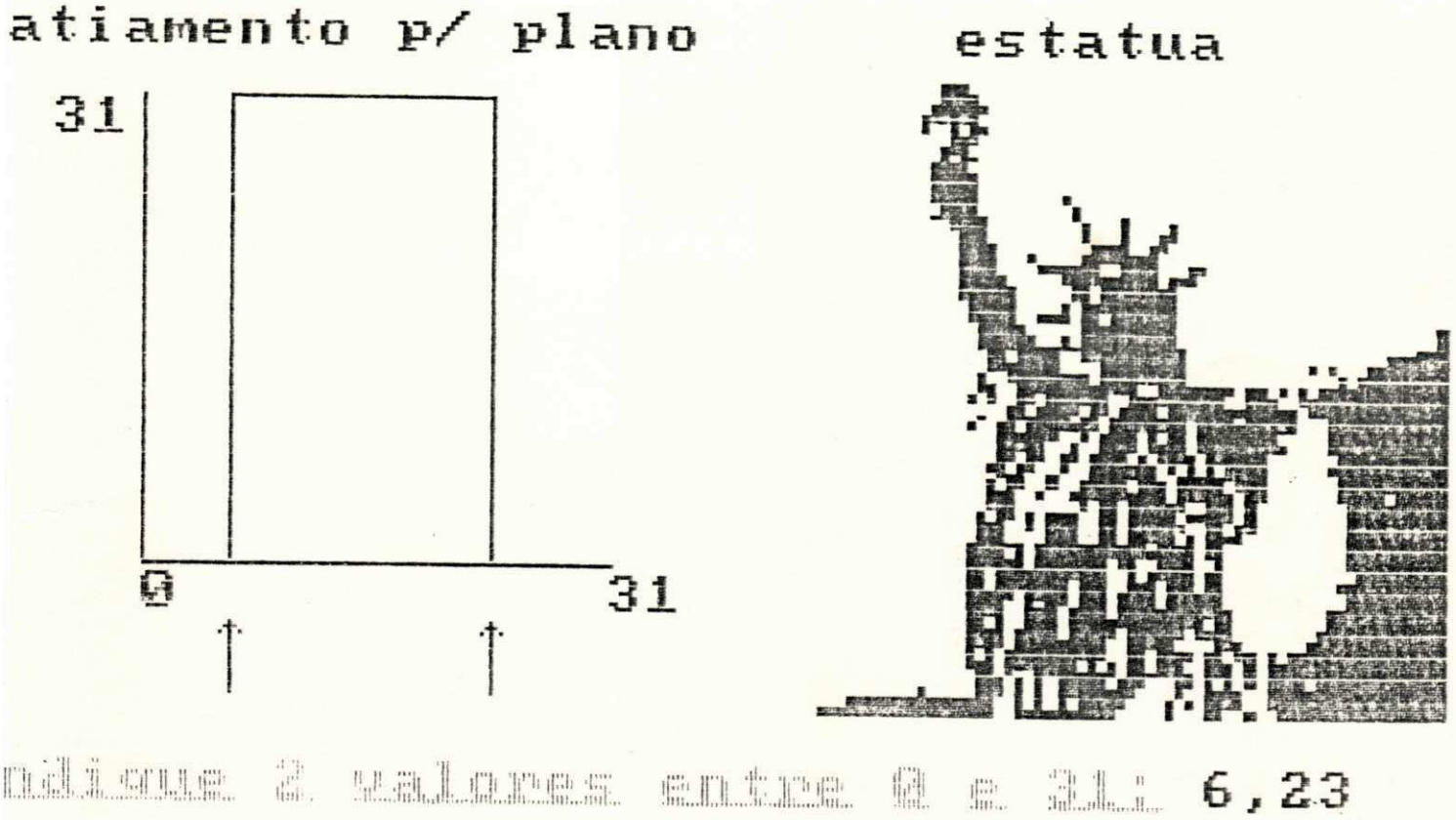


Figura 4.15. Fatiamento por plano

7. Inversão da escala de cinza (Fig. 4.16)

$$s = (M-1) - r \quad \text{para} \quad 0 \leq r \leq L-1 \qquad \text{Eq. 4.18}$$

A função acima realiza o mapeamento dos níveis de cinza na ordem inversa da intensidade dos pixels. O nível de cinza mais escuro na imagem de entrada corresponde ao nível mais claro na imagem de saída e vice-versa

8. Dente de serra 3-ciclos (Fig. 4.17)

$$\begin{aligned} s &= 3r && \text{se } 0 \leq r \leq 10 \qquad \text{Eq. 4.19} \\ s &= 3(r-10) && \text{se } 10 < r \leq 20 \\ s &= 3(r-20) && \text{se } 20 < r \leq L-1 \end{aligned}$$

Esta função realiza o mapeamento dos níveis de cinza, dividindo a faixa de 0 a L-1, dos pontos da imagem de entrada, em três intervalos, de acordo com a Eq. 4.19.

As funções 7 e 8 são utilizadas quando se deseja compensar o efeito da não linearidade do monitor de vídeo, em que os níveis de cinza tendem para um tom mais escuro.

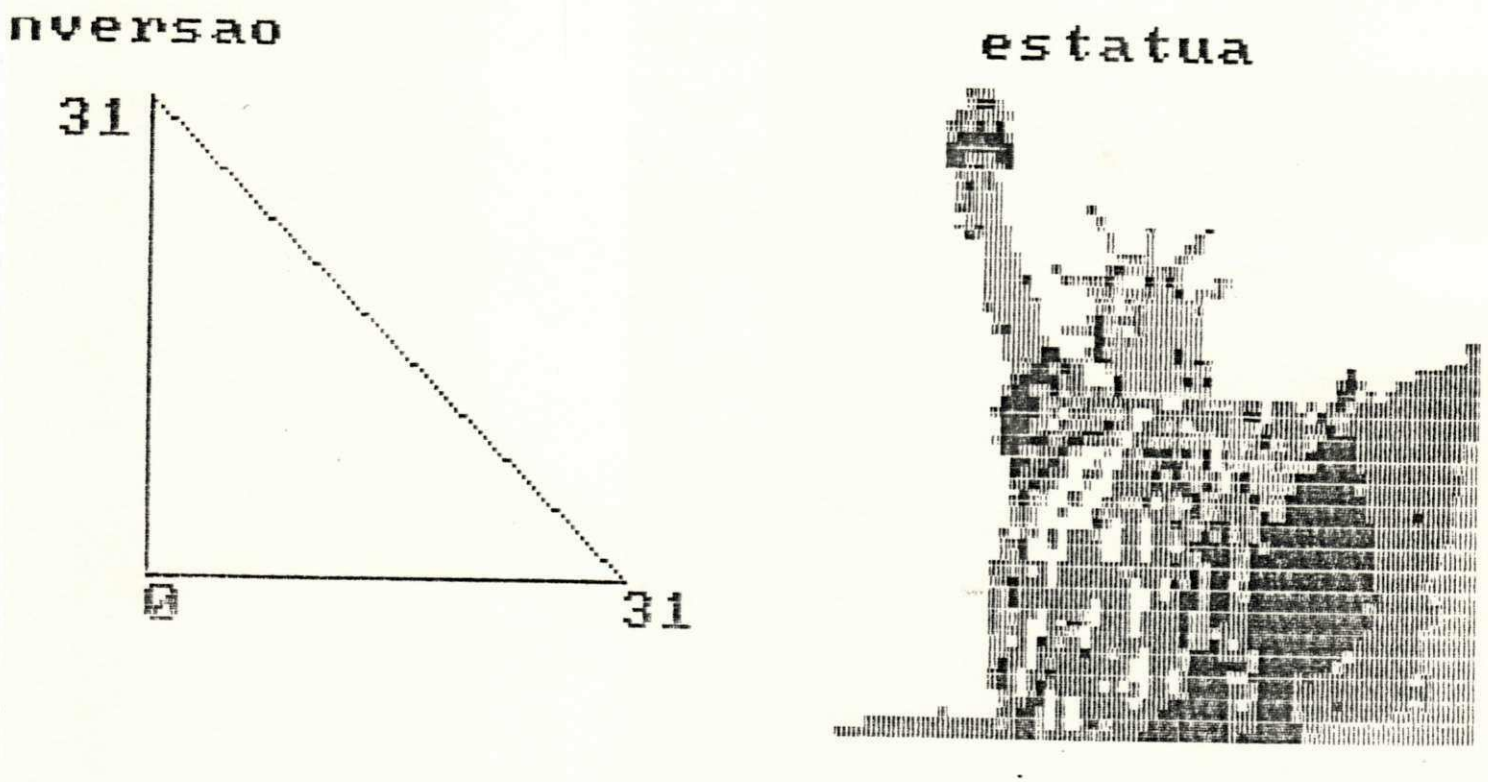


Figura 4.16. Inversão da escala de cinza

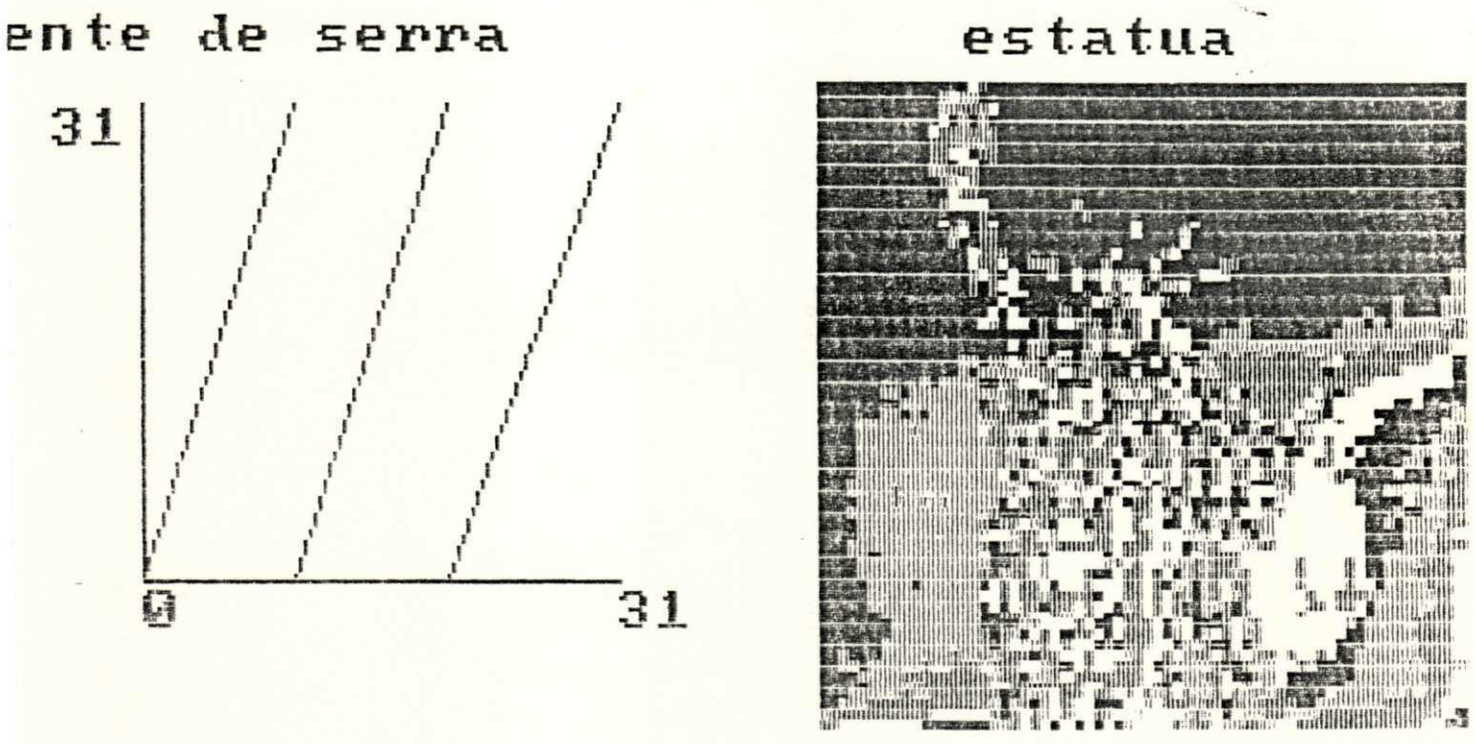


Figura 4.17. Dente de serra 3-ciclos

4.2. Conclusão

Além das funções que representam o movimento e o "display" de imagens no sistema, PICTÓREA contém, nestes módulos, um conjunto de operações pontuais que são utilizadas na área de realce de imagens. Este campo do processamento digital de imagens é bastante vasto, o que pode resultar num grande interesse, por parte do aluno, na elaboração de vários outros algoritmos referentes ao tema.

5. PROGRAMA SPDI - 2ª PARTE

Continuando a abordagem anterior, apresentamos, neste capítulo, o módulo que trata da filtragem espacial.

O módulo F6 (FILTROS ESPACIAIS), do MENU PRINCIPAL, compreende um conjunto de algoritmos que atuam diretamente sobre os pixels da imagem, resultando numa suavização ou num aumento de contraste da imagem original. Estas transformações são do tipo local, ou seja, o valor do nível de cinza de um ponto p, após a transformação, depende do valor do nível de cinza do ponto original e de outros pontos da sua vizinhança.

Pode-se distinguir duas maneiras de se processar uma imagem. Uma delas consiste de uma transformação em que a imagem resultante difere, de algum modo, da imagem original. A outra maneira envolve um resultado que não é uma imagem mas, sim, uma parametrização ou classificação desta [4]. As técnicas de realce, vistas aqui, enquadram-se na primeira forma e são utilizadas no processamento de imagens para diminuir o efeito do ruído, salientar contornos de objetos e destacar outras propriedades específicas.

5.1. Módulo: Filtros espaciais

As seguintes funções fazem parte do módulo FILTROS ESPACIAIS:

FILTROS ESPACIAIS

F1 Convolução

F2 Deteção de bordas

F3 Suavização

F4 Display de imagens

F9 Auxílio

F10 Retorna

Um sensor ou um canal de transmissão pode ser considerado fonte de ruído de uma imagem. Os pixels com erro aparecem freqüentemente diferenciados dos seus vizinhos e são espacialmente descorrelacionados. Devido a isto, o ruído numa imagem geralmente tem um espectro de freqüência espacial maior que os componentes normais da imagem [10]. A aplicação de um simples filtro passa-baixas pode ser eficiente na remoção deste ruído.

No domínio espacial, a suavização é conseguida através da convolução da imagem com uma matriz formada pelos coeficientes do filtro de suavização, conhecida como "máscara". O nível de cinza, localizado no centro da máscara, é substituído por um novo valor que é determinado em função dos níveis de cinza dos pixels contidos nesta máscara.

A convolução discreta entre um vetor de imagem I, de AxB elementos, e uma máscara de convolução H, formada por CxD elementos, é dada através da seguinte relação [7]:

$$I(x,y)H(x,y) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} I(m,n)H(x-m, y-n) \qquad \text{Eq. 5.1}$$

para x = 0, 1, 2, ..., M-1 e y=0, 1, 2, ..., N-1. A matriz MxN, resultante, representa um período da convolução discreta. M e N são escolhidos de tal forma que não haja interferência de períodos adjacentes na convolução, ou seja:

$M \geq A + C - 1$ e

$N \geq B + D - 1$

A função F1, do menu FILTROS ESPACIAIS, possibilita a realização de

filtragens espaciais através de máscaras de convolução. O usuário pode escolher uma máscara de dimensão 3x3 ou 5x5 e, logo após, indicar os coeficientes de peso ou valores que definem esta máscara. Estes coeficientes tanto podem ser iguais, como é o caso do filtro da média, quanto podem decrescer da direção do pixel central à extremidade da máscara.

Para a suavização do ruído, a matriz H (Eq. 5.1) deve representar um filtro passa-baixas em que todos os coeficientes são positivos [10]. Algumas máscaras são ilustradas a seguir.

MÁSCARA 1: $H=1/9 \quad \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$

MÁSCARA 2: $H=1/10 \quad \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{bmatrix}$

MÁSCARA 3: $H=1/16 \quad \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$

Estas máscaras são normalizadas a fim de evitar que o processo de suavização introduza uma polarização de intensidade na imagem filtrada.

Como o filtro passa-baixas não considera mudanças no conteúdo da imagem, enquanto realiza o processo de suavização, ele apresenta o efeito indesejável de nublar os contornos dos objetos.

A Fig. 5.1 apresenta uma imagem filtrada utilizando uma máscara 3x3 de um filtro passa-baixas (MÁSCARA 2).

O realce ou aguçamento de bordas de uma imagem também pode ser feito

através da convolução discreta. Neste caso, o vetor H representa um filtro passa-altas. Algumas máscaras para este tipo de filtragem são dadas abaixo:

MÁSCARA 1: $H = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$

MÁSCARA 2: $H = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{bmatrix}$

MÁSCARA 3: $H = \begin{bmatrix} 1 & -2 & 1 \\ -2 & 5 & -2 \\ 1 & -2 & 1 \end{bmatrix}$

Os filtros passa-altas podem enfatizar o ruído que porventura venha estar presente na imagem, sendo este um dos efeitos indesejáveis deste tipo de processamento.

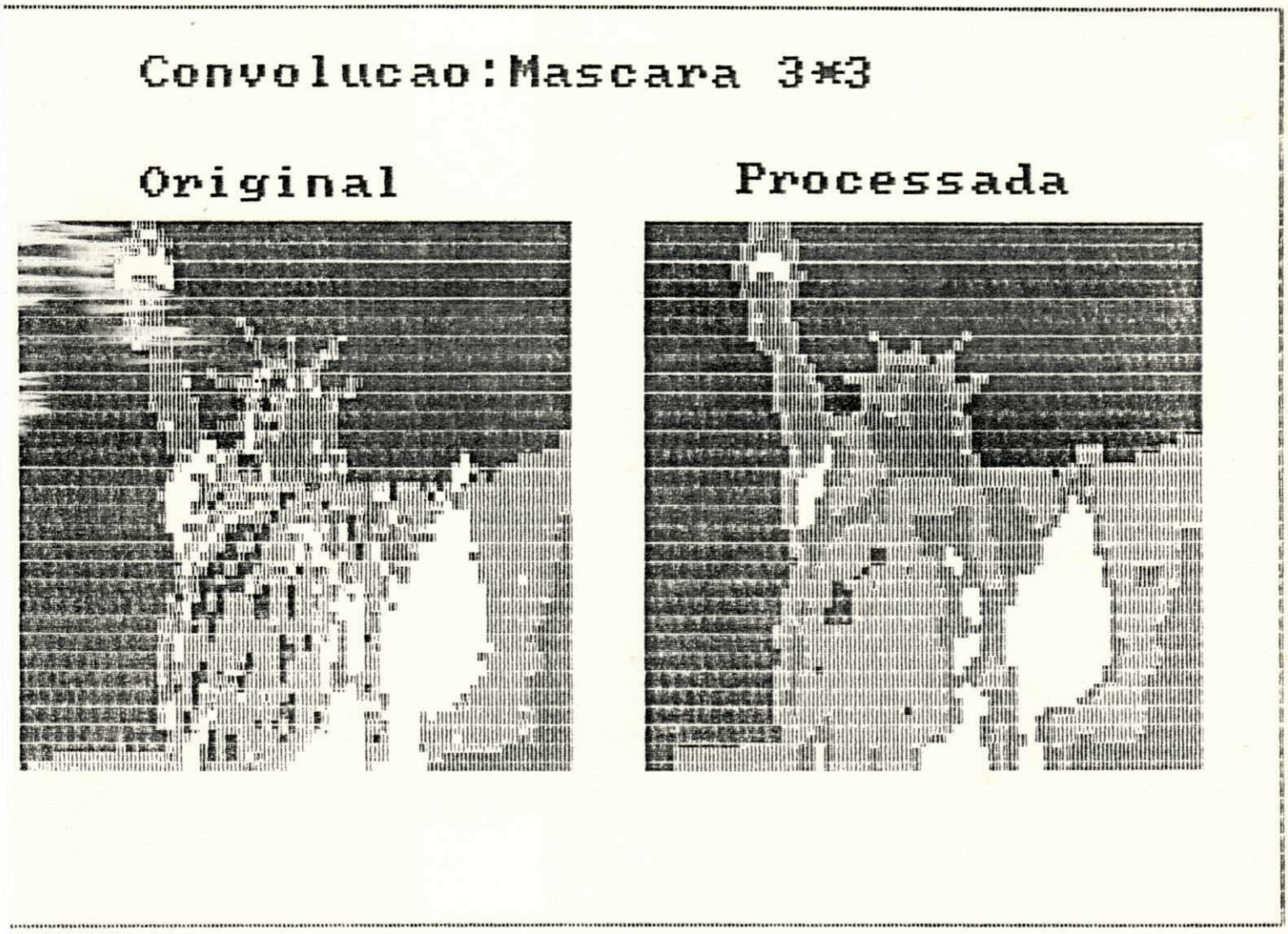


Figura 5.1. Convolução da imagem com uma máscara de um filtro passa-baixas

A Fig. 5.2 apresenta uma imagem filtrada utilizando uma máscara 3x3 de um filtro passa-altas (MÁSCARA 1).

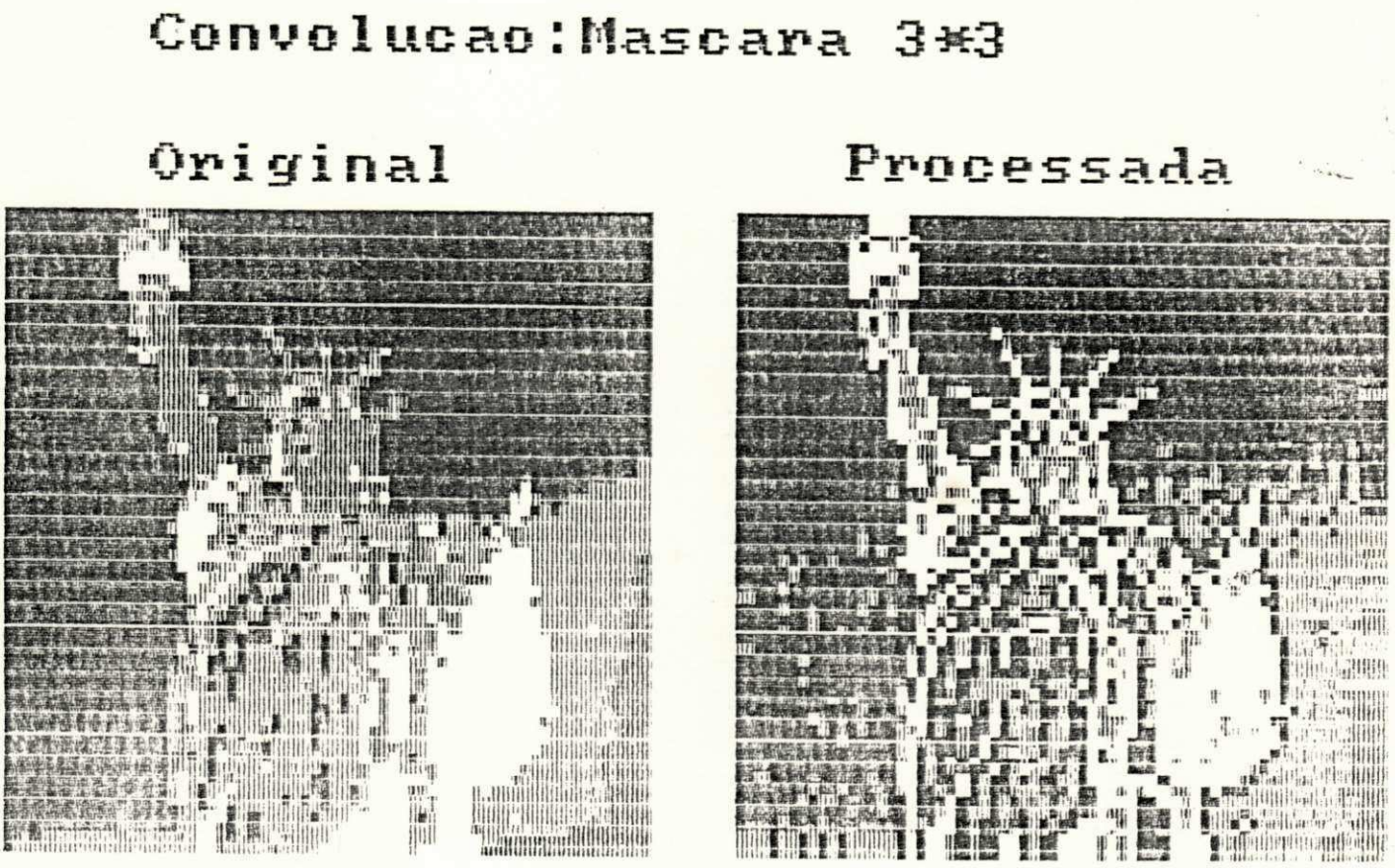


Figura 5.2. Convolução da imagem com uma máscara de um filtro passa-altas

5.1.1. Submódulo 1:/filtros espaciais/deteção de bordas

F2, em FILTROS ESPACIAIS, apresenta uma série de detetores de bordas que servem para identificar descontinuidades locais na luminosidade da imagem, gerando, assim, uma imagem do tipo gradiente. O menu DETEÇÃO DE BORDAS contém as seguintes funções:

DETEÇÃO DE BORDAS

F1 Gradiente de Roberts [7]

F2 Operador de Sobel [11]

F3 Operador de Prewitt [12]

F4 Operadores direcionais

F9 Retorna

OUTRA TECLA Menu principal

Segundo Pratt [10], uma borda pode ser definida como sendo uma mudança ou descontinuidade local na luminosidade de uma imagem.

A integração representa a técnica de suavização de uma imagem no domínio espacial, e a diferenciação, o realce de bordas desta imagem. Entre as técnicas de diferenciação, a mais utilizada é o gradiente [7]. Para uma imagem f(x,y), uma aproximação típica do gradiente é dada pela relação:

$$g(x,y) = G[f(x,y)] = \{[f(x,y)-f(x+1,\ y+1)]^2 + [f(x+1,y) - f(x,y+1)]\}^{1/2} \qquad \text{Eq. 5.2}$$

Um resultado semelhante e mais aconselhável à implementação computacional é obtido usando valores absolutos:

$$g(x,y) = G[f(x,y)] \cong |f(x,y)-f(x+1, y+1)| + |f(x+1,y)-f(x,y+1)| \qquad \text{Eq. 5.3}$$

Este método representa uma operação de diferenças cruzadas introduzidas por Roberts e é comumente chamado de operador de Roberts. Este operador pode ser visto como a combinação das duas máscaras 2x2 indicadas abaixo:

MÁSCARA 1: $\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$ MÁSCARA 2: $\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$

5.1.1.1. Submódulo II:/filtros espaciais/deteção de bordas/gradiente de Roberts

Existem inúmeras formas de se obter uma imagem do tipo G [f(x,y)]. A função F1, do menu DETEÇÃO DE BORDAS, apresenta algumas dessas alternativas:

GRADIENTE DE ROBERTS

F1 Aplicação direta do gradiente

F2 Gradiente com fundo definido

F3 Imagem gradiente binária

F9 Retorna

OUTRA TECLA Menu principal

A mais simples dessas alternativas, F1, consiste do cálculo do gradiente de f(x,y), ou seja:

$$g(x,y) = G[f(x,y)] \qquad \text{Eq. 5.4}$$

Uma desvantagem deste método é que regiões uniformes em f(x,y) apare

cem escuras em g(x,y), devido aos pequenos valores relativos do gradiente nestas regiões (Fig. 5.3).

A função F2 utiliza a seguinte relação para o cálculo do gradiente de Roberts:

$$g(x,y) = \begin{cases} G[f(x,y)] & \text{se } G[f(x,y)] > L \\ N_F & \text{em outro caso} \end{cases} \qquad \text{Eq. 5.5}$$

L representa um limiar não negativo e N_F é o nível de cinza atribuído ao fundo da imagem.

Este método é empregado quando se deseja fazer um estudo da variação dos pixels das bordas sem que haja uma interferência dos níveis de cinza do fundo da imagem (Fig. 5.4).

F3 fornece uma imagem gradiente, utilizando a seguinte relação:

$$g(x,y) = \begin{cases} N_B & \text{se } G[f(x,y)] > L \\ N_F & \text{em outro caso} \end{cases} \qquad \text{Eq. 5.6}$$

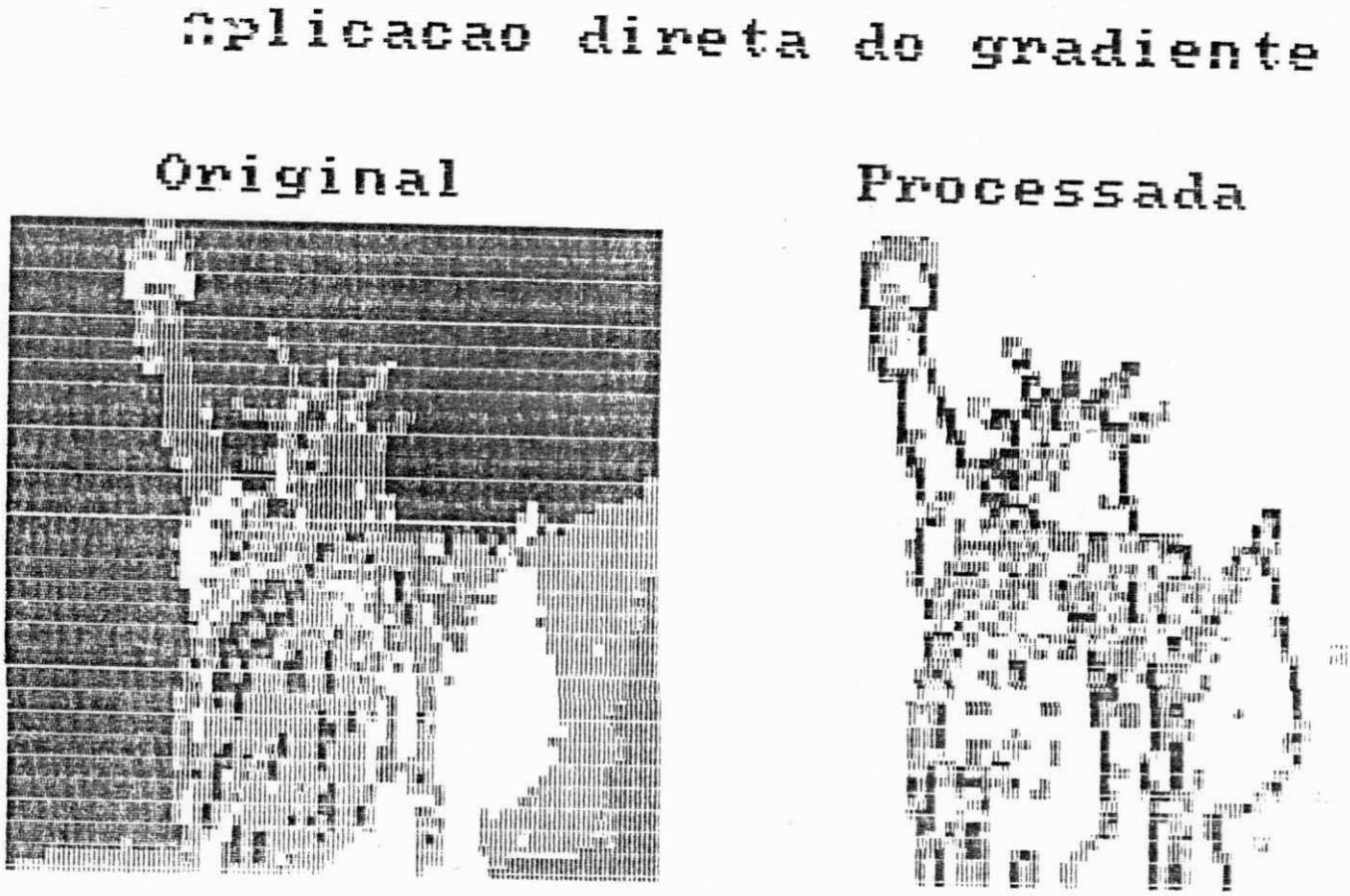


Figura 5.3. Aplicação direta do gradiente de Roberts

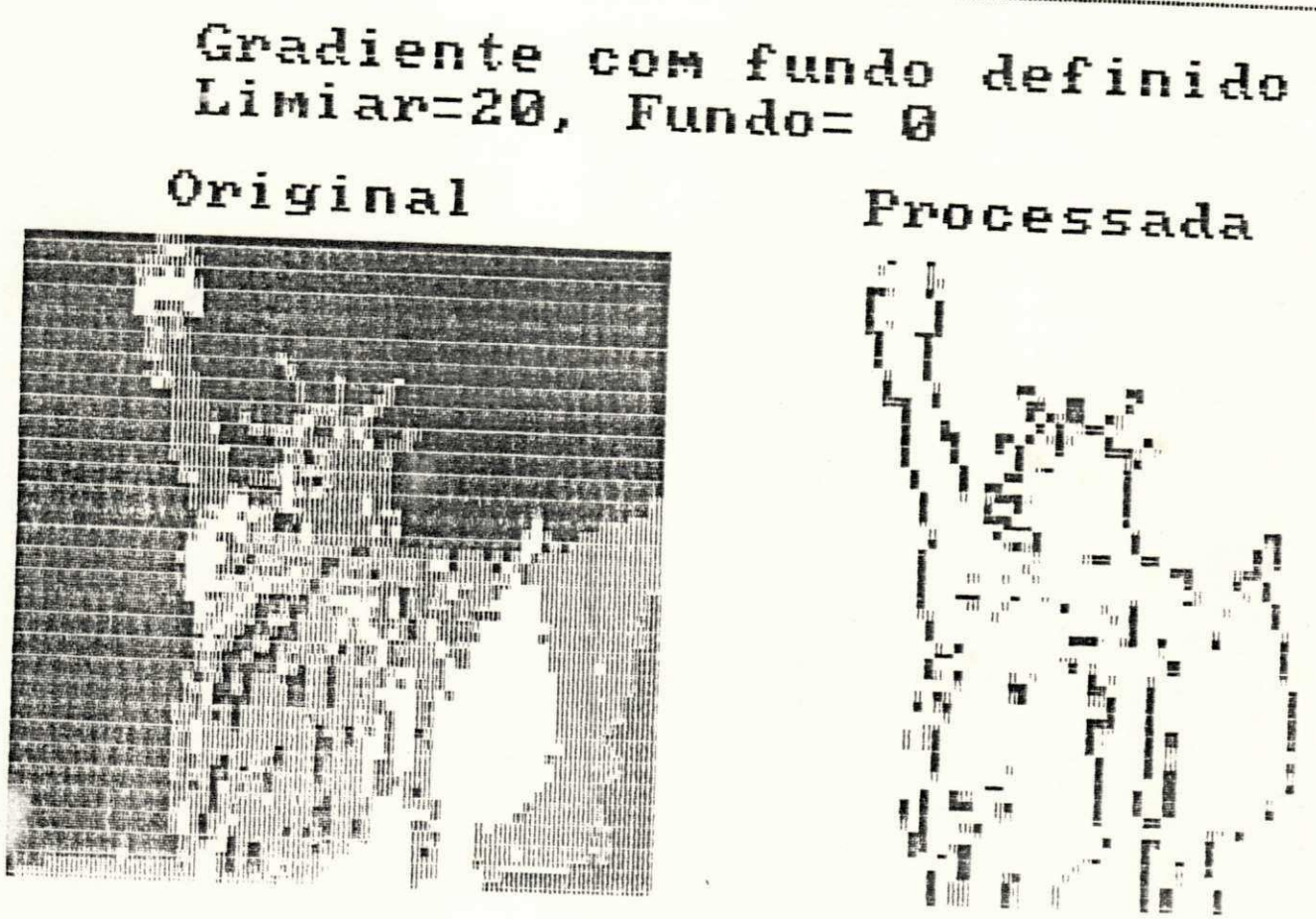


Figura 5.4. Gradiente com fundo definido

N_B é o nível atribuído às bordas da imagem. Neste caso, temos uma imagem binária onde N_F e N_B representam os valores do fundo e das bordas respectivamente (Fig. 5.5). Este método é utilizado quando se tem interesse apenas na localização das bordas da imagem.

Sobel e Prewitt sugeriram operadores 3x3 que podem ser entendidos como duas máscaras de gradiente, uma apontando para a direção norte, e a outra, para a direção leste. Estas máscaras são:

Máscaras de SOBEL		Máscaras de PREWITT	
M_{Norte}	$\begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$	M_{Norte}	$\begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$
M_{Leste}	$\begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}$	M_{Leste}	$\begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix}$

O gradiente da imagem é obtido calculando-se a magnitude da saída das duas máscaras ortogonais. A direção da borda é obtida através de uma operação tangente inversa das saídas das duas máscaras.

As funções F2 e F3 do menu DETEÇÃO DE BORDAS geram imagens do tipo gradiente, utilizando as máscaras mencionadas acima.

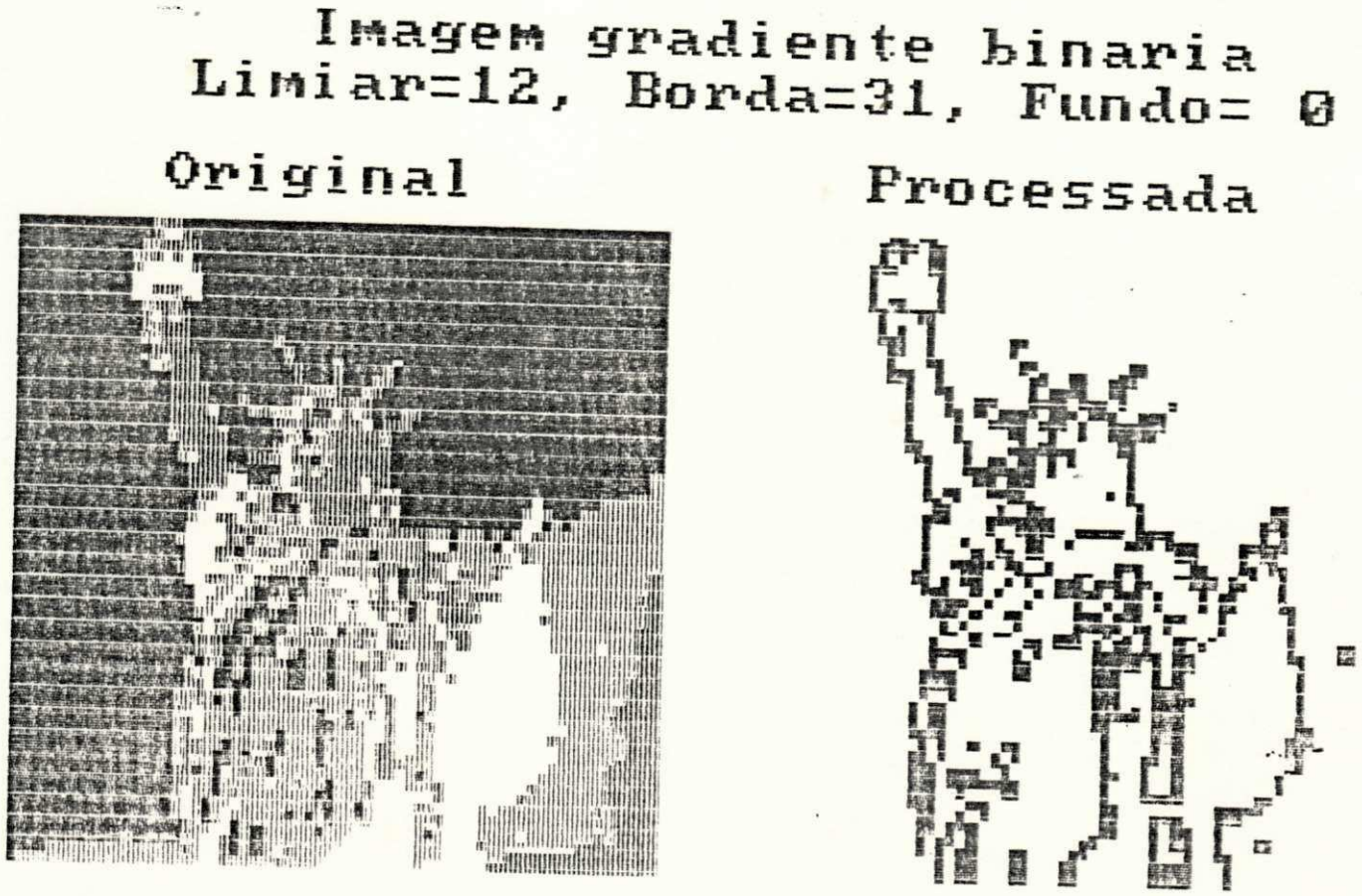


Figura 5.5. Imagem gradiente binária

5.1.1.2 Submódulo II:/filtros espaciais/deteção de bordas/operadores direcionais

Os operadores direcionais em F4 representam um conjunto de máscaras com aproximações discretas de bordas ideais em várias direções. Ao acessar esta função, o usuário dispõe dos seguintes operadores:

OPERADORES DIRECIONAIS

F1 Máscaras direcionais de Prewitt [12]

F2 Máscaras direcionais de Kirsch [13]

F3 Máscaras simples de 3 níveis [14]

F4 Máscaras simples de 5 níveis [14]

F9 Retorna

OUTRA TECLA Menu principal

Apresentamos, a seguir, o conjunto destas máscaras e suas respectivas direções cardeais (Fig. 5.6). A Fig. 5.7 indica a direção cardeal da resposta máxima. As máscaras na direção Norte, por exemplo, produzem uma resposta máxima para mudanças verticais na luminosidade, ou seja, para bordas horizontais. Os números de 0 a 7 indicam as direções das bordas correspondentes às direções cardeais.

Direc. borda	Direc. grad.	Másc. Prewitt	Másc. Kirsch	Másc. 3-Níveis	Másc. 5-Níveis
		1 1 1	5 5 5	1 1 1	1 2 1
0	Norte	1 -2 1	-3 0 -3	0 0 0	0 0 0
		-1 -1 -1	-3 -3 -3	-1 -1 -1	-1 -2 -1
		1 1 1	5 5 -3	1 1 0	2 1 0
1	Noroeste	1 -2 -1	5 0 -3	1 0 -1	1 0 -1
		1 -1 -1	-3 -3 -3	0 -1 -1	0 -1 -2
		1 1 -1	5 -3 -3	1 0 -1	1 0 -1
2	Oeste	1 -2 -1	5 0 -3	1 0 -1	2 0 -2
		1 1 -1	5 -3 -3	1 0 -1	1 0 -1
		1 -1 -1	-3 -3 -3	0 -1 -1	0 -1 -2
3	Sudoeste	1 -2 -1	5 0 -3	1 0 -1	1 0 -1
		1 1 1	5 5 -3	1 1 0	2 1 0
		-1 -1 -1	-3 -3 -3	-1 -1 -1	-1 -2 -1
4	Sul	1 -2 1	-3 0 -3	0 0 0	0 0 0
		1 1 1	5 5 5	1 1 1	1 2 1
		-1 -1 1	-3 -3 -3	-1 -1 0	-2 -1 0
5	Sudeste	-1 -2 1	-3 0 5	-1 0 1	-1 0 1
		1 1 1	-3 5 5	0 1 1	0 1 2
		-1 1 1	-3 -3 5	-1 0 1	-1 0 1
6	Leste	-1 -2 1	-3 0 5	-1 0 1	-2 0 2
		-1 1 1	-3 -3 5	-1 0 1	-1 0 1
		1 1 1	-3 5 5	0 1 1	0 1 2
7	Nordeste	-1 -2 1	-3 0 5	-1 0 1	-1 0 1
		-1 -1 -1	-3 -3 -3	-1 -1 0	-2 -1 0

Figura 5.6. Máscaras direcionais

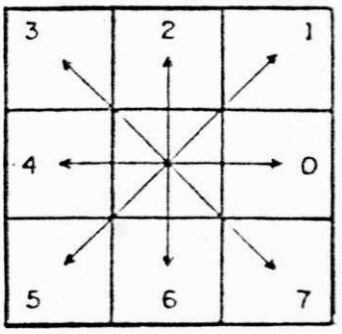


Figura 5.7. Direções das bordas

Como podemos observar, cada grupo da figura 5.6 acima, aplica oito máscaras em cada vizinhança. A magnitude do gradiente é igual à resposta mais forte entre as oito máscaras. A direção é dada pela orientação da máscara com resposta mais forte.

5.1.2. Submódulo 1:/filtros espaciais/suavização

Os seguintes procedimentos de suavização são abordados no módulo FILTROS ESPACIAIS deste trabalho:

SUAVIZAÇÃO

F1 Filtro da média [7]

F2 Filtros da ordem [15]

F3 Suavização com vizinhança selecionada por variância [16]

F4 Suavização com vizinhança selecionada por soma de diferenças absolutas [17, 18]

F5 Filtro da média com os k-vizinhos mais próximos [19]

F6 Filtro sigma [20]

F9 Retorna

OUTRA TECLA Menu principal

Estes procedimentos são aplicados na redução do ruído das imagens, observando a preservação de características importantes, como as bordas, por exemplo. Eles requerem alguns critérios de processamento que podem ser obtidos, ou por uma comparação local da vizinhança ou por um conhecimento a priori que permita a aplicação de determinado tipo de janela, ou parâmetro, a ser utilizado na filtragem.

A seguir descrevemos, separadamente, cada uma das funções deste submódulo.

5.1.2.1. F1 - filtro da média

O filtro da média representa um processo linear de suavização, que utiliza uma máscara onde todos os pesos são iguais a 1. Esta máscara, mencionada no item 5.1, é novamente apresentada aqui:

$$M = 1/9 \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Para uma janela wxw da imagem com níveis de cinza p(i), onde i = 1, 2, ...w^2, a média é igual a:

$$M = \frac{1}{w^2} \sum_{i=1}^{w^2} p(i) \qquad \text{Eq. 5.7}$$

Desta forma, um pixel afetado por um ruído do tipo aditivo, aleatório e descorrelacionado, pode ser substituído pela média M, reduzindo, assim, as variações dos níveis de cinza da imagem.

Este filtro não se mostra muito eficiente, quando se trata de preservação de bordas, mas realiza eficientemente a remoção do ruído na imagem (Fig. 5.8).

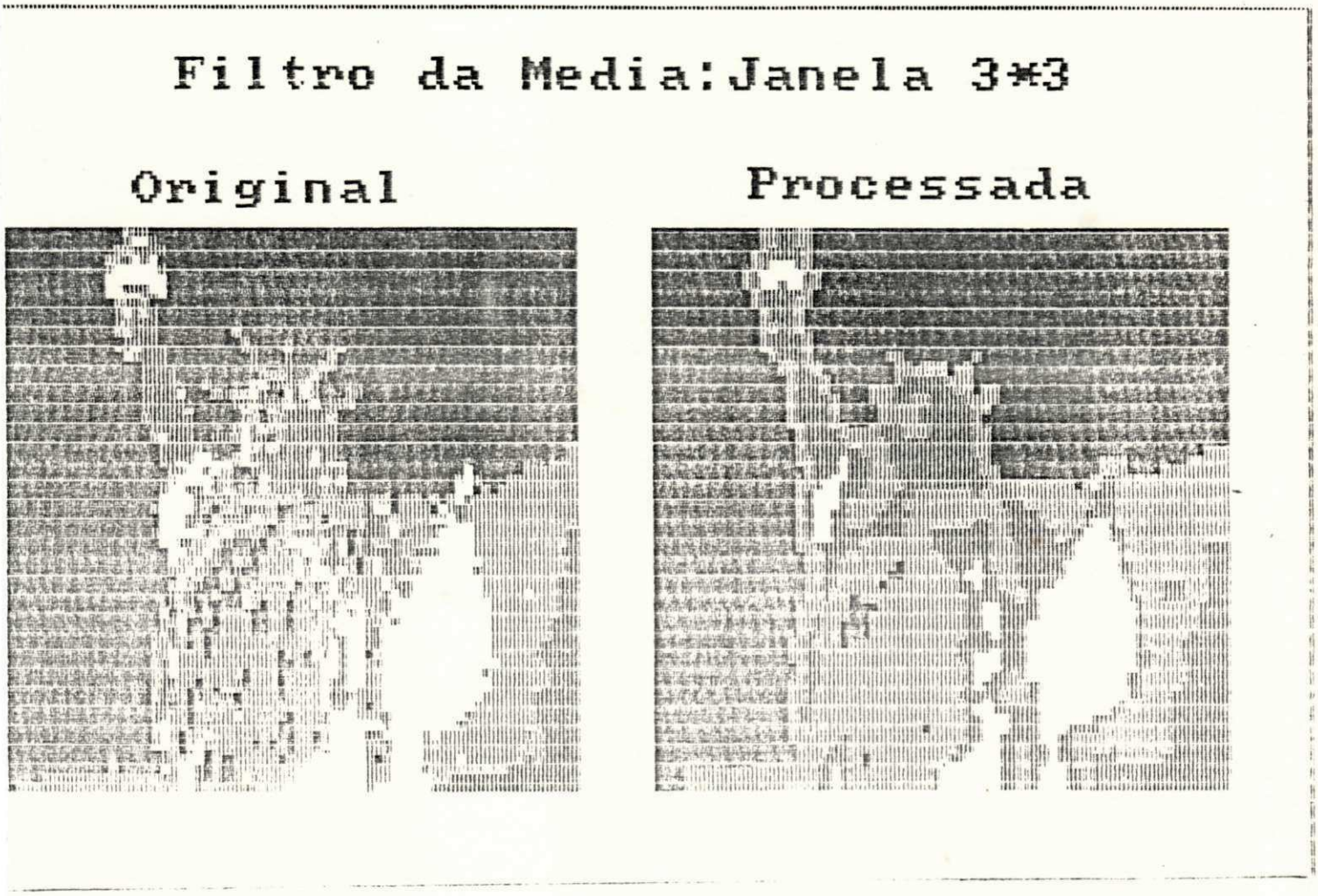


Figura 5.8. Filtro da média

5.1.2.2. F2 - Filtros da ordem

Sejam p(1), p(2), ... p(E) os níveis de cinza dos E elementos contidos numa janela wxw de uma imagem I(m,n), e c(1), c(2), ... c(E) estes mesmos elementos reagrupados em ordem crescente. A relação seguinte define o filtro da ordem $R_{E,K}$ operando na imagem I(m, n):

$$R_{E,K}\ I(m,n) = c(k) \qquad \text{Eq. 5.8}$$

onde **k** representa o k-ésimo valor dos níveis de cinza, quando organizados em ordem crescente, da janela wxw.

Os filtros da ordem com k=1 e k=E, são conhecidos, respectivamente, como os operadores min e max [21]. Para k=(E+1)/2 temos o conhecido filtro da mediana [10], em que cada pixel da imagem é substituído pelo nível de cinza mediano da escala crescente dos níveis de cinza da janela wxw.

A função F2, deste módulo, possibilita a realização do filtro da ordem através de uma janela de dimensão 3x3 ou 5x5.

5.1.2.3. F3 - Suavização com vizinhança selecionada por variância (SVSV)

Em seus estudos sobre técnicas de filtragem, Nagao e Matsuyama [16] subdividiram uma janela 5x5, contendo um ponto central **p**, em nove regiões, como ilustra a Fig. 5.9.

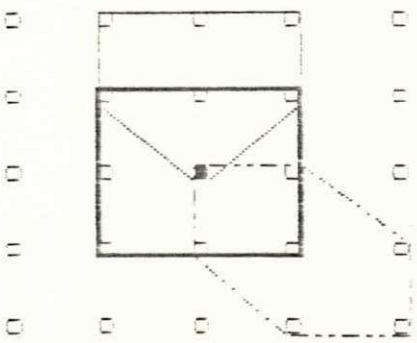


Figura 5.9. Vizinhanças de Nagao e Matsuyama

Estas regiões são formadas por quatro vizinhanças pentagonais, quatro vizinhanças hexagonais, e uma retangular, de dimensão 3x3, centrada no ponto **p**. Este ponto é substituído, durante o processo de filtragem, pelo nível de cinza médio dos elementos contidos na região que apresenta o maior índice de homogeneidade. Este índice é determinado pelo cálculo da variância dos níveis de cinza contidos em cada uma dessas regiões.

A estrutura computacional para este tipo de algoritmo é complexa e o tempo de processamento exigido é significativo; no entanto, o filtro SVSV apresenta-se como um bom filtro, no que se refere à remoção de ruídos com preservação de bordas.

5.1.2.4. F4 - Suavização com vizinhança selecionada por soma de diferenças absolutas (SSDA)

Este algoritmo de suavização, desenvolvido por Araújo [17,18], utiliza nove vizinhanças 3x3, superpostas numa janela de 5x5 pixels, como mostra a Fig. 5.10. O filtro SSDA calcula um índice de homogeneidade para cada uma das nove vizinhanças. Este índice é dado pela soma das diferenças absolutas (SDA) dos níveis de cinza entre o ponto central **p**, da janela, e os demais pontos da vizinhança.

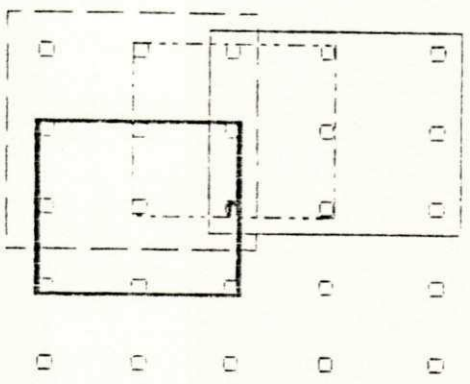


Figura 5.10. Vizinhanças 3x3 superpostas numa janela 5x5

O índice de homogeneidade é dado por:

$$SDA(k) = \sum_{k} [p(i) - p'] \quad k=1, 2, \ldots 9 \qquad \text{Eq. 5.9}$$

onde i=1, 2, ...9, p(i) representa o i-ésimo nível de cinza da k-ésima vizinhança, e p' é o nível de cinza do ponto p.

O algoritmo substitui p' pela média dos níveis de cinza da vizinhança com menor soma das diferenças absolutas dos seus elementos.

Como a vizinhança selecionada está contida inteiramente na região que contém o ponto central, o algoritmo consegue, através da região mais homogênea, reduzir o efeito do ruído, enquanto preserva as bordas da imagem (Fig. 5.11).

Este método de filtragem espacial, bem como o estudado no item anterior, são conhecidos como métodos de suavização por vizinhanças seletivas e constituem uma alternativa aos métodos que utilizam vizinhanças fixas.

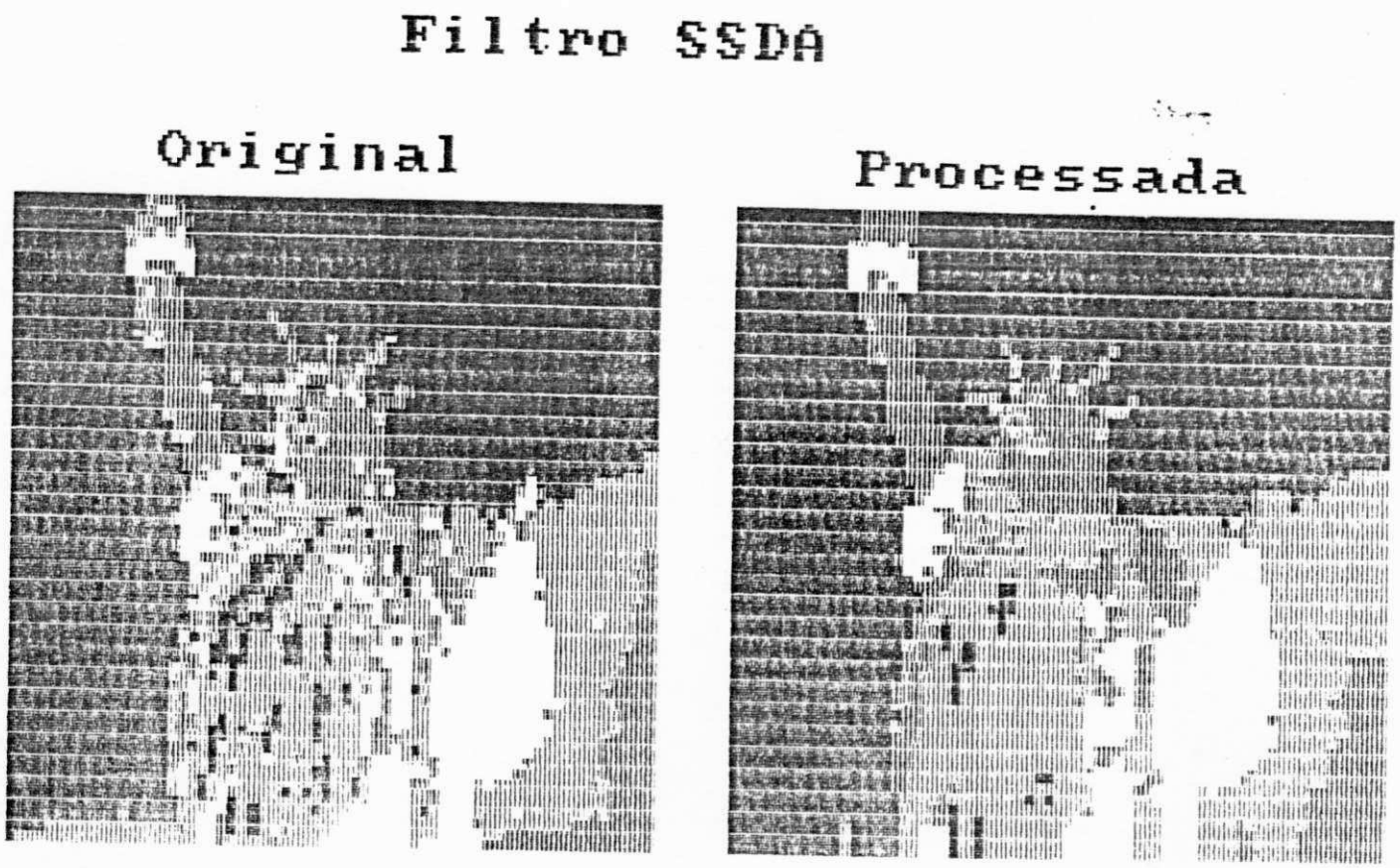


Figura 5.11. Suavização com vizinhança selecionada por soma de diferenças absolutas

5.1.2.5. F5 - Filtro da média com os k-vizinhos mais próximos

Este filtro substitui o ponto central P, de uma janela wxw, pela média dos k-vizinhos cujos níveis de cinza mais se aproximam do valor de **p**.

O filtro da média com os k-vizinhos mais próximos consegue reduzir ruídos e preservar bordas, baseando-se na alta correlação existente entre os níveis de cinza contidos na região wxw da imagem. À medida que o valor de k aumenta, aumenta também a capacidade de remoção do ruído; no entanto, as bordas da imagem se tornam cada vez mais "nubladas".

5.1.2.6. Filtro sigma

O filtro sigma realiza a suavização do ruído na imagem, substituindo o ponto central **p**, de uma janela, pela média dos elementos que têm seus níveis de cinza dentro de uma faixa sigma de intensidade fixada por **p**.

O algoritmo para o filtro sigma pode ser descrito da seguinte forma. Sejam p(i) e p'(i) o nível de cinza do i-ésimo pixel de uma janela wxw, e o i-ésimo pixel suavizado, respectivamente. Considere, ainda, um ruído do tipo aditivo com valor médio zero e desvio-padrão σ. Então, a seguir:

1. Estabeleça uma faixa de intensidade (p(i)-T,p(i)+T) onde $T=2\sigma$
2. Adicione todos os pixels da janela wxw cujos níveis de cinza se encontram dentro da faixa determinada em 1.
3. Calcule a média desses níveis de cinza
4. Faça p'(i) = média.

Como podemos observar, qualquer pixel fora da faixa dos dois sigmas pertence, provavelmente, a outra vizinhança que não wxw, devendo, portanto, ser excluído da suavização. Esta faixa de intensidade pode ser grande

o suficiente para incluir 95,5% dos elementos da mesma distribuição da janela, ou pequena o suficiente para excluir pixels que representam bordas mais complexas.

O filtro sigma realiza satisfatoriamente a remoção de ruídos com preservação de bordas, sendo indicado quando se deseja preservar detalhes e linhas finas de uma imagem.

5.2. Conclusão

Inúmeros algoritmos de suavização e deteção de bordas têm sido apresentados na literatura. PICTÓREA contém alguns exemplos desses trabalhos, ilustrados a partir de técnicas de filtragem espacial do tipo linear, estatística e com vizinhança seletiva.

6. PROGRAMA SPDI - 3ª PARTE

Os dois módulos restantes do MENU PRINCIPAL permitem a geração de algumas imagens definidas por PICTÓREA e a adição de ruído uniforme e gaussiano a uma imagem qualquer do sistema.

6.1. Módulo: Geração de imagens

Este módulo do MENU PRINCIPAL apresenta um conjunto de funções que permitem a criação de imagens sintéticas a serem utilizadas na realização de alguns experimentos de processamento digital de imagens. As mesmas são geradas a partir da definição de certos parâmetros de entrada, tais como: pontos de ligação, cores, dimensões etc.

As seguintes funções estão presentes neste módulo:

GERAÇÃO DE IMAGENS

F1 Círculo

F2 Retângulo

F3 Quatro quadrados

F4 Tabuleiro

F5 Display de imagens

F9 Auxílio

F10 Retorna

A Fig. 6.1 apresenta uma imagem tabuleiro gerada a partir da função F4 deste menu.

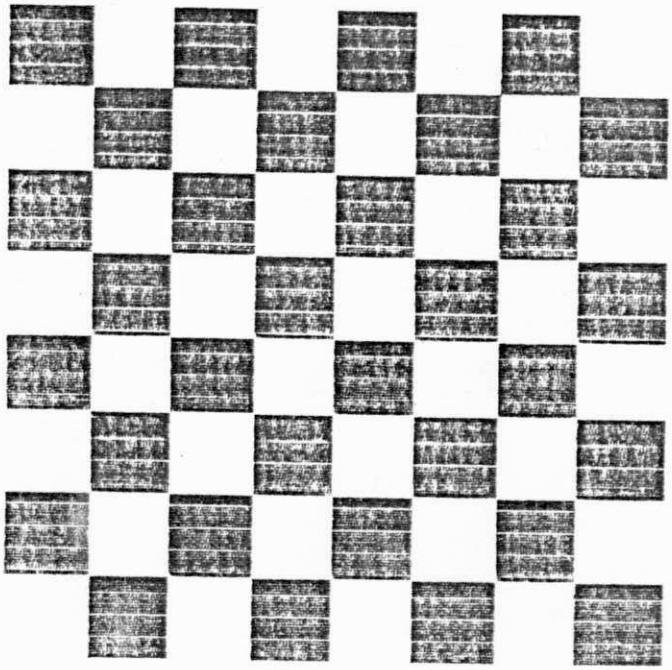


Figura 6.1. Imagem tabuleiro

6.2. Módulo: Geração de ruídos

O módulo GERAÇÃO DE RUÍDOS permite a adição de ruídos não correlacionados às imagens do sistema, através das distribuições gaussiana e uniforme.

Este módulo compreende as seguintes funções:

GERAÇÃO DE RUÍDOS

F1 Ruído com distribuição uniforme

F2 Ruído com distribuição gaussiana

F3 Display de imagens

F9 Auxílio

F10 Retorna

O usuário deve indicar os valores da média e do desvio-padrão para cada uma das referidas distribuições.

A Fig. 6.2 mostra uma imagem com ruído do tipo gaussiano (média=1,6, desvio-padrão = 7,0).

6.3. Conclusão

O conjunto de funções do módulo GERAÇÃO DE IMAGENS pode servir para ilustrar facilmente algumas técnicas simples empregadas na análise de imagens digitalizadas (histograma, perfil de varredura etc.). O módulo GERAÇÃO DE RUÍDOS permite o estudo da influência do ruído, gaussiano ou uniforme, numa determinada imagem. A partir da imagem ruidosa, o usuário pode fazer, por exemplo, uma comparação entre os resultados dos diversos filtros de suavização espacial presentes no sistema.

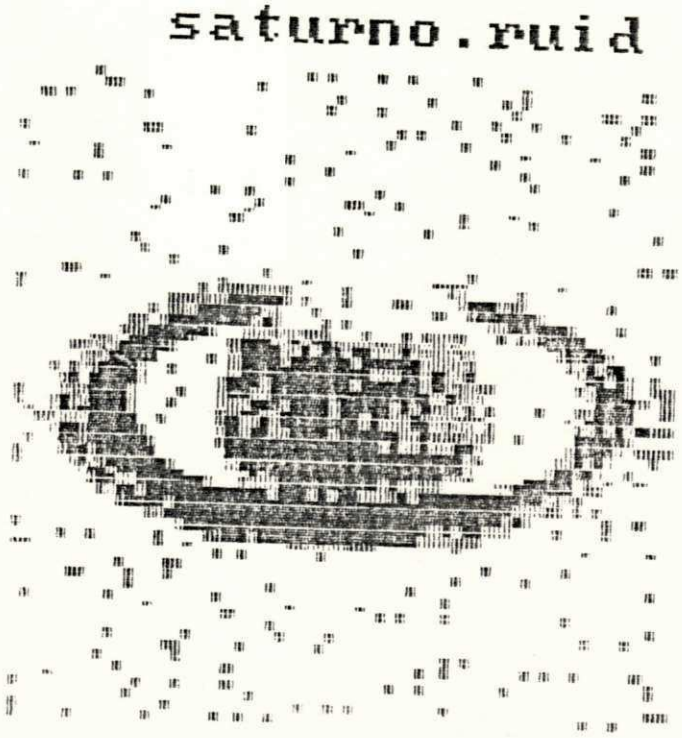


Figura 6.2. Imagem com ruído

CONCLUSÃO

PICTÓREA foi projetado para possibilitar um desempenho melhor, tanto do aluno quanto do orientador, no ensino de uma disciplina de processamento digital de imagens.

Apresentando uma interface sistema-usuário bastante simples, o que permite uma interação rápida no que diz respeito aos métodos de acesso aos diversos módulos do programa, PICTÓREA oferece um conjunto de algoritmos que representam muito bem as aplicações comumente utilizadas no tratamento de imagens.

De modo geral, PICTÓREA visa acompanhar, experimentalmente, os conceitos teóricos sobre PDI fornecidos em sala-de-aula. O orientador deve, dentro das suas necessidades, traçar um planejamento detalhado no que diz respeito à utilização do sistema. Ao aluno cabe, basicamente, verificar os resultados práticos, tirar suas conclusões e, numa etapa seguinte, criar seus próprios algoritmos, expandindo, assim, as potencialidades do sistema.

Ilustramos, a seguir, uma possível utilização do programa numa aula sobre filtros espaciais. Esta aula pode desenvolver-se nas seguintes etapas:

1. O orientador introduz o conceito de filtragem espacial acompanhado, por exemplo, da apresentação de alguns algoritmos de realce contidos no sistema.

2. A execução dos algoritmos deve ser visualizada para que o aluno tire suas conclusões no que concerne ao tempo de execução, capacidade de remoção de ruído, preservação de bordas etc. As informações quantitativas

e qualitativas do processamento podem ser obtidas através dos valores do EMQ e da MDA entre uma imagem original e uma imagem processada, e através do histograma das duas imagens, por exemplo. Em suma, todas as informações que sirvam para caracterizar uma imagem devem ser usadas intensivamente na análise dos resultados.

3. Na etapa final, o orientador sugere alguns algoritmos de filtragem, não disponíveis no sistema, e pede que o aluno os implemente, verifique e forneça resultados, tais como impressão da imagem processada, impressão dos níveis de cinza desta imagem etc.

SUGESTÕES

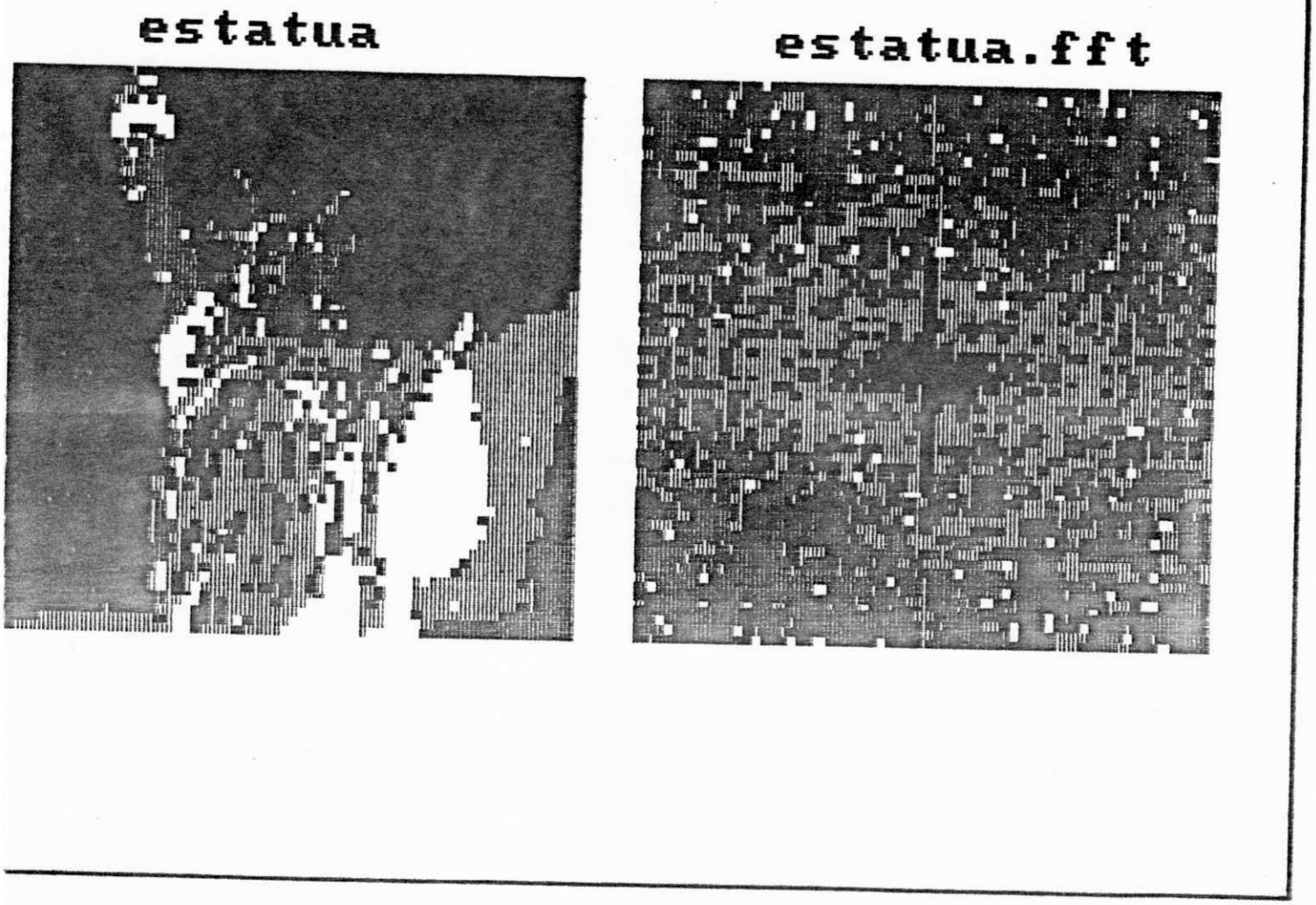
. Através da utilização do pacote gráfico, GDT, utilizado no sistema, PICTÓREA pode ter a sua estrutura ampliada, onde novos módulos e funções podem ser implementados.

. A introdução no sistema de linguagens do tipo Turbo C ou Turbo Pascal, por exemplo, pode oferecer melhor resultado, no que se refere à apresentação dos diversos módulos de PICTÓREA. O emprego de menus com janelas é uma das facilidades propostas por estas linguagens e que pode tornar mais dinâmica a interação sistema-usuário.

. O usuário pode pensar em expandir o seu sistema através da implementação de outros algoritmos que trabalhem no domínio da frequência, utilizando, por exemplo, as transformadas rápidas de Fourier ou de Walsh fornecidas juntamente com PICTÓREA. (Figura seguinte)

. Dispondo de memórias de imagens maiores que 64x64 e de unidade de visualização de maior capacidade, o usuário pode ampliar consideravelmente as características de PICTÓREA empregando-o, assim, em atividades ligadas

à pesquisa na área de PDI. Para isto, basta que sejam feitas algumas modificações nas subrotinas referentes à manipulação das imagens no sistema, como, por exemplo, as funções de leitura e escrita, funções de display de imagens etc.

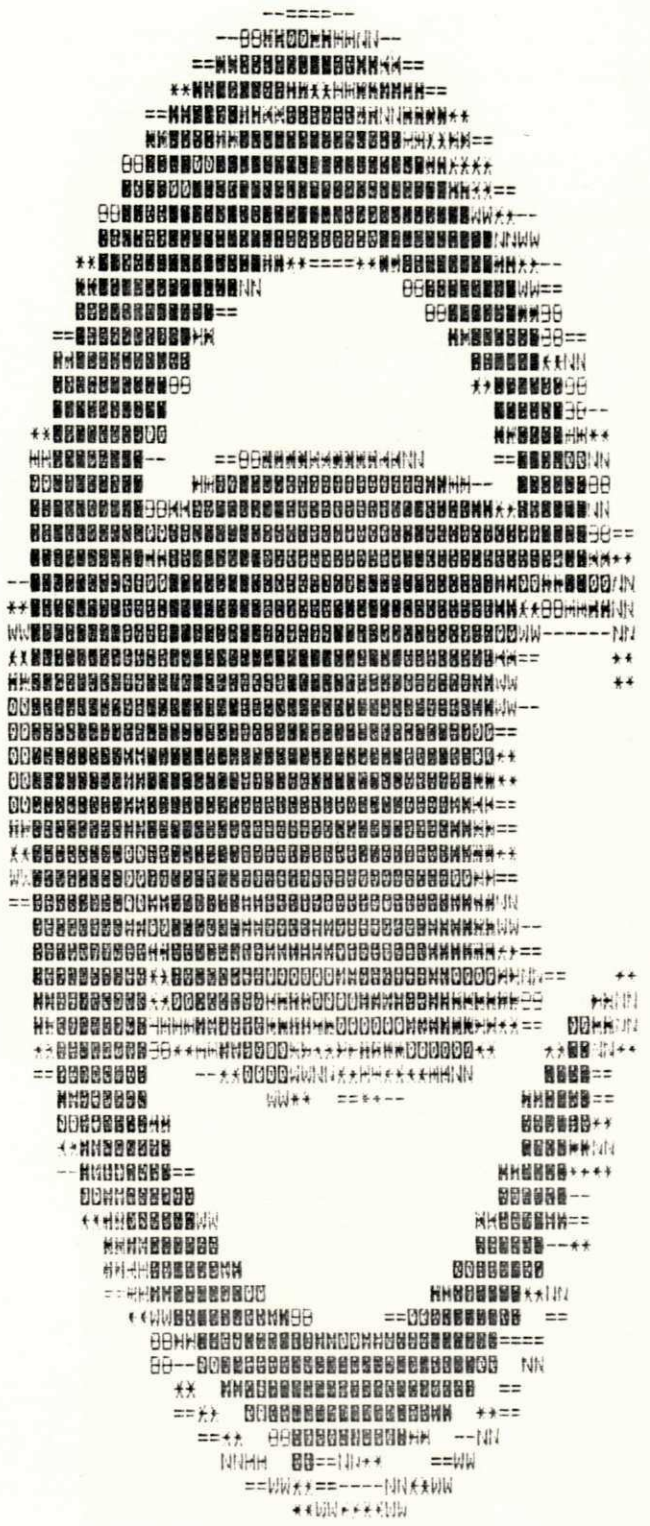


Transformada Rápida de Fourier de uma imagem

APÊNDICE 1

ESCALA LINEAR

ESCALA RAIZ QUADRÁTICA



ESCALA LOGARÍTMICA

ESCALA ABSORÇÃO

APÊNDICE 2

```
{introducao pictorea: sistema portatil para tratamento de imagens}
{Apresenta MENU DE ABERTURA}
{$M 8000,0,16000}
{$R-}
Uses Crt,Dos,Turbo3;
Type Str80  = String[80];
     Str8   = String[8];
     Str255 = String[255];
Var VideoMem : Array [1..25,1..80,1..2] of Byte absolute $B800:0000;
    Drive  :  String[80];
    Tecla  :  Char;
    CodigoErro,
    Posicao,
    Intl   :  Integer;
    OKey   :  Boolean;

Function Cadeia (Numero : Word) : Str8;
{retorna uma string do numero enviado}
Var CadAux: Str8;
Begin
  Str (Numero:2,CadAux);
  If CadAux[1] = ' '
    Then
      CadAux[1]:='0';
  Cadeia:=CadAux;
End;

Function PegueHora : Str8;
{retorna uma string com a hora atual}
Var Hora,Minuto,Segundo,Centesimo : Word;
    CadAux : Str8;
Begin
  CadAux:='';
  GetTime (Hora,Minuto,Segundo,Centesimo);
  CadAux:=Cadeia(Hora)+':'+Cadeia(Minuto)+':'+Cadeia(Segundo);
  PegueHora:=CadAux;
End;

Function PegueData : Str8;
{retorna uma string com a data atual}
Var Ano,Mes,Dia,DiaSemana : Word;
    CadAux : Str8;
Begin
  CadAux:='';
  GetDate(Ano,Mes,Dia,DiaSemana);
  CadAux:=Cadeia(Dia)+'/'+Cadeia(Mes)+'/'+Copy (Cadeia(Ano),3,2);
  PegueData:=CadAux;
End;

Procedure ApagaCentro;
{apaga o centro da tela}
  Begin
    Window(14,12,67,16);
    ClrScr;
    Window(1,1,80,25);
  End;

Procedure LerSeta (Var Tecla: Char);
{esta procedure ler as teclas de seta, ESC, ENTER}
```

```
Var EspKey: Boolean;
Begin
  Repeat
    Tecla:=ReadKey;
    If Keypressed and (Tecla=#0)
      Then
        Begin
          Tecla:=ReadKey;
          EspKey:=TRUE;
        End
      Else
        EspKey:=FALSE;
  Until ((Tecla in [#27,#13]) and not EspKey) or ((Tecla in [#71..#73,#79..#81,
          #115..#119,#75,#77,#160,#132,#164]) and EspKey);
End;

Procedure InvertCampo (PosX,PosY,Tamanho : Integer);
{ inverte (desinverte) o campo especificado }
  Var Int1  :  Integer;
  Begin
    For Int1 := PosX to Tamanho+PosX-1 do
      If VideoMem [PosY,Int1,2] = 46
        Then VideoMem [PosY,Int1,2]:=14
        Else VideoMem [PosY,Int1,2]:=46;
  End;

Procedure Cursor(Tipo:Char);
{ tipos:  A: apaga      I: insere      N: normal}
  Var Registros : Registers;
      Ch,Cl     : Byte;

  Begin
    With Registros do
      Begin
        Ax:=$0100;
        Case Tipo of
          'A'  :  Begin  Ch := 8;  Cl := 7;  End;
          'I'  :  Begin  Ch := 4;  Cl := 7;  End;
          'N'  :  Begin  Ch := 6;  Cl := 7;  End;
        End;
        Cx := 256*Ch + Cl;
        Intr ($10,Dos.Registers(Registros));
      End;
  End;

Procedure Erro(Programa:Str80);
Var Tecla : Char;
Begin
  Cursor ('A');
  ApagaCentro;
  Window (14,12,67,16);
  Gotoxy (1,1);
  Writeln ('ERRO ao tentar acessar o programa:');
  Writeln (') ',Programa);
  Gotoxy (1,4);
  Writeln ('Pressione <ENTER> ou <ESC>');
  Repeat
    LerSeta(Tecla);
  Until Tecla in [#13,#27];
```

```
  ApagaCentro;
End;

Procedure Diretorio;
{este procedure mostra o diretorio de path especificado}
 Var Arquivo  :  Array [1..250] of String[8];
     Informacao: SearchRec;            {contem informacoes sobre o arquivo}
     Tecla    :  Char;                 {tecla pressionada                  }
     NumArq,                           {numero de arquivos encontrados     }
     Cont,Int1,InicAux,                {variaveis auxiliares               }
     Posicao,
     Inicio   :  Integer;              {ordem do primeiro arquivo mostrado}

Procedure MostraPagina;
{esta procedure mostra os nomes dos arquivos}
  Begin
    ClrScr;
    Cont:=1;
    While (Cont <=  20) and (Cont+Inicio-1 <= NumArq) do
      Begin
        Gotoxy ((Cont-trunc((Cont-1)/5)*5)*10-9,Trunc((Cont-1)/5)+1);
        Int1:=Pos('.',Arquivo[Cont+Inicio-1]);
        If Int1 = 0
          Then
            Int1:=9;
        Write (Copy(Arquivo[Cont+Inicio-1],1,Int1-1));
        Cont:=Cont+1;
      End;
  End;

Begin
  Window (14,12,67,16);
  ClrScr;
  Drive:='';
  Gotoxy (1,1);
  Writeln ('Indique o driver a ser pesquisado:');
  Write ('> ');
  Readln (Drive);
  For Int1:=1 to Length (Drive) do
    Drive[Int1]:=UpCase(Drive[Int1]);
  ClrScr;
  Cursor ('A');
  FindFirst (Drive+'*.img',Archive,Informacao);    {encontra arquivos}
  Cont:=1;
  While (DosError = 0) and (Cont < 250) do
   Begin
     Arquivo[Cont]:=Informacao.Name;
     Inc(Cont);
     FindNext(Informacao);
   End;
  NumArq:=Cont-1;
  InicAux:=0;
  Inicio:=1;            Posicao:=0;
  Gotoxy (1,0);
  Write ('Diretorio: '+Drive+'*.IMG:');
  Window (14,13,67,16);
  If NumArq = 0
    Then
```

```
      Begin
        Gotoxy (1,2);
        Writeln ('Nenhum arquivo encontrado.');
        Writeln ('Pressione <ENTER> ou <ESC>');
        Repeat
          LerSeta(Tecla);
        Until Tecla in [#27,#13];
      End
    Else
      Repeat
        If InicAux <> Inicio Then MostraPagina;
        LerSeta (Tecla);
        InicAux:=Inicio;
        Case Tecla of
{dw,pd}  #80,#81
              :  If Inicio+Posicao+19 < NumArq
                   Then Inc(Inicio,20)
                   Else
                     Begin
                       While Inicio+19 < NumArq do Inicio:=Inicio+20;
                       Posicao:=NumArq-Inicio;
                     End;
{up,pu} #72,#73  :  If Inicio >= 21 Then Dec(Inicio,20);
{end,^end,^pd}  #79,#117,#118 :  Begin
                         While Inicio+19 < NumArq do Inicio:=Inicio+20;
                         Posicao:=NumArq-Inicio;
                       End;
{home,^home,^pu} #71,#119,#132 :  Begin
                         Inicio:=1;
                         Posicao:=0;
                       End;
                     End;
      If Posicao < 0          {mostrar pagina acima}
        Then
          Begin
            Dec (Inicio,20);
            Inc (Posicao,20)
          End
        Else
          If Posicao > 19        {mostrar pagina abaixo}
            Then
              Begin
                Inc (Inicio,20);
                Dec (Posicao,20);
              End;
  Until Tecla in [#27,#13];
  Window (14,12,67,16);
  ClrScr;
  Window (1,1,80,25);
End;

Procedure Usuario;
{procedure para opcao de rodar programa do usuario}
  Var NomePrograma : String[80];
  Begin
    Cursor ('N');
    ApagaCentro;
    Window (14,12,67,16);
    Gotoxy (1,1);
```

```
    Writeln ('Digite o nome do programa a ser executado:');
    Gotoxy (1,2);
    Write ('> ');
    Readln(NomePrograma);
    ClrScr;
    Window (1,1,80,25);
    Exec  (NomePrograma,'');
    CodigoErro:=DosError;
    If CodigoErro <> 0
      Then
        Erro (NomePrograma);
    Cursor ('A');
    ClrScr;
  End;

Begin        {PROGRAMA PRINCIPAL}
CBreak:=False;                    {desliga interpretacao de Break}
TextAttr:=14;
{desenha a tela}
Posicao:=0;
Clrscr;
Repeat
  Cursor('A');
  Gotoxy (1,1);
  Write ('ZDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD?');
  For Int1:=2 to 23 do
    Begin
      Gotoxy (1,Int1);  Write ('3');
      Gotoxy (80,Int1); Write ('3');
    End;
  Write ('@DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDY');
  Gotoxy (2,2);
  Write ('ZDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD?');
  Gotoxy (2,3);
  Write ('3 PICTOREA   -   UMA  FERRAMENTA  DE  ENSINO  PARA  TRATAMENTO  DE  IMAGENS  3');
  Gotoxy (2,4);
  Write ('@DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDY');
  Gotoxy (2,5);
  Write ('ZDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD?');
  Gotoxy (2,6);
  Write ('3                      VERSAO  1.1   -   JULHO DE 1988                       3');
  Gotoxy (2,7);
  Write ('3                                   POR                                      3');
  Gotoxy (2,8);
  Write ('3    NEUCIMAR J. LEITE,  ARNALDO DE A. ARAUJO  E  JOAO M. DE CARVALHO       3');
  Gotoxy (2,9);
  Write ('@DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDY');
  Gotoxy (12,11);
  Write ('ZDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD?');
  Gotoxy (12,12);
  Write ('3  PROGRAMA SPDI                                       3');
  Gotoxy (12,13);
  Write ('3  PROGRAMA DO USUARIO                                 3');
  Gotoxy (12,14);
  Write ('3  DIRETORIO DE IMAGENS                                3');
  Gotoxy (12,15);
  Write ('3  AUXILIO                                             3');
  Gotoxy (12,16);
  Write ('3  ENCERRA                                             3');
```

```
 Gotoxy (12,17);
 Write ('@DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDY');
 Gotoxy (2,19);
 Write ('ZDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD?');
 Gotoxy (2,20);
 Write ('3        LABORATORIO DE SINAIS IMAGENS E COMPUTACAO GRAFICA              3');
 Gotoxy (2,21);
 Write ('3                     DEE / CCT / UFPB  -  CAMPUS  II                    3');
 Gotoxy (2,22);
 Write ('3             CX. POSTAL 10105  -  58100  CAMPINA GRANDE - PB            3');
 Gotoxy (2,23);
 Write ('@DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDY');

{ler teclado}
 InvertCampo (15,12+Posicao,22);
 Repeat
   Repeat
     Repeat
       Gotoxy (3,18);
       Write (PegueData,'                                                    ',
              PegueHora);
     Until Keypressed;
     Tecla:=ReadKey;
     If KeyPressed and (Tecla = #0)
       Then
         Begin
           Tecla:=ReadKey;
           OKey:=TRUE
         End
       Else
         OKey:=FALSE;
   Until ((Tecla in [#80 {dn}, #72 {up}, #71 {home}, #79 {end}]) and OKey) or
         ((Tecla = #13) and not Okey);

   InvertCampo (15,12+Posicao,22);
   Case Tecla of
{up}     #72  :  If Posicao > 0 Then Dec (Posicao) Else Posicao:=4;
{dn}     #80  :  If Posicao < 4 Then Inc (Posicao) Else Posicao:=0;
{home}   #71  :  Posicao:=0;
{end}    #79  :  Posicao:=4;
   End;
   InvertCampo (15,12+Posicao,22);

 Until Tecla = #13;        {teclou enter}

If Posicao in [0,2,3] Then Cursor ('N');
{interpretacao da opcao escolhida}
 Case Posicao of
{spdi}    0  :  Begin
                  Window (14,12,67,16);
                  ClrScr;
                  Writeln ('Lendo programa SPDI.EXE. Aguarde ...');
                  Exec ('Spdi.exe','');       {programa,parametros}
                  CodigoErro:=DosError;
                  If CodigoErro <> 0
                    Then
                      Begin
                        Gotoxy (1,1);
                        Writeln ('ERRO ao tentar acessar o programa:       ');
```

```
                    Writeln ('> SPDI.EXE');
                    Writeln;
                    Writeln ('Pressione <ENTER> ou <ESC>');
                    Repeat
                      LerSeta(Tecla);
                    Until Tecla in [#13,#27];
                  End;
                Window (1,1,80,25);
                ClrScr;
              End;
{user}    1 : Usuario;
{dir}     2 : Diretorio;
{aux}     3 : Begin
                Window (14,12,67,16);
                ClrScr;
                Writeln ('Aguarde...');
                Exec ('SpdiAux.exe','');
                CodigoErro:=DosError;
                If CodigoErro <> 0
                  Then
                    Begin
                      Gotoxy (1,1);
                      Writeln ('ERRO ao tentar acessar o programa:       ');
                      Writeln ('> SPDIAUX.EXE');
                      Writeln;
                      Writeln ('Pressione <ENTER> ou <ESC>');
                      Repeat
                        LerSeta(Tecla);
                      Until Tecla in [#13,#27];
                    End;
                Window (1,1,80,25);
                ClrScr;
              End;
  End;                                    {fim do case}
  Cursor ('A');
Until Posicao = 4;                        {loop tela principal}
Cursor ('N');
ClrScr;
End.                                      {fim do programa}
```

```
      PROGRAM PICTOREA
*************************************************************
*  DEFINE CARACTERISTICAS DO MODO CURSOR E MODO GRAFICO
*  E INICIALIZA O SISTEMA
*************************************************************
C
      IMPLICIT INTEGER*2 (C-Z)
      DIMENSION WORKIN(19),WORKOUT(66)
      COMMON /PRAESTATLOC/AREA,/PRAVOLTA/GRAFICO
C
      DATA WORKIN /11*1,68,73,83,80,76,65,89,32/
C
C     ABRIR ESTACAO DE TRABALHO:
C
      STATUS=VOPNWK(WORKIN,DISPLAY,WORKOUT)
      STATUS=VCRCOL(DISPLAY,1,4,FORE,BACK)
      STATUS=VCLRWK(DISPLAY)
C
C     INICIALIZA VARIAVEIS DE SINALIZACAO
C
      INDICA=0
      INDICB=0
      AREA=0.
      GRAFICO=0
C
      CALL MENUP(DISPLAY)
      END



      SUBROUTINE MENSAGE(DISPLAY,FLAG,*)
*************************************************************
*     APRESENTA MENSAGENS DE ERRO E INFORMACOES REFERENTES
*     A EXECUCAO DE PICTOREA
*************************************************************
C
      IMPLICIT INTEGER*2 (C-Z)
      CHARACTER BIP
      DIMENSION XY(4)
      COMMON/PRAVOLTA/GRAFICO
C
      XY(1)=0
      XY(2)=18000
      XY(3)=32000
      XY(4)=22000
      BIP=CHAR(7)
C
      STATUS=VRVON(DISPLAY)
      DO 10 I=1,4
      STATUS=VCURAD(DISPLAY,I,1)
10    STATUS=VEREOL(DISPLAY)
C
C     INDICA ATRAVES DE UM SINAL SONORO A OCORRENCIA DE UM ERRO NA
C     EXECUCAO DO PROGRAMA
C
      WRITE(*,*) BIP
      IF(FLAG.EQ.4.OR.FLAG.EQ.5) THEN
      STATUS=VCLARY(DISPLAY,XY,1,1,1,4,2)
      GOTO 20
     ENDIF
```

```
      STATUS=VCURAD(DISPLAY,1,2)
C
C     APRESENTAR MENSAGENS DE ERRO
C
20    GOTO (1,2,3,4,5) FLAG
C
1     STATUS=VCTXTS(DISPLAY,50,'*** Erro na especificacao da memoria de
     .imagem ***')
      GOTO 1000
2     STATUS=VCTXTS(DISPLAY,26,'*** Arquivo nao existe ***')
      GOTO 1000
3     STATUS=VCTXTS(DISPLAY,48,'*** Erro na especificacao do dado de ent
     .rada ***')
      GOTO 1000
C
C     ATENCAO! MODO GRAFICO
C
4     STATUS=VGTXTS(DISPLAY,100,IY(4)-1000,34,'*** Dado fora da faixa [0
     ...31] ***')
      GOTO 100
C
5     STATUS=VGTXTS(DISPLAY,100,IY(4)-1000,37,'*** Erro nos valores do i
     .ntervalo ***')
      GOTO 100
C
100   STATUS=VGTXTS(DISPLAY,100,IY(4)-2000,35,'Tecle <ENTER> para retorn
     .ar ao menu')
      GRAFICO=1
1000  FUNC=VOLTAR(DISPLAY)
      GRAFICO=0
C
      STATUS=VRVOFF(DISPLAY)
      RETURN1
      END
C
      INTEGER*2 FUNCTION VOLTAR(DISPLAY)
      IMPLICIT INTEGER*2 (C-Z)
      DIMENSION EXE(2)
      CHARACTER VOLTA
      COMMON/PRAVOLTA/GRAFICO
      IF(GRAFICO.EQ.1) GOTO 1001
      STATUS=VCURAD(DISPLAY,2,2)
      STATUS=VCRCOL(DISPLAY,1,2,EXE(1),EXE(2))
      STATUS=VCTXTS(DISPLAY,35,'Tecle <ENTER> para retornar ao menu')
      STATUS=VCRCOL(DISPLAY,1,4,EXE(1),EXE(2))
1001  VOLTA=' '
      STATUS=VRQSTR(DISPLAY,1,0,EXE,VOLTA)
      IF(VOLTA.NE.' ') GOTO 1001
      VOLTAR=1
      END
```

```
      SUBROUTINE MENSINF(DISPLAY,FLAG)
*************************************************************
*     APRESENTA MENSAGENS CONTENDO INFORMACOES A RESPEITO
*     DA EXECUCAO DO PROGRAMA
*************************************************************
C
      IMPLICIT INTEGER*2(C-Z)
      DIMENSION EXE(2)
      CHARACTER VOLTA
C
      DO 10 I=1,4
      STATUS=VCURAD(DISPLAY,1,1)
10    STATUS=VEREOL(DISPLAY)
C
      STATUS=VCURAD(DISPLAY,1,2)
      GOTO (1,2) FLAG
1     STATUS=VCTXTS(DISPLAY,76,'*** Fim de execucao. O resultado do proc
     .essamento esta armazenado em MI3 ***')
      GOTO 100
2     STATUS=VCTXTS(DISPLAY,24,'*** Fim de execucao. ***')
100   FUNC=VOLTAR(DISPLAY)
      END

      SUBROUTINE MENUP(DISPLAY)
*************************************************
*     APRESENTA MENU PRINCIPAL DE GPDI
*************************************************
      IMPLICIT INTEGER*2 (C-Z)
      CHARACTER NOME*14,STRING,STRIND,F(8)*29,TITULO*14
      LOGICAL EX
      COMMON IA(64,64),IB(64,64),IC(64,64),NOME,/COMUNIC/STRING,STRIND
     .     /SINAL/INDICA,INDICB
C
      DATA TITULO /'MENU PRINCIPAL'/
      DATA F(1) /'F1 Entrada/saida de imagens'/
      DATA F(2) /'F2 Display de imagens'/
      DATA F(3) /'F3 Histogramas e estatisticas'/
      DATA F(4) /'F4 Aritmetica de imagens'/
      DATA F(5) /'F5 Transf. radiometricas'/
      DATA F(6) /'F6 Filtros espaciais'/
      DATA F(7) /'F7 Geracao de imagens'/
      DATA F(8) /'F8 Geracao de ruidos '/
C
10    STATUS=VENCUR(DISPLAY)
C*****************************************************************
C     APRESENTA OPCOES
C
C*****************************************************************
C
11    CALL MCURSOR(DISPLAY,TITULO,14)
C
      L=1
      DO 5 J=10,16,2
      STATUS=VCURAD(DISPLAY,J,8)
      STATUS=VCTXTS(DISPLAY,29,F(L))
      STATUS=VCURAD(DISPLAY,J,39)
      STATUS=VCTXTS(DISPLAY,29,F(L+1))
      L=L+2
5     CONTINUE
```

```
C
       CALL FINAL(DISPLAY,0)
C*************************************************************
C*************************************************************
1000   STATUS=VRQCHC(DISPLAY,CHAIN,CAFIN)
C
       IF(CAFIN.EQ.1.AND.STATUS.GT.0)  CALL INOUT(DISPLAY)
C
       IF(CAFIN.EQ.2) THEN
       CALL PISCA (10,39,F(2),0,DISPLAY,21,*10)
       CALL IMAGEDIS(DISPLAY,*10)
       STATUS=VCLRWK(DISPLAY)
       GOTO 11
      ENDIF
C
       IF(CAFIN.EQ.3) CALL DSPHIS(DISPLAY)
C
       IF(CAFIN.EQ.4) CALL ARITMET(DISPLAY)
C
       IF(CAFIN.EQ.5) CALL TRANSF(DISPLAY)
C
       IF(CAFIN.EQ.6) CALL FILTRO(DISPLAY)
C
       IF(CAFIN.EQ.7) CALL GERAR(DISPLAY)
C
       IF(CAFIN.EQ.8) CALL GERUIDO(DISPLAY)
C
       IF(CAFIN.EQ.9) CALL AUXILIO(DISPLAY,7)
C
       IF(CAFIN.NE.10) GO TO 10
C
C     FECHAR ESTACAO DE TRABALHO:
C
       STATUS=VCRCOL(DISPLAY,1,0,L,J)
       STATUS=VCLRWK(DISPLAY)
       STATUS=VCLSWK(DISPLAY)
       STOP
       END

    SUBROUTINE IMAGEDIS(DISPLAY,*)
*****************************************************
*     APRESENTA DISPLAY DE IMAGENS
*****************************************************
       IMPLICIT INTEGER*2 (C-Z)
       DIMENSION EXE(2)
       CHARACTER NOME*14,NOME1*14,STRING,STRIND,STRIN1
       COMMON IA(64,64),IB(64,64),IC(64,64),NOME,/COMUNIC/STRING,STRIND
    ./POSICAO/XPOS,YPOS
C
       STATUS=VCURAD(DISPLAY,25,1)
       STATUS=VEREOL(DISPLAY)
       PRESS=ALTF(DISPLAY)
       STATUS=VCURAD(DISPLAY,23,10)
       STATUS=VCTXTS(DISPLAY,29,'Display de duas imagens(S/N)?')
80     STRIND=' '
       STATUS=VSMSTR(DISPLAY,1,0,EXE,STRIND)
       EINT=ICHAR(STRIND)
       IF(EINT.EQ.27) RETURN 1
       IF(STRIND.NE.'S'.AND.STRIND.NE.'s'.AND.STRIND.NE.'N'.AND.STRIND.NE
    ..'n') GOTO 80
       STATUS=VCTXTS(DISPLAY,1,STRIND)
```

```
      STATUS=VCURAD(DISPLAY,25,2)
      IF(STRIND.NE.'S'.AND.STRIND.NE.'s') GOTO 85
      CALL LEITURA(DISPLAY,31,*90)
      NOME1=NOME
      STRIN1=STRING
      CALL LEITURA(DISPLAY,32,*90)
      GOTO 86
C
C     DISPLAY DE UMA UNICA IMAGEM:
C
85    CALL LEITURA(DISPLAY,3,*90)
      NOME1=NOME
      STRIN1=STRING
      STRING=' '
86    CALL MOLDURA(DISPLAY)
      DM=0
      IX=2200
      IY=6000
      IF(STRING.NE.' ') GOTO 87
      IX=9000
      DM=1
87    IF(STRIN1.EQ.'1') CALL IMAGEM (IA,NOME1,DISPLAY,IX,IY,DM,1)
      IF(STRIN1.EQ.'2') CALL IMAGEM (IB,NOME1,DISPLAY,IX,IY,DM,2)
      IF(STRIN1.EQ.'3') CALL IMAGEM (IC,NOME1,DISPLAY,IX,IY,DM,3)
      IF(STRING.EQ.'1') CALL IMAGEM (IA,NOME,DISPLAY,17200,6000,1,1)
      IF(STRING.EQ.'2') CALL IMAGEM (IB,NOME,DISPLAY,17200,6000,1,2)
      IF(STRING.EQ.'3') CALL IMAGEM (IC,NOME,DISPLAY,17200,6000,1,3)
C
90    RETURN
      END
```

```
    SUBROUTINE ARITMET(DISPLAY)
*********************************************************
*     REALIZA A ARITMETICA DE IMAGENS
*********************************************************
C
      IMPLICIT INTEGER*2 (C-Z)
      CHARACTER NOME*14,NOME1*14,STRING,STRIND,SUBMENU*21,F(7)*70,
     .          STRIN1,STRIN2
      COMMON IA(64,64),IB(64,64),IC(64,64),NOME,/COMUNIC/STRING,STRIND
C
      DATA SUBMENU /'ARITMETICA DE IMAGENS'/
      DATA F(1) /'F1 Adicao de MI1 e MI2'/
      DATA F(2) /'F2 Subtracao de 2 imagens'/
      DATA F(3) /'F3 Diferenca absoluta entre MI1 e MI2'/
      DATA F(4) /'F4 Media da diferenca absoluta e Erro medio quadratico
     . entre 2 imagens'/
      DATA F(5) /'F5 Efeito zoom'/
      DATA F(6) /'F6 Reducao de imagem'/
      DATA F(7) /'F7 Display de imagens'/
C
C     APRESENTA OPCOES
C
10    STATUS=VCLRWK(DISPLAY)
C
      CALL MCURSOR(DISPLAY,SUBMENU,21)
C
```

```
      I=1
      DO 6 J=10,22,2
      STATUS=VCURAD(DISPLAY,J,10)
      STATUS=VCTITS(DISPLAY,70,F(I))
6     I=I+1
C
      CALL FINAL(DISPLAY,1)
C
      STATUS=VRQCHC(DISPLAY,CHAIN,CHAIN)
C
      IF(CHAIN.EQ.1.AND.STATUS.GT.0) THEN
      STATUS=VCURAD(DISPLAY,25,1)
      STATUS=VEREOL(DISPLAY)
      CALL PISCA(10,10,F(1),0,DISPLAY,22,*10)
      CALL ADICAD(IA,IB,IC,DISPLAY)
     ENDIF
C
      IF(CHAIN.EQ.2) THEN
      CALL PISCA(12,10,F(2),10,DISPLAY,25,*10)
      IF(STRING.EQ.'1') CALL SUBTRAC(IA,IB,IC,DISPLAY)
      IF(STRING.EQ.'2') CALL SUBTRAC(IB,IA,IC,DISPLAY)
     ENDIF
C
      IF(CHAIN.EQ.3) THEN
      STATUS=VCURAD(DISPLAY,25,1)
      STATUS=VEREOL(DISPLAY)
      CALL PISCA(14,10,F(3),0,DISPLAY,37,*10)
      CALL DIFABS(IA,IB,IC,DISPLAY)
     ENDIF
C
      IF(CHAIN.EQ.4) THEN
      CALL PISCA(16,10,F(4),31,DISPLAY,70,*10)
      STRIN1=STRING
      NOME1=NOME
      CALL LEITURA(DISPLAY,32,*10)
      IF(STRIN1.EQ.'1'.AND.STRING.EQ.'1') CALL ERROMED(IA,IA,NOME1,
     .NOME,1,1,DISPLAY)
      IF(STRIN1.EQ.'2'.AND.STRING.EQ.'2') CALL ERROMED(IB,IB,NOME1,
     .NOME,2,2,DISPLAY)
      IF(STRIN1.EQ.'3'.AND.STRING.EQ.'3') CALL ERROMED(IC,IC,NOME1,
     .NOME,3,3,DISPLAY)
      IF(STRIN1.EQ.'1'.AND.STRING.EQ.'2') CALL ERROMED(IA,IB,NOME1,
     .NOME,1,2,DISPLAY)
      IF(STRIN1.EQ.'2'.AND.STRING.EQ.'1') CALL ERROMED(IB,IA,NOME1,
     .NOME,2,1,DISPLAY)
      IF(STRIN1.EQ.'1'.AND.STRING.EQ.'3') CALL ERROMED(IA,IC,NOME1,
     .NOME,1,3,DISPLAY)
      IF(STRIN1.EQ.'3'.AND.STRING.EQ.'1') CALL ERROMED(IC,IA,NOME1,
     .NOME,3,1,DISPLAY)
      IF(STRIN1.EQ.'2'.AND.STRING.EQ.'3') CALL ERROMED(IB,IC,NOME1,
     .NOME,2,3,DISPLAY)
      IF(STRIN1.EQ.'3'.AND.STRING.EQ.'2') CALL ERROMED(IC,IB,NOME1,
     .NOME,3,2,DISPLAY)
     ENDIF
C
```

```
      IF(CHAIN.EQ.5) THEN
      STATUS=VCURAD(DISPLAY,24,8)
      STATUS=VCTXTS(DISPLAY,51,'O resultado do processamento esta armaze
     .nado em MI3')
      CALL PISCA(18,10,F(5),2,DISPLAY,14,*10)
      IF(STRING.EQ.'1') CALL ZOOM(IA,IC,NOME,DISPLAY,1)
      IF(STRING.EQ.'2') CALL ZOOM(IB,IC,NOME,DISPLAY,2)
     ENDIF
      IF(CHAIN.EQ.6) THEN
      CALL PISCA(20,10,F(6),1,DISPLAY,20,*10)
      IF(STRING.EQ.'1') CALL REDUCAO(IA,NOME,DISPLAY,1)
      IF(STRING.EQ.'2') CALL REDUCAO(IB,NOME,DISPLAY,2)
      IF(STRING.EQ.'3') CALL REDUCAO(IC,NOME,DISPLAY,3)
     ENDIF
C
      IF(CHAIN.EQ.7) THEN
      CALL PISCA(22,10,F(7),0,DISPLAY,21,*10)
      CALL IMAGEDIS(DISPLAY,*10)
     ENDIF
C
      IF(CHAIN.EQ.9) CALL AUXILIO(DISPLAY,3)
C
      IF(CHAIN.NE.10) GOTO 10
      END

    SUBROUTINE MCURSOR(DISPLAY,SUBMENU,NUM)
*****************************************************************
*     ESCREVE TEXTO PARA SUBMENU E ALTERA ALGUMAS CARACTERISTICAS
*     DO MODO CURSOR
*****************************************************************
C
      IMPLICIT INTEGER*2 (C-Z)
      CHARACTER SUBMENU*30
C
      STATUS=VCURAD(DISPLAY,7,14)
      STATUS=VCTXTS(DISPLAY,NUM,SUBMENU)
      END


   SUBROUTINE FINAL(DISPLAY,SINAL)
*****************************************************
*     APRESENTA AS ULTIMAS OPCOES DOS MENUS
*     SINAL:0- AUXILIO,FIM
*           1- AUXILIO,RETORNA
*           2- RETORNA E MENU PRINCIPAL
*****************************************************
C
      IMPLICIT INTEGER*2 (C-Z)
      COMMON /POSICAO/ LINHA,COLUNA
C
      COLUNA=55
      STATUS=VRVON(DISPLAY)
      STATUS=VCURAD(DISPLAY,1,COLUNA)
      IF(SINAL.EQ.2) GOTO 4
      STATUS=VCTXTS(DISPLAY,26,'F9 Auxilio              ')
      STATUS=VCURAD(DISPLAY,2,COLUNA)
      IF(SINAL.NE.0) GOTO 4
      STATUS=VCTXTS(DISPLAY,26,'F10 Fim                 ')
      GOTO 5
4     STATUS=VCTXTS(DISPLAY,26,'F10 Retorna             ')
```

```
      IF(SINAL.EQ.1) GOTO 5
      POS=3
      IF(SINAL.EQ.2) POS=2
      STATUS=VCURAD(DISPLAY,POS,COLUNA)
      STATUS=VCTXTS(DISPLAY,26,'OUTRA TECLA Menu principal')
C
5     STATUS=VQCURA(DISPLAY,LINHA,YOUT)
      STATUS=VCURAD(DISPLAY,25,1)
      STATUS=VCTXTS(DISPLAY,15,'        OPCAO: ')
      STATUS=VRVOFF(DISPLAY)
      END

    SUBROUTINE MASCARA(IA,IC,DISPLAY,FLAG)
*****************************************************************
*     FAZ A LEITURA DA MASCARA DE CONVOLUCAO PARA REALIZACAO
*     DA FILTRAGEM ESPACIAL
*****************************************************************
C
      IMPLICIT INTEGER*2 (C-Z)
      CHARACTER OPCAO
      DIMENSION IA(64,64),IC(64,64),MASC(25),XYEND(2)
      COMMON/POSICAO/XPOS,YPOS
C
      XPOS=1
      YPOS=55
      STATUS=VCLRWK(DISPLAY)
      PRESS=ALTF(DISPLAY)
C
C     LER TAMANHO DA MASCARA
C
      STATUS=VCURAD(DISPLAY,10,28)
      STATUS=VRVON(DISPLAY)
      STATUS=VCTXTS(DISPLAY,22,'Mascara? F1-3*3 F2-5*5')
C
50    OPCAO=' '
      STATUS=VRQSTR(DISPLAY,1,0,XYEND,OPCAO)
      EINT=ICHAR(OPCAO)
      STATUS=VRVOFF(DISPLAY)
      IF(EINT.EQ.27) GOTO 1000
      IF(EINT.LT.59.OR.EINT.GT.60) GOTO 50
      CHAIN=EINT-58
      STATUS=VCURAD(DISPLAY,10,28)
      STATUS=VCTXTS(DISPLAY,22,'                    ')
      STATUS=VRVON(DISPLAY)
      DIM=3
      IF(CHAIN.EQ.2) DIM=5
      STATUS=VCURAD(DISPLAY,14,16)
C
      GOTO (5,10) CHAIN
C
5     STATUS=VCTXTS(DISPLAY,48,'Indique valores p/ a mascara 3*3 e tecle'
     . <ENTER>')
      STATUS=VCURAD(DISPLAY,16,16)
      STATUS=VCTXTS(DISPLAY,9,'*   *   *')
C
      GOTO 20
C
10    STATUS=VCTXTS(DISPLAY,48,'Indique valores p/ a mascara 5*5 e tecle'
     . <ENTER>')
      STATUS=VCURAD(DISPLAY,16,16)
      STATUS=VCTXTS(DISPLAY,17,'*   *   *   *   *')
```

```
C
20      I=16
        DO 25 J=1,DIM
        I=I+1
        STATUS=VCURAD(DISPLAY,I,13)
25      STATUS=VCTXTS(DISPLAY,1,'*')
        STATUS=VRVOFF(DISPLAY)
C
C     LER VALORES DE ENTRADA
C
        I=17
        K=0
        DO 30 J=1,DIM
        Y=12
        DO 35 J=1,DIM
        Y=Y+4
        STATUS=VCURAD(DISPLAY,I,Y)
        K=K+1
        CALL LER12 (DISPLAY,5,MASC(K),6,*1000)
35      CONTINUE
        I=I+1
30      CONTINUE
C
C     EXECUTAR POSSIVEIS ALTERACOES NOS VALORES DE ENTRADA
C
40      CALL QUESTAO(DISPLAY,*100,*99)
C
C     FAZER MODIFICACOES
C
        CALL MODIFIQUE( DISPLAY,MASC,DIM,*100,*1000)
99      RETURN
C
C     REALIZAR OPERACAO DE CONVOLUCAO
C
100     GOTO (101,102) CHAIN
101     CALL CONV03(IA,IC,MASC,DISPLAY,FLAG)
        RETURN
102     CALL CONV05(IA,IC,MASC,DISPLAY,FLAG)
C
1000    RETURN
        END
C
C
        SUBROUTINE QUESTAO(DISPLAY,*,*)
        IMPLICIT INTEGER*2 (C-Z)
        DIMENSION EXE(2)
        CHARACTER STRING
C
        STATUS=VCURAD(DISPLAY,22,20)
        STATUS=VCTXTS(DISPLAY,24,'Alguma alteracao (s/n)? ')
        STATUS=VCURAD(DISPLAY,22,43)
40      STRING=' '
        STATUS=VRQSTR(DISPLAY,1,0,EXE,STRING)
        EINT=ICHAR(STRING)
        IF(EINT.EQ.0) GOTO 40
C
C     VERIFICAR SE OPCAO <ESC> (ABANDONA)
C
```

```
      IF(EINT.EQ.27) RETURN 2
C
      IF(STRIND.NE.'S'.AND.STRIND.NE.'s'.AND.STRIND.NE.'N'.AND.STRIND.
     .NE.'n') GOTO 40
      STATUS=VCTXTS(DISPLAY,1,STRIND)
      IF(STRIND.NE.'S'.AND.STRIND.NE.'s') RETURN 1
C
      return
      END
C
C
      SUBROUTINE MODIFIQUE(DISPLAY,MASC,DIM,*,*)
************************************************************
*     EXECUTA POSSIVEIS MODIFICACOES NA MASCARA DE ENTRADA
************************************************************
C
      IMPLICIT INTEGER*2 (C-Z)
      DIMENSION MASC(25)
C
1     X=17
      CONTA=0
      DO 10 I=1,DIM
      Y=12
      DO 15 J=1,DIM
      Y=Y+4
      STATUS=VCURAD(DISPLAY,X,Y)
      CALL LER12(DISPLAY,5,RESERV,6,*1000)
      CONTA=CONTA+1
      IF(RESERV.EQ.0) GOTO 15
      MASC(CONTA)=RESERV
15    CONTINUE
      X=X+1
10    CONTINUE
      CALL QUESTAO(DISPLAY,*100,*1001)
      GOTO 1
C
100   RETURN 1
1000  RETURN 2
1001  RETURN
      END
C
C
      SUBROUTINE CONV03(IA,IC,MASC,DISPLAY,FLAG)
*********************************************************************
*     REALIZA FILTRAGEM ATRAVES DE UMA MASCARA DE CONVOLUCAO 3*3
*********************************************************************
C
      IMPLICIT INTEGER*2 (C-Z)
      DIMENSION IA(64,64),IC(64,64),MASC(25),XY(4)
      DATA PXIM,PYIM /17500,3500/
C
C     DETERMINA O PESO DA MASCARA
C
      PESO=0
      DO 5 I=1,9
5     PESO=PESO+MASC(I)
C
C     INFORMA SOBRE MASCARA NAO VALIDA
C
```

```
      IF(PESO.EQ.0) PESO=1
C
      CALL MOLDURA(DISPLAY)
      STATUS=VGTXTS(DISPLAY,5000,20810,22,'Convolucao:Mascara 3*3')
      CALL IMAGEM(IA,'Original      ',DISPLAY,2000,3900,0,FLAG)
      STATUS=VGTXTS(DISPLAY,PXIM+2000,PYIM+14700,10,'Processada')
C
C
      XY(2)=PYIM+13440+210
      DO 10 X=2,63
      XY(1)=PXIM-210
      XY(2)=XY(2)-210
      XY(4)=XY(2)+210
      DO 10 Y=2,63
      XY(1)=XY(1)+210
      XY(3)=XY(1)+210
      IC(X,Y)=(IA(X-1,Y-1)*MASC(9)+IA(X-1,Y)*MASC(8)+IA(X-1,Y+1)*MASC(7)
     .        +IA(X,Y-1)*MASC(6)  +IA(X,Y)*MASC(5)  +IA(X,Y+1)*MASC(4)
     .        +IA(X+1,Y-1)*MASC(3)+IA(X+1,Y)*MASC(2)
     .        +IA(X+1,Y+1)*MASC(1))/PESO
      IF(IC(X,Y).LT.0) IC(X,Y)=0
      IF(IC(X,Y).GT.31)IC(X,Y)=31
C
C     APRESENTAR IMAGEM
C
      CALL IMAGEP(IC(X,Y),XY,DISPLAY)
10    CONTINUE
C
      CALL COMPLET(IC,IA,1,DISPLAY,FLAG)
100   RETURN
      END
C
C
      SUBROUTINE CONV05(IA,IC,MASC,DISPLAY,FLAG)
*****************************************************************
*     REALIZA FILTRAGEM ATRAVES DE UMA MASCARA DE CONVOLUCAO 5*5
*****************************************************************
C
      IMPLICIT INTEGER*2 (C-Z)
      DIMENSION IA(64,64),IC(64,64),MASC(25),XY(4)
      DATA PXIM,PYIM /17500,3500/
C
C     DETERMINA PESO DA MASCARA
C
      PESO=0
      DO 5 I=1,25
5     PESO=PESO+MASC(I)
C
C     INFORMA SOBRE MASCARA NAO VALIDA
C
      IF(PESO.EQ.0) PESO=1
C
      CALL MOLDURA(DISPLAY)
      STATUS=VGTXTS(DISPLAY,5000,20810,22,'Convolucao:Mascara 5*5')
      CALL IMAGEM(IA,'Original      ',DISPLAY,2000,4110,0,FLAG)
      STATUS=VGTXTS(DISPLAY,PXIM+2000,PYIM+14700,10,'Processada')
C
```

```
      IY(2)=PYIM+13440+210
      DO 10 X=3,62
      IY(1)=PYIM-210
      IY(2)=IY(2)-210
      IY(4)=IY(2)+210
      DO 10 Y=3,62
      IY(1)=IY(1)+210
      IY(3)=IY(1)+210
      IC(X,Y)=(IA(X-2,Y-2)*MASC(25)+IA(X-2,Y-1)*MASC(24)
     .        +IA(X-2,Y)*MASC(23)  +IA(X-2,Y+1)*MASC(22)
     .        +IA(X-2,Y+2)*MASC(21)+IA(X-1,Y-2)*MASC(20)
     .        +IA(X-1,Y-1)*MASC(19)+IA(X-1,Y)*MASC(18)
     .        +IA(X-1,Y+1)*MASC(17)+IA(X-1,Y+2)*MASC(16)
     .        +IA(X,Y-2)*MASC(15)  +IA(X,Y-1)*MASC(14)
     .        +IA(X,Y)*MASC(13)    +IA(X,Y+1)*MASC(12)
     .        +IA(X,Y+2)*MASC(11)  +IA(X+1,Y-2)*MASC(10)
     .        +IA(X+1,Y-1)*MASC(9) +IA(X+1,Y)*MASC(8)
     .        +IA(X+1,Y+1)*MASC(7) +IA(X+1,Y+2)*MASC(6)
     .        +IA(X+2,Y-2)*MASC(5) +IA(X+2,Y-1)*MASC(4)
     .        +IA(X+2,Y)*MASC(3)   +IA(X+2,Y+1)*MASC(2)
     .        +IA(X+2,Y+2)*MASC(1))/PESO
      IF(IC(X,Y).LT.0) IC(X,Y)=0
      IF(IC(X,Y).GT.31)IC(X,Y)=31
C
C     APRESENTAR IMAGEM
C
      CALL IMAGEP(IC(X,Y),IY,DISPLAY)
10    CONTINUE
C
      CALL COMPLET(IC,IA,2,DISPLAY,FLAG)
C
100   RETURN
      END
```

```
    SUBROUTINE AUXILIO(DISPLAY,FLAG)
*********************************************************
*     APRESENTA INFORMACOES DE AUXILIO PARA O USUARIO
*********************************************************
C
      IMPLICIT INTEGER*2 (C-Z)
C
      STATUS=VCLRWK(DISPLAY)
C
      STATUS=VCURAD(DISPLAY,8,10)
      YPOS=9
      GOTO(1,2,3,4,5,6,7,8) FLAG
C
1     STATUS=VCTXTS(DISPLAY,14,'E/S DE IMAGENS')
      STATUS=VCURAD(DISPLAY,10,5)
      STATUS=VCTXTS(DISPLAY,70,'Apresenta um conjunto de funcoes que rea
     .lizam o movimento de  arquivos')
      STATUS=VCURAD(DISPLAY,11,5)
      STATUS=VCTXTS(DISPLAY,22,'de imagens no sistema.')
      YPOS=13
      GOTO 10
C
```

```
2       STATUS=VCTXTS(DISPLAY,26,'HISTOGRAMAS E ESTATISTICAS')
        STATUS=VCURAD(DISPLAY,10,5)
        STATUS=VCTXTS(DISPLAY,70,'Apresenta um conjunto de funcoes que for
     .necem algumas  medidas estatis')
        STATUS=VCURAD(DISPLAY,11,5)
        STATUS=VCTXTS(DISPLAY,18,'ticas das imagens.')
        STATUS=VCURAD(DISPLAY,12,5)
        STATUS=VCTXTS(DISPLAY,47,'A especificacao do NOME DO ARQUIVO e" op
     .cional.')
        YPOS=14
        GOTO 10
C
3       STATUS=VCTXTS(DISPLAY,21,'ARITMETICA DE IMAGENS')
        STATUS=VCURAD(DISPLAY,10,5)
        STATUS=VCTXTS(DISPLAY,70,'Apresenta um conjunto de funcoes que rea
     .lizam algumas  operacoes entre')
        STATUS=VCURAD(DISPLAY,11,5)
        STATUS=VCTXTS(DISPLAY,20,'arquivos de imagens.')
        STATUS=VCURAD(DISPLAY,12,5)
        STATUS=VCTXTS(DISPLAY,47,'A especificacao do NOME DO ARQUIVO e" op
     .cional.')
        YPOS=14
        GOTO 10
C
4       STATUS=VCTXTS(DISPLAY,28,'TRANSFORMACOES RADIOMETRICAS')
        STATUS=VCURAD(DISPLAY,10,5)
        STATUS=VCTXTS(DISPLAY,70,'Apresenta um conjunto de funcoes que mod
     .ificam diretamente o valor dos')
        STATUS=VCURAD(DISPLAY,11,5)
        STATUS=VCTXTS(DISPLAY,26,'niveis de cinza da imagem.')
        STATUS=VCURAD(DISPLAY,12,5)
        STATUS=VCTXTS(DISPLAY,47,'A especificacao do NOME DO ARQUIVO e" op
     .cional.')
        STATUS=VCURAD(DISPLAY,13,5)
        STATUS=VCTXTS(DISPLAY,52,'O resultado do processamento esta armaze
     .nado em MI3.')
        YPOS=15
        GOTO 10
C
5       STATUS=VCTXTS(DISPLAY,17,'FILTROS ESPACIAIS')
        STATUS=VCURAD(DISPLAY,10,5)
        STATUS=VCTXTS(DISPLAY,70,'Apresenta um conjunto de filtros que atu
     .am diretamente  nos pixels  da')
        STATUS=VCURAD(DISPLAY,11,5)
        STATUS=VCTXTS(DISPLAY,70,'imagem, resultando  numa suavizacao ou a
     .umento de contraste  da imagem')
        STATUS=VCURAD(DISPLAY,12,5)
        STATUS=VCTXTS(DISPLAY,9,'original.')
        STATUS=VCURAD(DISPLAY,13,5)
        STATUS=VCTXTS(DISPLAY,47,'A especificacao do NOME DO ARQUIVO e" op
     .cional.')
        STATUS=VCURAD(DISPLAY,14,5)
        STATUS=VCTXTS(DISPLAY,52,'O resultado do processamento esta armaze
     .nado em MI3.')
        YPOS=16
        GOTO 10
C
```

```
6       STATUS=VCTXTS(DISPLAY,18,'GERACAO DE IMAGENS')
        STATUS=VCURAD(DISPLAY,10,5)
        STATUS=VCTXTS(DISPLAY,70,'Apresenta um conjunto de funcoes que ger
     .am  algumas imagens, de acordo')
        STATUS=VCURAD(DISPLAY,11,5)
        STATUS=VCTXTS(DISPLAY,42,'com os parametros fornecidos pelo usuari
     .o.')
        STATUS=VCURAD(DISPLAY,12,5)
        STATUS=VCTXTS(DISPLAY,47,'A especificacao do NOME DO ARQUIVO e" op
     .cional.')
        YPOS=14
        GOTO 10
C
8       STATUS=VCTXTS(DISPLAY,17,'GERACAO DE RUIDOS')
        STATUS=VCURAD(DISPLAY,10,5)
        STATUS=VCTXTS(DISPLAY,70,'Permite a adicao de ruidos do tipo GAUSS
     .IANO e UNIFORME as imagens do ')
        STATUS=VCURAD(DISPLAY,11,5)
        STATUS=VCTXTS(DISPLAY,8,'sistema.')
        STATUS=VCURAD(DISPLAY,12,5)
        STATUS=VCTXTS(DISPLAY,47,'A especificacao do NOME DO ARQUIVO e" op
     .cional.')
        STATUS=VCURAD(DISPLAY,13,5)
        STATUS=VCTXTS(DISPLAY,52,'O resultado do processamento esta armaze
     .nado em MI3.')
        YPOS=15
C
10      STATUS=VCURAD(DISPLAY,YPOS-1,5)
        STATUS=VCTXTS(DISPLAY,65,'MI1 representa o "default" na especifica
     .cao da memoria de imagem.')
        GOTO 11

7       STATUS=VCTXTS(DISPLAY,14,'MENU PRINCIPAL')
11      STATUS=VCURAD(DISPLAY,YPOS+1,5)
        STATUS=VCTXTS(DISPLAY,70,'Aperte uma das teclas de funcao especial
     . (F1..F10) para selecionar uma')
        STATUS=VCURAD(DISPLAY,YPOS+2,5)
        STATUS=VCTXTS(DISPLAY,15,'funcao do menu.')
        IF(FLAG.NE.7) GOTO 12
        STATUS=VCURAD(DISPLAY,YPOS+3,5)
        STATUS=VCTXTS(DISPLAY,65,'MI1 representa o "default" na especifica
     .cao da memoria de imagem.')
12      STATUS=VCURAD(DISPLAY,YPOS+5,5)
        STATUS=VCTXTS(DISPLAY,55,'Para maiores informacoes consultar o MAN
     .UAL DO USUARIO.')
C
C     RETORNA AO MENU
C
        FUNC=VOLTAR(DISPLAY)
C
        END
```

```
      SUBROUTINE PPONTOG(IA,IC,DISPLAY,FLAG)
*****************************************************************
*     REALIZA A DETECAO DE BORDAS UTILIZANDO O GRADIENTE DE
*     ROBERTS
*****************************************************************
C
      IMPLICIT INTEGER*2 (C-Z)
      DIMENSION IA(64,64),IC(64,64),XY(4)
      DATA PXIM,PYIM /17500,3400/
C
      CALL MOLDURA(DISPLAY)
      STATUS=VGTXTS(DISPLAY,5500,20810,29,'Aplicacao direta do gradien
     .te')
      CALL IMAGEM(IA,'Original      ',DISPLAY,2000,3600,0,FLAG)
      STATUS=VGTXTS(DISPLAY,PXIM+2000,PYIM+14400,10,'Processada')
C
      XY(2)=PYIM+13440+210
      DO 5 X=1,63
      XY(1)=PXIM-210
      XY(2)=XY(2)-210
      XY(4)=XY(2)+210
      DO 5 Y=1,63
      XY(1)=XY(1)+210
      XY(3)=XY(1)+210
      IC(X,Y)=IMAGRAD(X,Y,IA)
      CALL IMAGEP(IC(X,Y),XY,DISPLAY)
5     CONTINUE
C
C     COMPLETAR MEMORIA DE IMAGEM
C
      CALL COMPGRAD(PXIM,PYIM,XY,IA,IC,DISPLAY,FLAG)
C
      RETURN
      END
C
C
      SUBROUTINE IMAGEP(PONTOP,XY,DISPLAY)
*****************************************************************
*     APRESENTA IMAGEM NA TELA DE ACORDO COM OS VALORES
*     DE PONTOP
*****************************************************************
C
      IMPLICIT INTEGER*2 (C-Z)
      DIMENSION XY(4)
C
      COR=0
      IF(PONTOP.GE.8) COR=2
      IF(PONTOP.GE.16) COR=3
      IF(PONTOP.GE.24) COR=1
      STATUS=VCLARY(DISPLAY,XY,1,1,1,4,COR)
      RETURN
      END
C
C
```

```
      INTEGER*2 FUNCTION IMAGRAD(X,Y,IA)
*************************************
*     REALIZA O CALCULO DO GRADIENTE
*************************************
C
      IMPLICIT INTEGER*2 (C-Z)
      DIMENSION IA(64,64)
      IMAGRAD=IABS(IA(X,Y)-IA(X+1,Y+1))+IABS(IA(X+1,Y)-IA(X,Y+1))
      IF(IMAGRAD.GT.31) IMAGRAD=31
C
      RETURN
      END
C
C
      SUBROUTINE COMPGRAD (PXIM,PYIM,XY,IA,IC,DISPLAY,FLAG)
*************************************************************
*     COMPLETA MEMORIA DE IMAGEM PARA GRADIENTE
*************************************************************
C
      IMPLICIT INTEGER*2 (C-Z)
      DIMENSION IA(64,64),IC(64,64),XY(4)
C

      XY(1)=XY(1)+210
      XY(3)=XY(1)+210
      XY(2)=PYIM+13440+210
      DO 10 X=1,64
      XY(2)=XY(2)-210
      XY(4)=XY(2)+210
      IC(X,64)=IC(X,63)
      CALL IMAGEP(IC(X,64),XY,DISPLAY)
10    CONTINUE
C
      XY(2)=XY(2)-210+210
      XY(1)=PXIM-210
      DO 15 Y=1,64
      XY(1)=XY(1)+210
      XY(3)=XY(1)+210
      IC(64,Y)=IC(63,Y)
      CALL IMAGEP(IC(64,Y),XY,DISPLAY)
15    CONTINUE
C
      CALL CRFIM(DISPLAY)
C
      END

    SUBROUTINE COMFUNDO(IA,IC,DISPLAY,FLAG)
*************************************************************
*     REALIZA TRANSFORMACAD GRADIENTE COM PLANO DE FUNDO DEFINIDO
*************************************************************
C
      IMPLICIT INTEGER*2 (C-Z)
      CHARACTER TEXT*2
      DIMENSION IA(64,64),IC(64,64),XY(4)
      DATA PXIM,PYIM /17500,3400/
C
C     LER VALORES PARA LIMIAR E FUNDO
C
```

```
      CALL ENTRE(DISPLAY,LIMIAR,LB,6,*100)
      STATUS=VGTXTS(DISPLAY,5500,20810,28,'Gradiente com fundo definido'
     .)
      STATUS=VSAPOS(DISPLAY,5500,19500,I,J)
      STATUS=VATXTS(DISPLAY,7,'Limiar=',1,J)
      WRITE(TEXT,'(I2)') LIMIAR
      STATUS=VATXTS(DISPLAY,2,TEXT,I,J)
      STATUS=VATXTS(DISPLAY,8,', Fundo=',I,J)
      WRITE(TEXT,'(I2)') LB
      STATUS=VATXTS(DISPLAY,2,TEXT,I,J)
C
      CALL IMAGEM(IA,'Original      ',display,2000,3600,0,FLAG)
      STATUS=VGTXTS(DISPLAY,PXIM+2000,PYIM+14400,10,'Processada')
C
      XY(2)=PYIM+13440+210
      DO 5 I=1,63
      XY(1)=PXIM-210
      XY(2)=XY(2)-210
      XY(4)=XY(2)+210
      DO 5 Y=1,63
      XY(1)=XY(1)+210
      XY(3)=XY(1)+210
      GRD=IMAGRAD(X,Y,IA)
      IC(X,Y)=LB
      IF(GRD.GE.LIMIAR) IC(X,Y)=GRD
C
C     APRESENTAR RESULTADO DA TRANSFOMACAO NA TELA
C
      CALL IMAGEP(IC(X,Y),XY,DISPLAY)
C
5     CONTINUE
C
      CALL COMPGRAD(PXIM,PYIM,XY,IA,IC,DISPLAY,FLAG)
C
100   RETURN
      END



      SUBROUTINE GERAR(DISPLAY)
*************************************************
*     GERA ALGUMAS FIGURAS E ARMAZENA NA MEMORIA
*     DE IMAGEM ESPECIFICADA PELO USUARIO
*************************************************
C
      IMPLICIT INTEGER*2 (C-Z)
      CHARACTER NOME*14,STRING,STRIND,SUBMENU*18,F(5)*21
      COMMON IA(64,64),IB(64,64),IC(64,64),NOME,/COMUNIC/STRING,STRIND
      DATA SUBMENU /'GERACAO DE IMAGENS'/
      DATA F(1) /'F1 Circulo'/
      DATA F(2) /'F2 Retangulo'/
      DATA F(3) /'F3 Quatro quadrados'/
      DATA F(4) /'F4 Tabuleiro'/
      DATA F(5) /'F5 Display de imagens'/
C
5     STATUS=VCLRWK(DISPLAY)
C
6     CALL MCURSOR(DISPLAY,SUBMENU,18)
C
```

```
        I=1
        DO 4 J=10,18,2
        STATUS=VCURAD(DISPLAY,J,10)
        STATUS=VCTXTS(DISPLAY,21,F(I))
4       I=I+1
C
        CALL FINAL (DISPLAY,1)
C
        STATUS=VRQCHC(DISPLAY,CHAIN,CHAIN)
        IF(CHAIN.EQ.1.AND.STATUS.GT.0) THEN
        CALL PISCA(10,10,F(1),4,DISPLAY,10,*5)
        GOTO 100
      ENDIF
        IF(CHAIN.EQ.2) THEN
        CALL PISCA(12,10,F(2),4,DISPLAY,12,*5)
        GOTO 100
      ENDIF
        IF(CHAIN.EQ.3) THEN
        CALL PISCA(14,10,F(3),4,DISPLAY,19,*5)
        GOTO 100
      ENDIF
        IF(CHAIN.EQ.4) THEN
        CALL PISCA(16,10,F(4),4,DISPLAY,12,*5)
        GOTO 100
      ENDIF
C
        IF(CHAIN.EQ.5) THEN
        CALL PISCA(18,10,F(5),0,DISPLAY,21,*5)
        CALL IMAGEDIS(DISPLAY,*5)
      ENDIF
C
        IF(CHAIN.EQ.9) CALL AUXILID(DISPLAY,6)
C
90      IF (CHAIN.NE.10) GOTO 5
        RETURN
C
100     IF(STRING.EQ.'1') CALL GERADOR(1A,DISPLAY,CHAIN,1)
        IF(STRING.EQ.'2') CALL GERADOR(1B,DISPLAY,CHAIN,2)
        IF(STRING.EQ.'3') CALL GERADOR(1C,DISPLAY,CHAIN,3)
        GOTO 90
        END

        SUBROUTINE LER12(DISPLAY,NCHAR,VALOR,SINAL,*)
*************************************************************************
*     REALIZA A LEITURA DE NOME DE ARQUIVO E MEMORIA DE IMAGEM
*     E LER 1 OU 2 CARACTERES E CONVERTE-OS PARA VALORES INTEIROS
*     ESTA SUBROTINA EH UTILIZADA POR GERADOR,LER,ENTRE,MASCARA E RUIDO
*************************************************************************
C
        IMPLICIT INTEGER*2 (C-Z)
        REAL CNTREAL,NEGATIV
        DIMENSION EXE(2)
        CHARACTER LETRA(23),VREAL*8
        COMMON/PRALER/ LETRA,/SIGNDP/CNTREAL
        EQUIVALENCE(LETRA(1),VREAL)
C
        VALOR=0
        VALOR1=0
        FLAG=0
        PONTO=1
        NEGATIV=1.
```

```
      DO 10 I=1,8
10     LETRA(I)=' '
       STATUS=VQCURA(DISPLAY,LINI,LIMY)
       SINALK=0
       I=1
20     STATUS=VRDSTR(DISPLAY,1,0,EXE,LETRA(I))
       EINT=ICHAR(LETRA(I))
       IF(LETRA(I).EQ.' '.and.nchar.eq.24) RETURN
       if(letra(i).eq.' '.and.NCHAR.NE.24) GOTO 7
       IF(EINT.EQ.0) THEN
       SINALK=1
       GOTO 20
      ENDIF
       IF(EINT.EQ.27) THEN
       STATUS=VRVOFF(DISPLAY)
       RETURN 1
      ENDIF
       IF(EINT.EQ.75.AND.SINALK.EQ.1) THEN
       SINALK=0
       letra(i)=' '
       IF(I.NE.1) I=I-1
      LETRA(I)=' '
       STATUS=VQCURA(DISPLAY,LINHA,COLUNA)
       STATUS=VCURAD(DISPLAY,LINHA,COLUNA-1)
       STATUS=VCTITS(DISPLAY,1,' ')
       IF((COLUNA-1).LT.LIMY) COLUNA=COLUNA+1
       STATUS=VCURAD(DISPLAY,LINHA,COLUNA-1)
       GOTO 20
      ENDIF
       STATUS=VCTITS(DISPLAY,1,LETRA(I))
       I=I+1
       IF(I.NE.NCHAR) GOTO 20
C
       IF(NCHAR.EQ.24) RETURN
C
C
C     RECEBER DADOS REAIS (SUBROTINAS ENTRE E RUIDO)
C
7      IF(SINAL.GT.6) THEN
       IF(SINAL.EQ.7.AND.LETRA(6).NE.' ') GOTO 1000
       IF(LETRA(1).EQ.'.') GOTO 55
       IF(LETRA(2).EQ.'.') THEN
       ASSIGN 2 TO FMTO
       PONTO=2
       ELSE IF(LETRA(3).EQ.'.') THEN
       ASSIGN 3 TO FMTO
       PONTO=3
       ELSE IF(LETRA(4).EQ.'.') THEN
       ASSIGN 4 TO FMTO
       PONTO=4
      ENDIF
       IF(SINAL.EQ.7.OR.LETRA(5).NE.'.') GOTO 54
       ASSIGN 5 TO FMTO
       PONTO=5
54     IF(PONTO.EQ.1) GOTO 1000
       READ(VREAL,FMTO,ERR=1000) VALOR
C
C     PARTE FRACIONARIA (1 DIGITO)
C
```

```
55      READ(LETRA(PONTO+1),'(III)',ERR=1000) VALOR1
        IF(VALOR.LT.0) NEGATIV=-1.
        CNTREAL=VALOR+(VALOR1/10.)*NEGATIV
        IF(SINAL.EQ.7.AND.(CNTREAL.LT.0.OR.CNTREAL.GT.50.0)) GOTO 1000
        RETURN
C
2       FORMAT(III)
3       FORMAT(II2)
4       FORMAT(II3)
5       FORMAT(II4)
      ENDIF

C
C     RECEBER VALORES INTEIROS (SUBROTINAS ENTRE,MASCARA E GERADOR)
C
        IF(LETRA(4).NE.' ') GOTO 1000
        ASSIGN 4 TO FMTO
        IF(LETRA(3).EQ.' ') ASSIGN 3 TO FMTO
        IF(LETRA(2).EQ.' ') ASSIGN 2 TO FMTO
        IF(LETRA(1).EQ.',',.OR.LETRA(2).EQ.',',.OR.LETRA(3).EQ.',',.
     -OR.LETRA(2).EQ.' '.AND.(LETRA(1).EQ.'-'.OR.LETRA(1).EQ.'+'))
     .GOTO 1000
        READ(VREAL,FMTO,ERR=1000) VALOR
C
C     TESTES
C
        IF(VALOR.LT.1.AND.(SINAL.EQ.0.OR.SINAL.EQ.2.OR.SINAL.EQ.4.OR.
     -SINAL.EQ.5).OR.VALOR.GT.64.
     .AND.(SINAL.EQ.2.OR.SINAL.EQ.3).OR.SINAL.EQ.0.AND.VALOR.GT.9.
     .OR.(VALOR.LT.2.AND.SINAL.EQ.3).OR.SINAL.EQ.1.
     .AND.(VALOR.LT.0.OR.VALOR.GT.31).OR.SINAL.EQ.4.AND.VALOR.GT.25.
     .OR.SINAL.EQ.5.AND.VALOR.GT.32) GOTO 1000
C
        RETURN
99      RETURN1
1000    CALL MENSAGE(DISPLAY,3,*99)
        END

    SUBROUTINE ERROMED(IA,IB,NOME1,NOME2,FLAG1,FLAG2,DISPLAY)
*****************************************************************
*     DETERMINA A MEDIA DAS DIFERENCAS ABSOLUTAS E O ERRO       *
*     MEDIO QUADRATICO ENTRE DUAS IMAGENS                       *
*****************************************************************
C
        IMPLICIT INTEGER*2 (C-Z)
        CHARACTER NOME1*14,NOME2*14
        REAL*4  MDA,EMQ
        DIMENSION IA(64,64),IB(64,64),XY(4)
C
C     APRESENTA AS DUAS IMAGENS
C
        CALL MOLDURA(DISPLAY)
C
        CALL IMAGEM(IA,NOME1,DISPLAY,2200,6500,0,FLAG1)
        CALL IMAGEM(IB,NOME2,DISPLAY,17200,6500,0,FLAG2)
C
C     CALCULAR A MEDIA DAS DIFERENCAS ABSOLUTAS
C
        MDA=0.
```

```
      DO 5 I=1,64
      DO 5 J=1,64
5     MDA=MDA+IABS(IA(I,J)-IB(I,J))
      MDA=MDA/4096.
C
C     CALCULAR O ERRO MEDIO QUADRATICO
C
      EMQ=0.
      DO 6 I=1,64
      DO 6 J=1,64
6     EMQ=EMQ+(IA(I,J)-IB(I,J))**2
      EMQ=EMQ/4096.
C
C     APRESENTAR VALORES NO VIDEO
C
      XY(1)=4000
      XY(3)=23800+XY(1)
      XY(2)=3500
      XY(4)=XY(2)+2200
      STATUS=VCLARY(DISPLAY,XY,1,1,1,4,3)
      STATUS=VSAPOS(DISPLAY,XY(1)+500,XY(4)-1000,XOUT,YOUT)
      STATUS=VATXTS(DISPLAY,23,'Media difer. absoluta: ',XOUT,YOUT)
      CALL DSPNUM(MDA,0,XOUT,YOUT,1,DISPLAY)
      STATUS=VSAPOS(DISPLAY,XY(1)+500,XY(2)+200,XOUT,YOUT)
      STATUS=VATXTS(DISPLAY,23,'Erro medio quadratico: ',XOUT,YOUT)
      CALL DSPNUM(EMQ,0,XOUT,YOUT,1,DISPLAY)
C
      CALL CRFIM(DISPLAY)
      RETURN
      END

      SUBROUTINE GERUIDO(DISPLAY)
*********************************************
*     CHAMA ROTINA QUE GERA RUIDOS DO TIPO
*     UNIFORME OU GAUSSIANO
*********************************************
C
      IMPLICIT INTEGER*2(C-Z)
      CHARACTER NOME*14,STRING,STRIND,SUBMENU*17,F(3)*35
      COMMON IA(64,64),IB(64,64),IC(64,64),NOME,/COMUNIC/STRING,STRIND
      DATA SUBMENU /'GERACAO DE RUIDOS'/
      DATA F(1) /'F1 Ruido com distribuicao uniforme'/
      DATA F(2) /'F2 Ruido com distribuicao gaussiana'/
      DATA F(3) /'F3 Display de imagens'/
C
10    STATUS=VCLRWK(DISPLAY)
      CALL MCURSOR(DISPLAY,SUBMENU,17)
C
      I=1
      DO 6 J=10,14,2
      STATUS=VCURAD(DISPLAY,J,10)
      STATUS=VCTXTS(DISPLAY,35,F(I))
6     I=I+1
C
      CALL FINAL(DISPLAY,1)
C
C     LER OPCAO DE ENTRADA
C
```

```
      STATUS=VRQCHC(DISPLAY,CHFIN,CHFIN)
      IF(CHFIN.EQ.1.AND.STATUS.GT.0) THEN
      CALL PISCA (10,10,F(1),5,DISPLAY,34,*10)
      GOTO 100
     ENDIF
C
      IF(CHFIN.EQ.2) THEN
      CALL PISCA (12,10,F(2),5,DISPLAY,35,*10)
      GOTO 100
     ENDIF
C
      IF(CHFIN.EQ.3) THEN
      CALL PISCA (14,10,F(3),0,DISPLAY,21,*10)
      CALL IMAGEDIS(DISPLAY,*10)
     ENDIF
      IF(CHFIN.EQ.9) CALL AUXILIO(DISPLAY,8)
C
90    IF(CHFIN.NE.10) GOTO 10
      RETURN
C
100   IF(STRING.EQ.'1') CALL RUIDO(IA,IC,DISPLAY,CHFIN,1)
      IF(STRING.EQ.'2') CALL RUIDO(IB,IC,DISPLAY,CHFIN,2)
      GOTO 90
      END

 SUBROUTINE RUIDO(IA,IC,DISPLAY,CHFIN,FLAG)
*******************************************
*     GERACAO DE RUIDO GAUSSIANO OU UNIFORME
*******************************************
C
      IMPLICIT INTEGER*2  (C-Q,S-Z)
      IMPLICIT REAL (A,R)
      REAL A(55),SEED,PI,PI2,CNTREAL
      DIMENSION IA(64,64),IC(64,64)
      COMMON /SIGMDP/CNTREAL
C
C
      DO 5 I=1,64
      DO 5 J=1,64
5     IC(I,J)=IA(I,J)
C
C
      PI= 4*ATAN(1.)
      RAV=0.
      RSD=1.
1     STATUS=VCURAD(DISPLAY,23,1)
      STATUS=VRVON(DISPLAY)
      STATUS=VEREOS(DISPLAY)
      STATUS=VCTXTS(DISPLAY,28,'Indique valor medio (F4.1): ')
      CALL LER12(DISPLAY,8,NADA,8,*1000)
      RAV=CNTREAL
C     READ(*,'(F5.1)',ERR=1000) RAV
      STATUS=VCURAD(DISPLAY,24,1)
      STATUS=VCTXTS(DISPLAY,30,'Indique desvio padrao (F4.1): ')
      CALL LER12(DISPLAY,8,NADA,8,*1000)
      RSD=CNTREAL
C     READ(*,'(F5.1)',ERR=1000) RSD
      SEED=0.3114159
      PI2=PI*2
      RMIN=2.**(-20)
```

```
      STATUS=VCURAD(DISPLAY,2,1)
      STATUS=VRVOFF(DISPLAY)
      STATUS=VCTXTS(DISPLAY,10,'AGUARDE...')
      CALL ZIN55(A,SEED,RMIN)
      DO 10 I=1,64
      DO 10 J=1,64
      IF(CHFIN.EQ.2) GO TO 50
      R1=RAND(A)
      R2=RAND(A)
      RZ=SQRT(-2.*ALOG(R1))*SIN(PI2*R2)
      IN=IFIX(RAV+RSD*R2+.5)
      GO TO 60
50    R=RAND(A)
      IN=IFIX(RAV+RSD*(R*2.-1)+.5)
60    IC(I,J)=IC(I,J)+IN
      IF(IC(I,J).GT.31) IC(I,J)=31
      IF(IC(I,J).LT.0)  IC(I,J)=0
10    CONTINUE
      STATUS=VCURAD(DISPLAY,2,1)
      STATUS=VEREOL(DISPLAY)
C
      CALL MENSINF(DISPLAY,2)
1000  RETURN
      END
C
C
      REAL FUNCTION RAND(A)
      REAL A(55)
      DATA L /1/
      RAND=A(L)
      L=L+1
      IF(L.GT.55) L=IRN55(A)
      RETURN
      END
C
C
      SUBROUTINE ZIN55(A,SEED,RMIN)
      REAL A(55)
      DATA H /0.5/
      A(55)=SEED
      RK=RMIN
      RJ=SEED
      DO 10 I=1,54
      II=MOD(I*21,55)
      A(II)=RK
      RK=RJ-RK
      IF(RK.LT.0) RK=(RK+H)+H
      RJ=A(II)
10    CONTINUE
      L=IRN55(A)
      L=IRN55(A)
      L=IRN55(A)
      RETURN
      END
C
C
      INTEGER FUNCTION IRN55(A)
      REAL A(55)
      DATA H /0.5/
```

```
      DO 10 I=1,24
      RJ=A(I)-A(I+31)
      IF(RJ.LT.0.) RJ=(RJ+H)+H
      A(I)=RJ
10    CONTINUE
      DO 20 I=25,55
      RJ=A(I)-A(I-24)
      IF(RJ.LT.0.) RJ=(RJ+H)+H
      A(I)=RJ
20    CONTINUE
      IRN55=1
      RETURN
      END
      SUBROUTINE ADICAO(IA,IB,IC,DISPLAY)
*****************************************************************
*     REALIZA A ADICAO DE DUAS IMAGENS MI1+MI2 E ARMAZENA EM MI3
*****************************************************************
C
      IMPLICIT INTEGER*2 (C-Z)
      DIMENSION IA(64,64),IB(64,64),IC(64,64)
C
      DO 5 I=1,64
      DO 5 J=1,64
      IADIT=IA(I,J)+IB(I,J)
      IF(IADIT.GT.31) IADIT=31
5     IC(I,J)=IADIT
C
      CALL MENSINF(DISPLAY,1)
      RETURN
      END
C
      SUBROUTINE SUBTRAC(IA,IB,IC,DISPLAY)
*****************************************************************
*     REALIZA A SUBTRACAO ENTRE DUAS IMAGENS E ARMAZENA O RESULTADO
*     EM MI3
*****************************************************************
C
      IMPLICIT INTEGER*2 (C-Z)
      DIMENSION IA(64,64),IB(64,64),IC(64,64)
C
      DO 5 I=1,64
      DO 5 J=1,64
      SUBT=IA(I,J)-IB(I,J)
      IF(SUBT.LT.0) SUBT=0
5     IC(I,J)=SUBT
C
      CALL MENSINF(DISPLAY,1)
      RETURN
      END
C
C
      SUBROUTINE DIFABS(IA,IB,IC,DISPLAY)
*************************************************
*     REALIZA A DIFERENCA ABSOLUTA ENTRE 2 IMAGENS
*************************************************
C
      IMPLICIT INTEGER*2 (C-Z)
      DIMENSION IA(64,64),IB(64,64),IC(64,64)
C
```

```
      DO 5 I=1,64
      DO 5 J=1,64
5     IC(I,J)=IABS(IA(I,J)-IB(I,J))
C
      CALL MENSINF(DISPLAY,1)
      RETURN
      END


    SUBROUTINE ZOOM(IA,IC,NOME,DISPLAY,DN)
*********************************************
*     REALIZA EFEITO ZOOM
*********************************************
C
      IMPLICIT INTEGER*2 (C-Z)
      DIMENSION XYB(4),IA(64,64),IC(64,64)
      CHARACTER NOME*14
      COMMON /PASSO1/PASSO,/WIDE/LINFER,LINSUP,COLINF,COLSUP
      DATA XIM,YIM /1600,6000/
C
5     CALL MOLDURA(DISPLAY)
C
C     DEFINE CARACTERISTICAS DA JANELA
C
      STATUS=VSFCOL(DISPLAY,0)
      STATUS=VSFINT(DISPLAY,0)
C
      CALL IMAGEM(IA,NOME,DISPLAY,XIM,YIM,0,DN)
      XYB(1)=XIM+64*210+50
      XYB(2)=YIM-200
      XYB(3)=XYB(1)+200
      XYB(4)=YIM+64*210+100
      STATUS=VCLARY(DISPLAY,XYB,1,1,1,4,0)
C
C     LER TIPO DE DESLOCAMENTO
C
      STATUS=VGTXTS(DISPLAY,2000,4000,12,'Janela 16*16')
      STATUS=VGTXTS(DISPLAY,2000,3000,13,'Deslocamento?')
      STATUS=VGTXTS(DISPLAY,2000,2000,16,'F1-Ponto a ponto')
      STATUS=VGTXTS(DISPLAY,2000,1000,24,'F2-Comprimento da janela')
C
6     STATUS=VRQCHC(DISPLAY,CHAIN,CHAIN)
      DESLOC=1
      IF(CHAIN.EQ.2) DESLOC=16
      STATUS=VGTXTS(DISPLAY,2000,3000,19,'Fator de ampliacao?')
      STATUS=VGTXTS(DISPLAY,2000,2000,16,'F1-Zoom de 2     ')
      STATUS=VGTXTS(DISPLAY,2000,1000,24,'F2-Zoom de 4            ')
7     STATUS=VRQCHC(DISPLAY,CHAIN,CHAIN)
      FATOR=2*210
      IF(CHAIN.EQ.2) FATOR=4*210
C
      STATUS=VGTXTS(DISPLAY,2000,3000,34,'Posicione a janela e tecle <EN
     .TER>')
      STATUS=VGTXTS(DISPLAY,2000,2000,22,'F1-Esquerda F2-Direita')
      STATUS=VGTXTS(DISPLAY,2000,1000,33,'F3-P/cima   F4-P/baixo F5-Te
     .rmina')
C
C     DEFINE AREA 16*16 DA JANELA
C
```

```
      XYB(1)=XIM
      XYB(2)=YIM+13440-16*210
      XYB(3)=XIM+16*210
      XYB(4)=YIM+13440
C
C     DELOCAR JANELA NA IMAGEM
C
      PASSO=0
10    CALL MOVE(XIM,YIM,16,XYB,DESLOC,2,DIR,DISPLAY)
      IF(DIR.EQ.5) GOTO 25
C
C     APRESENTAR EFEITO ZOOM DOS PONTOS DEFINIDOS POR XYB
C
      LINFER=65-(XYB(4)-YIM)/210 +.5
      LINSUP=LINFER+15
      COLSUP=(XYB(3)-XIM)/210 +.5
      COLINF=COLSUP-15
C
      CALL REDZOM(IA,IC,FATOR,DISPLAY)
      GOTO 10
C
25    CALL CRRET(DISPLAY,*5,2000,2000)
      END
C
      SUBROUTINE CRRET(DISPLAY,*,XPOS,YPOS)
*******************************************************
*     APRESENTA OPCOES DE COPIA,REPETE E RETORNA
*******************************************************
      IMPLICIT INTEGER*2 (C-Z)
      DIMENSION LIMP(4)
C
      LIMP(1)=XPOS-500
      LIMP(2)=YPOS-1000
      LIMP(3)=30000
      LIMP(4)=YPOS+1800
      STATUS=VCLARY(DISPLAY,LIMP,1,1,1,4,0)
C
30    STATUS=VGTXTS(DISPLAY,XPOS,YPOS,29,'F1-Copia F2-Repete F3-Retorna'
     .)
      STATUS=VRQCHC(DISPLAY,CHAIN,CHAIN)
      IF(CHAIN.EQ.1.AND.STATUS.GT.0) THEN
      STATUS=VGTXTS(DISPLAY,2000,2000,29,'
     .)
      STATUS=VHDCPY(DISPLAY)
     ENDIF
      IF(CHAIN.EQ.2) RETURN 1
      IF(CHAIN.NE.3) GOTO 30
      STATUS=VENCUR(DISPLAY)
      RETURN
      END
C
C
      SUBROUTINE REDZOM(IA,IC,FATORI,DISPLAY)
*********************************************************************
*     APRESENTA EFEITO ZOOM OU REDUCAO DA IMAGEM DE ACORDO COM FATOR
*********************************************************************
C
      IMPLICIT INTEGER*2 (C-Z)
      CHARACTER SN
```

```
      DIMENSION IA(64,64),IC(64,64),XYZ(4)
      COMMON /WIDE/LINFER,LINSUP,COLINF,COLSUP
C
      PRIM=0
      FATOR=FATORZ
      XIM=17500
      YIM=18600
1     IF(FATOR.EQ.105) XIM=20000
      IF(FATOR.EQ.52) XIM=22000
C
      IF(FATOR.EQ.420)
     - STATUS=VGTXTS(DISPLAY,XIM,YIM+1300,9,'Zoom de 2')
      IF(FATOR.EQ.840)
     - STATUS=VGTXTS(DISPLAY,XIM+2500,YIM+1600,9,'Zoom de 4')
C
      IF(FATOR.EQ.105) STATUS=VGTXTS(DISPLAY,XIM-1500,YIM+800,12,'Reduca
     .o de 2')
      IF(FATOR.EQ.52) STATUS=VGTXTS(DISPLAY,XIM-3500,YIM+800,12,'Reducao
     . de 4')
C
      XYZ(2)=YIM+FATOR
      DO 5 I=LINFER,LINSUP
      XYZ(1)=XIM-FATOR
      XYZ(2)=XYZ(2)-FATOR
      XYZ(4)=XYZ(2)+FATOR
      DO 5 J=COLINF,COLSUP
      XYZ(1)=XYZ(1)+FATOR
      XYZ(3)=XYZ(1)+FATOR
      CALL IMAGEP(IA(I,J),XYZ,DISPLAY)
5     CONTINUE
C
      IF(FATOR.GT.105) THEN
C
C     ARMAZENAR ZOOM DA IMAGEM EM MI3
C
      DO 2 I=1,64
      DO 2 J=1,64
2     IC(I,J)=0
C
      DELTA=FATOR/210
      K=1-DELTA
      L=K
      DO 6 I=LINFER,LINSUP
      K=K+DELTA
      DO 7 J=COLINF,COLSUP
      L=L+DELTA
      KSUP=K+(DELTA-1)
      LSUP=L+(DELTA-1)
      DO 8 X=K,KSUP
      DO 8 Y=L,LSUP
8     IC(X,Y)=IA(I,J)
7     CONTINUE
      L=1-DELTA
6     CONTINUE
      RETURN
     ENDIF
      STATUS=VGTXTS(DISPLAY,2000,3000,17,'                ')
      STATUS=VGTXTS(DISPLAY,2000,1000,16,'                ')
      STATUS=VSAPOS(DISPLAY,2000,2000,XOUT,YOUT)
```

```
        IF(FATOR.EQ.105.AND.PRIM.EQ.0) THEN
        YIM=YIM-9000
        STATUS=VATXTS(DISPLAY,18,'Reducao de 4(S/N)?',XOUT,YOUT)
        FATOR=52
        GOTO 50
      ENDIF
        IF(FATOR.EQ.52.AND.PRIM.EQ.0) THEN
        YIM=YIM-6000
        STATUS=VATXTS(DISPLAY,18,'Reducao de 2(S/N)?',XOUT,YOUT)
        FATOR=105
        GOTO 50
      ENDIF
C
        RETURN
50      PRIM=1
        SN=' '
        STATUS=VSMSTR(DISPLAY,1,0,XYZ,SN)
        IF(SN.NE.'N'.AND.SN.NE.'n'.AND.SN.NE.'S'.AND.SN.NE.'s') GOTO 50
        STATUS=VGTXTS(DISPLAY,XOUT,YOUT+210,1,SN)
        IF(SN.EQ.'S'.OR.SN.EQ.'s') GOTO 1
        RETURN
        END

       SUBROUTINE REDUCAO(IA,NOME,DISPLAY,DN)
*************************************************************
*     FAZ A REDUCAO DA IMAGEM APRESENTADA NO VIDEO
*     DE ACORDO COM FATOR DE REDUCAO (2 OU 4)
*************************************************************
C
        IMPLICIT INTEGER*2 (C-Z)
        DIMENSION IA(64,64),LIMP(4)
        COMMON /WIDE/LINFER,LINSUP,COLINF,COLSUP
        DATA XIM,YIM,LIMP /1600,6000,17000,5000,30000,21000/
        LINFER=1
        COLINF=LINFER
        LINSUP=64
        COLSUP=LINSUP
C
        CALL MOLDURA(DISPLAY)
C
5       CALL IMAGEM(IA,NOME,DISPLAY,XIM,YIM,0,DN)
C
C     LIMPAR AREA
C
        STATUS=VCLARY(DISPLAY,LIMP,1,1,1,4,0)
C
C     LER FATOR DE REDUCAO
C
        STATUS=VGTXTS(DISPLAY,2000,3000,17,'Fator de reducao?')
        STATUS=VGTXTS(DISPLAY,2000,2000,29,'F1-Reducao de 2
     .)
        STATUS=VGTXTS(DISPLAY,2000,1000,15,'F2-Reducao de 4')
        STATUS=VRQCHC(DISPLAY,CHAIN,CHAIN)
        FATOR=210/2
        IF(CHAIN.EQ.2) FATOR=210/4
C
C     REALIZAR REDUCAO
C
```

```
      CALL REDZOM(IA,IA,FATOR,DISPLAY)
C
      CALL CRRET(DISPLAY,#5,2000,2000)
      END

      SUBROUTINE BINARIA (IA,IC,DISPLAY,FLAG)
*************************************************************
*     VARAIACAO DA TRANSFORMACAO GRADIENTE QUE TEM COMO
*     RESULTADO UMA IMAGEM BINARIA
*************************************************************
C
      IMPLICIT INTEGER*2 (C-Z)
      CHARACTER TEXT*2
      DIMENSION IA(64,64),IC(64,64),XY(4)
      DATA PXIM,PYIM /17500,3400/
C
C     LER VALORES PARA LIMIAR,BORDA E PLANO DE FUNDO
C
      LB=7
      CALL ENTRE(DISPLAY,LIMIAR,LG,LB,#100)
      STATUS=VGTXTS(DISPLAY,7900,20810,24,'Imagem gradiente binaria')
      STATUS=VSAPOS(DISPLAY,5500,19500,I,J)
      STATUS=VATXTS(DISPLAY,7,'Limiar=',I,J)
      WRITE(TEXT,'(I2)') LIMIAR
      STATUS=VATXTS(DISPLAY,2,TEXT,I,J)
      STATUS=VATXTS(DISPLAY,8,', Borda=',I,J)
      WRITE(TEXT,'(I2)') LG
      STATUS=VATXTS(DISPLAY,2,TEXT,I,J)
      STATUS=VATXTS(DISPLAY,8,', Fundo=',I,J)
      WRITE(TEXT,'(I2)') LB
      STATUS=VATXTS(DISPLAY,2,TEXT,I,J)
C
      CALL IMAGEM(IA,'Original       ',DISPLAY,2000,3600,0,FLAG)
      STATUS=VGTXTS(DISPLAY,PXIM+2000,PYIM+14400,10,'Processada')
C
      XY(2)=PYIM+13440+210
      DO 5 X=1,63
      XY(1)=PXIM-210
      XY(2)=XY(2)-210
      XY(4)=XY(2)+210
      DO 5 Y=1,63
      XY(1)=XY(1)+210
      XY(3)=XY(1)+210
      GRD=IMAGRAD(X,Y,IA)
      IC(X,Y)=LB
      IF(GRD.GE.LIMIAR) IC(X,Y)=LG
C
C     APRESENTAR RESULTADO NA TELA
C
      CALL IMAGEP(IC(X,Y),XY,DISPLAY)
5     CONTINUE
C
C     COMPLETAR MEMORIA DE IMAGEM PARA GRADIENTE
C
      CALL COMPGRAD(PXIM,PYIM,XY,IA,IC,DISPLAY,FLAG)
C
100   RETURN
      END
```

```
      SUBROUTINE SIGMA(IA,IC,DISPLAY,FLAG)
*****************************************************************
*     REALIZA A FILTRAGEM ESPACIAL UTILIZANDO O FILTRO SIGMA
*****************************************************************
C
      IMPLICIT INTEGER*2 (C-Z)
      REAL VREAL
      CHARACTER TAMANHO*10,TEXT*2,TEXT1
      DIMENSION IA(64,64),IC(64,64),VETOR(25),XY(4)
      COMMON/SIGMDP/VREAL,/PRASOK/TAMANHO
      DATA PXIM,PYIM /17500,3500/
C
      CALL ENTRE(DISPLAY,E,NADA,3,*100)
C
      STATUS=VSAPOS(DISPLAY,4000,2080,I,J)
      STATUS=VATXTS(DISPLAY,16,'Filtro Sigma(DP=',I,J)
      IREAL1=VREAL
      WRITE(TEXT,'(I2)') IREAL1
      STATUS=VATXTS(DISPLAY,2,TEXT,I,J)
      STATUS=VATXTS(DISPLAY,1,'.',I,J)
      IREAL2=NINT((VREAL-IREAL1)*10.)
      WRITE(TEXT1,'(I1)') IREAL2
      STATUS=VATXTS(DISPLAY,1,TEXT1,I,J)
      STATUS=VATXTS(DISPLAY,2,'):',I,J)
      STATUS=VATXTS(DISPLAY,10,TAMANHO,I,J)
C
      YPOS=3900
      IF(E.EQ.5) YPOS=4110
      CALL IMAGEM(IA,'Original       ',DISPLAY,2000,YPOS,0,FLAG)
      STATUS=VGTXTS(DISPLAY,PXIM+2000,PYIM+14700,10,'Processada')
C
      VREAL=2*VREAL
      XY(2)=PYIM+13440+210
      PONTOIJ=1
      DIM=2
      IJFIM=62
      IF(E.NE.5) GOTO 2
      IJFIM=60
      DIM=4
      PONTOIJ=2
 2    DO 5 I=1,IJFIM
      XY(1)=PXIM-210
      XY(2)=XY(2)-210
      XY(4)=XY(2)+210
      DO 5 J=1,IJFIM
      XY(1)=XY(1)+210
      XY(3)=XY(1)+210
      IW=I+DIM
      JW=J+DIM
      K=1
      DO 10 IF=I,IW
      DO 10 JF=J,JW
      VETOR(K)=IA(IF,JF)
      K=K+1
 10   CONTINUE
      PONTOP=SIGMA1(VETOR,E**2,VREAL)
C
C     ARMAZENAR RESULTADO NO PONTO CENTRAL DA JANELA
C
```

```
      IC(I+PONTOIJ,J+PONTOIJ)=PONTOP
      CALL IMAGEP(PONTOP,XY,DISPLAY)
5     CONTINUE
C
      E=E/2
      CALL COMPLET(IC,IA,E,DISPLAY,FLAG)
C
100   RETURN
      END
C
C
      INTEGER*2 FUNCTION SIGMA1(VETOR,E2,FAIXA)
*************************************************************
      IMPLICIT INTEGER*2 (C-Z)
      REAL FAIXA
      DIMENSION VETOR(E2),FAIXEL(25)
C
      DO 1 I=1,E2
1     FAIXEL(I)=0
C
      MEIO=VETOR(E2/2+1)
      FAIXA1=MEIO-0.31*FAIXA+.5
      FAIXA2=MEIO+0.31*FAIXA+.5
      CONT=0
C
C     VERIFICAR QUAIS ELEMENTOS DA JANELA SE ENCONTRAM DENTRO DA FAIXA
C     ESTABELECIDA:
C
      DO 5 I=1,E2
      IF(VETOR(I).LT.FAIXA1.OR.VETOR(I).GT.FAIXA2) GOTO 5
      FAIXEL(I)=VETOR(I)
      CONT=CONT+1
5     CONTINUE
C
C    CALCULAR A MEDIA DOS ELEMENTOS DENTRO DA FAIXA ESTABELECIDA:
C
      SIGMA1=0
      DO 10 I=1,E2
10    SIGMA1=SIGMA1+FAIXEL(I)
      SIGMA1=SIGMA1/CONT+.5
C
      RETURN
      END

    SUBROUTINE KVIZFIL(IA,IC,DISPLAY,FLAG)
*****************************************************************
*     EXECUTA O FILTRO DA MEDIA COM OS K-VIZINHOS MAIS PROXIMOS
*     (JANELAS 3*3 OU 5*5)
*****************************************************************
C
      IMPLICIT INTEGER*2 (C-Z)
      CHARACTER TAMANHO*10,TEXT*2
      DIMENSION IA(64,64),IC(64,64),VETOR(25),XY(4)
      COMMON/PRASOK/TAMANHO
      DATA PXIM,PYIM /17500,3500/
C
      CALL ENTRE(DISPLAY,E,KVAL,2,*100)
      STATUS=VSAPOS(DISPLAY,5000,20810,I,J)
      STATUS=VATXTS(DISPLAY,14,'Filtro MKVP(K=',I,J)
      WRITE(TEXT,'(I2)') KVAL
      STATUS=VATXTS(DISPLAY,2,TEXT,I,J)
      STATUS=VATXTS(DISPLAY,2,'):',I,J)
      STATUS=VATXTS(DISPLAY,10,TAMANHO,I,J)
```

```
1       STATUS=VEREOS(DISPLAY)
        STATUS=VCTXTS(DISPLAY,17,'Tamanho da janela?')
        STATUS=VCURAD(DISPLAY,24,1)
        STATUS=VCTXTS(DISPLAY,14,'F1-3*3  F2-5*5')
50      OPCAO=' '
        STATUS=VRQSTR(DISPLAY,1,0,XYEND,OPCAO)
        EINT=ICHAR(OPCAO)
        IF(EINT.EQ.27) GOTO 100
        IF(EINT.LT.59.OR.EINT.GT.60) GOTO 50
        CHAIN=EINT-58
        STATUS=VCURAD(DISPLAY,23,1)
        STATUS=VEREOS(DISPLAY)
        IF(CHAIN.EQ.2) THEN
        TAMANHO='Janela 5*5'
        STATUS=VCTXTS(DISPLAY,22,'JANELA 5*5            ')
        STATUS=VCURAD(DISPLAY,24,1)
        STATUS=VRVON(DISPLAY)
        STATUS=VEREOS(DISPLAY)
        E=5
        IF(INDICE.EQ.1)
     .STATUS=VCTXTS(DISPLAY,34,'Escolha a ordem do pixel [1..25]: ')
        IF(INDICE.EQ.2)
     .STATUS=VCTXTS(DISPLAY,38,'Indique valor dos K-vizinhos [1..25]: ')
        IF(INDICE.EQ.3) GOTO 80
        CALL LER12(DISPLAY,4,KVAL,4,*100)
        GOTO 10
       ENDIF
C
        TAMANHO='Janela 3*3'
        STATUS=VCTXTS(DISPLAY,22,'JANELA 3*3            ')
        STATUS=VCURAD(DISPLAY,24,1)
        STATUS=VRVON(DISPLAY)
        STATUS=VEREOS(DISPLAY)
        E=3
        IF(INDICE.EQ.1)
     .STATUS=VCTXTS(DISPLAY,33,'Escolha a ordem do pixel [1..9]: ')
        IF(INDICE.EQ.2)
     .STATUS=VCTXTS(DISPLAY,37,'Indique valor dos K-vizinhos [1..9]: ')
        IF(INDICE.EQ.3) GOTO 80
        CALL LER12(DISPLAY,4,KVAL,0,*100)
        GOTO 10
C
4       STATUS=VRVON(DISPLAY)
        STATUS=VEREOS(DISPLAY)
        STATUS=VCTXTS(DISPLAY,33,'Indique valor do limiar [0..31]: ')
        CALL LER12(DISPLAY,4,E,1,*100)
        STATUS=VCURAD(DISPLAY,24,1)
        IF(INDICE.EQ.5.OR.INDICE.EQ.7) THEN
        STATUS=VCTXTS(DISPLAY,36,'Indique valor para a borda [0..31]: ')
        CALL LER12(DISPLAY,4,KVAL,1,*100)
        XP=25
       ENDIF
C
        STATUS=VCURAD(DISPLAY,XP,1)
        IF(INDICE.EQ.6.OR.INDICE.EQ.7) THEN
        STATUS=VCTXTS(DISPLAY,40,'Indique valor do plano de fundo [0..31]:
     .')
        IF(INDICE.NE.6) GOTO 5
```

```
        CALL LER12(DISPLAY,4,KVAL,1,*100)
        GOTO 10
5       CALL LER12(DISPLAY,4,SIMAL,1,*100)
      ENDIF
C
10      CALL MOLDURA(DISPLAY)
C
        STATUS=VRVOFF(DISPLAY)
        RETURN
100     STATUS=VRVOFF(DISPLAY)
        RETURN 1
C
80      STATUS=VCTXTS(DISPLAY,46,'Indique desvio padrao do ruido [0.0 .. 5
     .0.0]: ')
        CALL LER12 (DISPLAY,8,NADA,7,*100)
        GOTO 10
C
        END

        SUBROUTINE LEITURA(DISPLAY,FLAG,*)
C*****************************************************************
C     APRESENTA MENSAGENS PARA A ENTRADA DE DADOS DO USUARIO
C*****************************************************************
C
        IMPLICIT INTEGER*2 (C-Z)
        DIMENSION EXE(2)
        CHARACTER LETRA(23),MIMG,DIMG,NOME*14,LER(15),MSG1*61,MSG2*56,
     .MSG3*63,MSG4*44,MSG5*40,MSG6*63,MSG7*50,MSG10*49,MSG12*52
        COMMON IA(64,64),IB(64,64),IC(64,64),NOME,/COMUNIC/MIMG,DIMG,
     .          /PRALER/LETRA
        EQUIVALENCE(NOME,LER(1))
        DATA MSG1 /'Digite: NOME DO ARQUIVO,memoria de imagem (MI1,MI2 ou
     .MI3)_ '/
        DATA MSG2 /'Digite: NOME DO ARQUIVO,memoria de imagem (MI1 ou MI2)
     ._ '/
        DATA MSG3 /'Digite: NOME DO ARQUIVO 1,memoria de imagem 1(MI1,MI2
     .ou MI3)_ '/
        DATA MSG4 /'Digite: Memoria de Imagem (MI1,MI2 ou MI3)_ '/
        DATA MSG5 /'Digite: Memoria de imagem (MI1 ou MI2)_ '/
        DATA MSG6 /'Digite: NOME DO ARQUIVO 2,memoria de imagem 2(MI1,MI2
     .ou MI3)_ '/
        DATA MSG7 /'Digite memoria de imagem fonte: (MI1,MI2 ou MI3)_ '/
        DATA MSG10/'Digite minuendo: memoria de imagem (MI1 ou MI2)_ '/
        DATA MSG12/'Digite memoria de imagem destino: (MI1,MI2 ou MI3)_ '/
C
        PRESS=ALTF(DISPLAY)
C
C     INFORMAR SOBRE ENTRADAS ANTERIORES,NO CASO DE DUAS ENTRADAS SUCESSIVAS
C     EM LER.FOR (FLAG=32)
C
        STATUS=VCURAD(DISPLAY,24,9)
        IF(FLAG.EQ.12) THEN
        STATUS=VCTXTS(DISPLAY,27,'MEMORIA DE IMAGEM FONTE: MI')
        GOTO 21
      ENDIF
        IF(FLAG.EQ.32) THEN
        STATUS=VCTXTS(DISPLAY,19,'NOME DO ARQUIVO 1: ')
        STATUS=VCTXTS(DISPLAY,14,NOME)
        STATUS=VCTXTS(DISPLAY,24,';memoria de imagem 1: MI')
        GOTO 21
      ENDIF
```

```
      GOTO 20
21    STATUS=VCTXTS(DISPLAY,1,MIMG)
20    DO 1 I=1,23
1     LETRA(I)=' '
      NOME=' '
      MIMG=' '
      MINT=0
C
      STATUS=VCURAD(DISPLAY,25,1)
      STATUS=VRVON(DISPLAY)
      IF(FLAG.EQ.1.OR.FLAG.EQ.15) STATUS=VCTXTS(DISPLAY,61,MSG1)
      IF(FLAG.EQ.2) STATUS=VCTXTS(DISPLAY,56,MSG2)
      IF(FLAG.EQ.3) STATUS=VCTXTS(DISPLAY,61,MSG1)
      IF(FLAG.EQ.31) STATUS=VCTXTS(DISPLAY,63,MSG3)
      IF(FLAG.EQ.32) STATUS=VCTXTS(DISPLAY,63,MSG6)
      IF(FLAG.EQ.4) STATUS=VCTXTS(DISPLAY,44,MSG4)
      IF(FLAG.EQ.5) STATUS=VCTXTS(DISPLAY,40,MSG5)
      IF(FLAG.EQ.6) STATUS=VCTXTS(DISPLAY,50,MSG7)
      IF(FLAG.EQ.8) STATUS=VCTXTS(DISPLAY,50,MSG8)
      IF(FLAG.EQ.9) STATUS=VCTXTS(DISPLAY,50,MSG9)
      IF(FLAG.EQ.10) STATUS=VCTXTS(DISPLAY,49,MSG10)
      IF(FLAG.EQ.11) STATUS=VCTXTS(DISPLAY,51,MSG11)
      IF(FLAG.EQ.12) STATUS=VCTXTS(DISPLAY,52,MSG12)
     STATUS=VEREOL(DISPLAY)
C
C
      CALL LER12(DISPLAY,24,VALOR,NADA,*100)
C
3      STATUS=VRVOFF(DISPLAY)
      DO 15 I=1,15
      IF(LETRA(I).EQ.',') GOTO 4
      LER(I)=LETRA(I)
15    CONTINUE
C
C    CASOS DE DEFAULT
C
      IF(FLAG.NE.15) GOTO 16
      MIMG='1'
      RETURN
C
16    IF(NOME.EQ.'mi1'.OR.NOME.EQ.'MI1'.OR.NOME.EQ.'mi2'.OR.NOME.EQ.
     .'MI2'.OR.NOME.EQ.'mi3'.OR.NOME.EQ.'MI3') THEN
      MIMG=LER(3)
      NOME=' '
      RETURN
      ELSE
      IF(((FLAG.GE.4.AND.FLAG.LE.6).OR.(FLAG.GE.8.AND.FLAG.LE.12)).AND.
     .NOME.NE.' ') GOTO 90
      MIMG='1'
      RETURN
     ENDIF
C
4     MIMG=LETRA(I+3)
      READ(MIMG,'(I1)',ERR=90) MINT
c     write(*,*) 'mimg= ',mimg,' diag= ',diag,' mint= ',mint,'dint=',dint
c     .,' FLAG= ',flag
c     pause
      IF(((FLAG.EQ.2.OR.FLAG.EQ.5.OR.(FLAG.GE.8.AND.FLAG.LE.11)).AND.
     .  (MINT.GT.2)).OR.MINT.GT.3.OR.MINT.LT.1) GOTO 90
C
```

```
      YPOS=3900
      IF(E.EQ.5) YPOS=4110
      CALL IMAGEM(IA,'Original       ',DISPLAY,2000,YPOS,0,FLAG)
      STATUS=VGTXTS(DISPLAY,PXIM+2000,PYIM+14700,10,'Processada')
C
      IY(2)=PYIM+13440+210
      PONTOIJ=1
      DIM=2
      IJFIM=62
      IF(E.NE.5) GOTO 2
      IJFIM=60
      DIM=4
      PONTOIJ=2
2     DO 5 I=1,IJFIM
      IY(1)=PXIM-210
      IY(2)=IY(2)-210
      IY(4)=IY(2)+210
      DO 5 J=1,IJFIM
      IY(1)=IY(1)+210
      IY(3)=IY(1)+210
      IW=I+DIM
      JW=J+DIM
      K=1
      DO 10 IF=I,IW
      DO 10 JF=J,JW
      VETOR(K)=IA(IF,JF)
      K=K+1
10    CONTINUE
      PONTOP=MEDIAK(VETOR,E**2,KVAL)
C
C     ARMAZENAR VALOR DA MEDIA DOS K-VIZINHOS NO PONTO CENTRAL DA JANELA
C
      IC(I+PONTOIJ,J+PONTOIJ)=PONTOP
      CALL IMAGEP(PONTOP,IY,DISPLAY)
C
5     CONTINUE
C
      E=E/2
      CALL COMPLET (IC,IA,E,DISPLAY,FLAG)
C
100   RETURN
      END
C
C
      INTEGER*2 FUNCTION MEDIAK(VETOR,E2,KVAL)
      IMPLICIT INTEGER*2 (C-Z)
      DIMENSION VETOR(*),ELEM(25),DIFERE(25),RESERV(25)
      COMMON/ORD/ORDEM(25)
C
      MEIO=E2/2 +1
      CENTRO=VETOR(MEIO)
C
C     CALCULAR A DIFERENCA ENTRE O PONTO CENTRAL E OS DEMAIS PIXELS DA
C     JANELA
C
      DO 10 I=1,E2
      DIFERE(I)=ABS(CENTRO-VETOR(I))
10    RESERV(I)=DIFERE(I)
C
```

```
C     COLOCAR O VETOR DIFERE EM ORDEM CRESCENTE. O RESULTADO E ARMAZENADO
C     NO VETOR ORDEM
C
        FUNCT=CRECNT(RESERV,E2,1)
C
C     SEPARAR OS K-ELEMENTOS MENORES DE DIFERE E ARMAZENAR EM ELEM(I)
C
        SINAL=0
        K=1
        DO 15 I=1,KVAL
        IF(ORDEM(K).EQ.DIFERE(MEIO).AND.SINAL.EQ.0) THEN
        if(kval.ne.9.and.kval.ne.25)    K=K+1
        SINAL=1
      ENDIF
        ELEM(I)=ORDEM(K)
15      K=K+1
C
C     SEPARAR OS K-VIZINHOS MAIS PROXIMOS DA JANELA
C
        DO 20 I=1,KVAL
        DO 20 J=1,E2
        IF(J.EQ.MEIO.and.(kval.ne.9.and.kval.ne.25)) GOTO 20
        IF(ELEM(I).eq.DIFERE(J)) then
        ELEM(I)=VETOR(J)
        DIFERE(J)=32
      endif
20      CONTINUE

C
C     CALCULAR A MEDIA DOS K-VIZINHOS MAIS PROXIMOS AO PONTO CENTRAL DA
C     JANELA
C
        MEDIAK=0
        DO 30 I=1,KVAL
30      MEDIAK=MEDIAK+ELEM(I)
        MEDIAK=MEDIAK/KVAL +.5
C
        END


        SUBROUTINE ENTRE (DISPLAY,E,KVAL,SINAL,*)
*****************************************************************
*     REALIZA A LEITURA DOS DADOS DE ENTRADA PARA OS FILTROS SIGMA,
*     K-VIZINHOS,ORDEM E GRADIENTES.ESPECIFICA AS JANELAS 3*3 OU 5*5
*****************************************************************
C
        IMPLICIT INTEGER*2 (C-Z)
        CHARACTER OPCAO,TAMANHO*10
        REAL VREAL
        DIMENSION XYEND(2)
        COMMON /SIGMDP/VREAL,/PRASOK/TAMANHO
C
        STATUS=VCURAD(DISPLAY,23,1)
        XP=24
C
        INDICE=SINAL
2       GOTO(1,1,1,4,4,4,4) INDICE
C
```

```
      RETURN
90    CALL MENSAGE(DISPLAY,1,*100)
100   RETURN 1
      END
C
C
      INTEGER*2 FUNCTION ALTF(DISPLAY)
*****************************************
*     MOSTRA OPCAO ALT F (ABANDONA) NA TELA
*****************************************
C
      IMPLICIT INTEGER*2 (C-Z)
      COMMON  /POSICAO/XPOS,YPOS
C
      STATUS=VCRCOL(DISPLAY,1,0,I,J)
      STATUS=VCURAD(DISPLAY,XPOS,YPOS)
      STATUS=VCTXTS(DISPLAY,26,'<ESC> Abandona            ')
      ALTF=VCRCOL(DISPLAY,1,4,I,J)
      END
      SUBROUTINE ES1 (IT,DISPLAY)
C
C************************************************
C              SUB-ROTINA ES
C    FUNCAO: CARREGA,DESCARREGA IMAGENS
C************************************************
C
      IMPLICIT INTEGER*2 (C-Z)
      CHARACTER NOME*14,STRING,STRIND
      LOGICAL EX
      COMMON IA(64,64),IB(64,64),IC(64,64),NOME,/COMUNIC/STRING,STRIND
     .       /SINAL/INDICA,INDICB
      IF(IT.EQ.-1) GOTO 26
C
      INQUIRE(FILE=NOME,EXIST=EX)
      IF(.NOT.EX.OR.NOME.EQ.' ') CALL MENSAGE(DISPLAY,2,*1000)
      STATUS=VCURAD(DISPLAY,2,2)
      IF(STRING.EQ.'1') THEN
      INDICA=0
      CALL CARREG (*1000,IA,NOME,STRING,DISPLAY)
     ENDIF
      IF(STRING.EQ.'2') THEN
      INDICB=0
      CALL CARREG (*1000,IB,NOME,STRING,DISPLAY)
     ENDIF
      CALL MENSAGE(DISPLAY,1,*1000)
C
26    IF(NOME.EQ.' ') CALL MENSAGE(DISPLAY,2,*1000)
      STATUS=VCURAD(DISPLAY,2,2)
      IF(STRING.EQ.'1') CALL ARMAZ (*1000,IA,NOME,STRING,DISPLAY)
      IF(STRING.EQ.'2') CALL ARMAZ (*1000,IB,NOME,STRING,DISPLAY)
      IF(STRING.EQ.'3') CALL ARMAZ (*1000,IC,NOME,STRING,DISPLAY)
      CALL MENSAGE(DISPLAY,1,*1000)
C
1000  RETURN
      END
C
      SUBROUTINE ARMAZ (*,IA,NOME,STRING,DISPLAY)
      IMPLICIT INTEGER*2 (C-Z)
      DIMENSION IA(64,64)
```

```
      CHARACTER NOME*14,STRING
      OPEN(1,FILE=NOME,ACCESS='DIRECT',RECL=128,FORM='FORMATTED',
     1 STATUS='NEW')
      STATUS=VCTXTS(DISPLAY,24,'AGUARDE...ARMAZENANDO MI')
      STATUS=VCTXTS(DISPLAY,1,STRING)
      STATUS=VCTXTS(DISPLAY,13,' NO ARQUIVO: ')
      STATUS=VCTXTS(DISPLAY,14,NOME)
      DO 30 I=1,64
      WRITE(1,3,REC=I)(IA(I,J),J=1,64)
30    CONTINUE
      CLOSE(1)
3     FORMAT(64I2)
      RETURN 1
      END
C
      SUBROUTINE CARRES (*,IA,NOME,STRING,DISPLAY)
      IMPLICIT INTEGER*2 (C-Z)
      CHARACTER NOME*14,STRING
      DIMENSION IA(64,64)
      OPEN(1,FILE=NOME,ACCESS='DIRECT',FORM='FORMATTED',RECL=128)
      STATUS=VCTXTS(DISPLAY,24,'AGUARDE...  CARREGANDO: ')
      STATUS=VCTXTS(DISPLAY,14,NOME)
      STATUS=VCTXTS(DISPLAY,6,' EM MI')
      STATUS=VCTXTS(DISPLAY,1,STRING)
      DO 20 I=1,64
      READ(1,3,REC=I)(IA(I,J),J=1,64)
3     FORMAT(64I2)
20    CONTINUE
      CLOSE(1)
      RETURN 1
      END

      SUBROUTINE GERADOR(IA,DISPLAY,CHAIN,FLAG)
*************************************************
*     GERA ALGUMAS FIGURAS E ARMAZENA NA MEMORIA
*     DE IMAGEM ESPECIFICADA
*************************************************
C
      IMPLICIT INTEGER*2 (C-Z)
      DIMENSION IA(64,64),XY(4)
      COMMON/SINAL/INDICA,INDICB,/SINAL1/INDICC
C
      IF(FLAG.EQ.1) INDICA=0
      IF(FLAG.EQ.2) INDICB=0
      IF(FLAG.EQ.3) INDICC=0
C
      XP=20
      STATUS=VRVON(DISPLAY)
      IF(CHAIN.EQ.1.OR.CHAIN.EQ.2) THEN
C
      STATUS=VCURAD(DISPLAY,XP,1)
      STATUS=VEREOS(DISPLAY)
      STATUS=VCTXTS(DISPLAY,54,'Indique nivel de cinza p/ interior da fi
     .gura [0..31]: ')
      CALL LER12(DISPLAY,4,CI,1,*90)
      STATUS=VCURAD(DISPLAY,XP+1,1)
      STATUS=VCTXTS(DISPLAY,54,'Indique nivel de cinza p/ exterior da fi
     .gura [0..31]: ')
      CALL LER12(DISPLAY,4,CE,1,*90)
C
```

```
      GOTO (10,20) CHAIN
10    STATUS=VCURAD(DISPLAY,XP+2,1)
      STATUS=VCTXTS(DISPLAY,22,'Indique raio [1..32]: ')
      CALL LER12(DISPLAY,4,RAIO,5,*90)
      STATUS=VCURAD(DISPLAY,XP+3,1)
      STATUS=VCTXTS(DISPLAY,40,'Indique coordenada X do centro [1..64]:
     .')
      CALL LER12(DISPLAY,4,X,2,*90)
      STATUS=VCURAD(DISPLAY,XP+4,1)
      STATUS=VCTXTS(DISPLAY,40,'Indique coordenada Y do centro [1..64]:
     .')
      CALL LER12(DISPLAY,4,Y,2,*90)
      STATUS=VCURAD(DISPLAY,2,1)
      STATUS=VRVOFF(DISPLAY)
      STATUS=VCTXTS(DISPLAY,11,'AGUARDE... ')
      STATUS=VRVON(DISPLAY)
C
C     ARMAZENAR CIRCULO NA MEMORIA DE IMAGEM
C
      DO 11 I=1,64
      DO 11 J=1,64
      IA(I,J)=CE
      IF(SQRT(REAL((X-1)**2+(Y-J)**2)).LE.REAL(RAIO)) IA(I,J)=CI
11    CONTINUE
      GOTO 91
C
C     GERAR RETANGULO
C
20    STATUS=VCURAD(DISPLAY,XP+2,1)
      STATUS=VCTXTS(DISPLAY,59,'Indique coordenada X do vertice superior
     . esquerdo [1..64]: ')
      CALL LER12(DISPLAY,4,XV,2,*90)
      STATUS=VCURAD(DISPLAY,XP+3,1)
      STATUS=VCTXTS(DISPLAY,59,'Indique coordenada Y do vertice superior
     . esquerdo [1..64]: ')
      CALL LER12(DISPLAY,4,YV,2,*90)
      STATUS=VCURAD(DISPLAY,XP+4,1)
      STATUS=VCTXTS(DISPLAY,40,'Indique comprimento horizontal [1..64]:
     .')
      CALL LER12(DISPLAY,4,Y,2,*90)
      STATUS=VCURAD(DISPLAY,XP+5,1)
      STATUS=VCTXTS(DISPLAY,38,'Indique comprimento vertical [1..64]: ')
      CALL LER12(DISPLAY,4,X,2,*90)
C
C     ARMAZENAR FIGURA NA MEMORIA DE IMAGEM
C
      DO 21 I=1,64
      DO 21 J=1,64
      IA(I,J)=CE
      IF(I.GE.XV.AND.I.LT.(XV+X).AND.J.GE.YV.AND.J.LT.(YV+Y))
     .  IA(I,J)=CI
21    CONTINUE
C
      GOTO 91
     ENDIF
C
      IF(CHAIN.EQ.3) THEN
      STATUS=VCURAD(DISPLAY,XP,1)
      STATUS=VEREOS(DISPLAY)
```

```
      STATUS=VCTITS(DISPLAY,46,'Indique nivel de cinza do quadrado 1 [0.
     ..31]: ')
      CALL LER12(DISPLAY,4,COR1,1,*90)
      STATUS=VCURAD(DISPLAY,XP+1,1)
      STATUS=VCTITS(DISPLAY,46,'Indique nivel de cinza do quadrado 2 [0.
     ..31]: ')
      CALL LER12(DISPLAY,4,COR2,1,*90)
      STATUS=VCURAD(DISPLAY,XP+2,1)
      STATUS=VCTITS(DISPLAY,46,'Indique nivel de cinza do quadrado 3 [0.
     ..31]: ')
      CALL LER12(DISPLAY,4,COR3,1,*90)
      STATUS=VCURAD(DISPLAY,XP+3,1)
      STATUS=VCTITS(DISPLAY,46,'Indique nivel de cinza do quadrado 4 [0.
     ..31]: ')
      CALL LER12(DISPLAY,4,COR4,1,*90)
C
C     GERAR RETANGULOS E ARMAZENAR NA MEMORIA DE IMAGEM ESPECIFICADA
C
      DO 30 I=1,32
      DO 30 J=1,32
30    IA(I,J)=COR1
      DO 31 I=1,32
      DO 31 J=33,64
31    IA(I,J)=COR2
      DO 32 I=33,64
      DO 32 J=1,32
32    IA(I,J)=COR3
      DO 33 I=33,64
      DO 33 J=33,64
33    IA(I,J)=COR4
      GOTO 91
     ENDIF
      IF(CHAIN.EQ.4) THEN
      STATUS=VCURAD(DISPLAY,XP,1)
      STATUS=VEREOS(DISPLAY)
      STATUS=VCTITS(DISPLAY,34,'Indique nivel de cinza 1 [0..31]: ')
      CALL LER12(DISPLAY,4,COR1,1,*90)
      STATUS=VCURAD(DISPLAY,XP+1,1)
      STATUS=VCTITS(DISPLAY,34,'Indique nivel de cinza 2 [0..31]: ')
      CALL LER12(DISPLAY,4,COR2,1,*90)
40    STATUS=VCURAD(DISPLAY,XP+2,1)
      STATUS=VCTITS(DISPLAY,44,'Indique numero de divisoes [2, 4, 8 ou 1
     .6]: ')
      CALL LER12(DISPLAY,4,NQH,3,*90)
      IF(NQH.NE.2.AND.NQH.NE.4.AND.NQH.NE.8.AND.NQH.NE.16) CALL
     .MENSAGE(DISPLAY,3,*90)
C
      DHQ=64/NQH
      DO 210 I=1,64
      DO 205 J=1,64
      IA(I,J)=COR1
      IF(((J/DHQ)*DHQ).LT.J) GOTO 205
      T=COR1
      COR1=COR2
      COR2=T
205   CONTINUE
      IF(((I/DHQ)*DHQ).LT.I) GOTO 210
      T=COR1
```

```
      COR1=COR2
      COR2=T
210   CONTINUE
      GOTO 91
     ENDIF
C
90    STATUS=VRVOFF(DISPLAY)
      RETURN
91    STATUS=VRVOFF(DISPLAY)
      CALL MENSINF(DISPLAY,2)
C
      END


      SUBROUTINE PISCA (IVAL,YVAL,TEXT,FLAG,DISPLAY,NUM,*)
************************************************************
*     DEFINE CARACTERISTICAS NO MODO CURSOR
************************************************************
C
      IMPLICIT INTEGER*2 (C-Z)
      DIMENSION REQATT(4),REQT(4)
      CHARACTER TEXT*62,STRING,STRIND
      COMMON /COMUNIC/STRING,STRIND
      DATA REQATT,REQT /0,0,0,0,1,0,1,0/
      STATUS=VCRATT(DISPLAY,REQT,REQT)
           status=vrvon(display)
C
      STATUS=VCURAD(DISPLAY,IVAL,YVAL)
      STATUS=VCTXTS(DISPLAY,NUM,TEXT)
      STATUS=VCRATT(DISPLAY,REQATT,REQATT)
          status=vrvoff(display)
C
      IF(FLAG.EQ.0) RETURN
      CALL LEITURA (DISPLAY,FLAG,*10)
      RETURN
10    RETURN 1
      END


    SUBROUTINE INOUT(DISPLAY)
****************************************
*     EXECUTA ENTRADA E SAIDA DE DADOS
****************************************
C
      IMPLICIT INTEGER*2 (C-Z)
      CHARACTER SUBMENU*14,F(4)*30,NOME*14,STRING,STRIND
      COMMON IA(64,64),IB(64,64),IC(64,64),NOME,/COMUNIC/STRING,STRIND
      DATA SUBMENU /'E/S DE IMAGENS'/
      DATA F(1) /'F1 Carrega imagem'/
      DATA F(2) /'F2 Armazena imagem'/
      DATA F(3) /'F3 Imprime imagem'/
      DATA F(4) /'F4 Transfere imagens entre MIs'/
C
C     APRESENTA OPCOES DO SUBMENU: E/S DE IMAGENS
C
10    STATUS=VCLRWK(DISPLAY)
      CALL MCURSOR(DISPLAY,SUBMENU,14)
C
```

```
        I=1
        DO 6 J=10,16,2
        STATUS=VCURAD(DISPLAY,J,10)
        STATUS=VCTITS(DISPLAY,30,F(I))
6       I=I+1
        CALL FINAL(DISPLAY,1)
C
C     LER OPCAO DE ENTRADA
C
        STATUS=VRQCHC(DISPLAY,CHAIN,CHAIN)
        IF(CHAIN.EQ.1.AND.STATUS.GT.0) THEN
        CALL PISCA(10,10,F(1),2,DISPLAY,17,*10)
        CALL ESI(1,DISPLAY)
      ENDIF
        IF(CHAIN.EQ.2) THEN
        CALL PISCA(12,10,F(2),15,DISPLAY,18,*10)
        CALL ESI(-1,DISPLAY)
      ENDIF
        IF(CHAIN.EQ.3) THEN
        CALL PISCA (14,10,F(3),4,DISPLAY,17,*10)
        IF(STRING.EQ.'1') CALL DSP(IA,DISPLAY,*10)
        IF(STRING.EQ.'2') CALL DSP(IB,DISPLAY,*10)
        IF(STRING.EQ.'3') CALL DSP(IC,DISPLAY,*10)
      ENDIF
C
        IF(CHAIN.EQ.4) THEN
        CALL PISCA(16,10,F(4),6,DISPLAY,30,*10)
        STRIND=STRING
        CALL LEITURA(DISPLAY,12,*10)
        CALL MINVERT(DISPLAY)
      ENDIF
C
        IF(CHAIN.EQ.9) CALL AUXILIO(DISPLAY,1)
C
        IF(CHAIN.NE.10) GOTO 10
        RETURN
        END
C
      SUBROUTINE SETA (DISPLAY,XY,END1,END2,DIREC)
C*************************************************************
C     INDICA POSICAO NAS COORDENADAS X,Y DE UM GRAFICO
C     ATRAVES DE SETAS
C*************************************************************
C
        IMPLICIT INTEGER*2 (C-Z)
        DIMENSION SENT(6)
        CHARACTER DIREC
C
        SENT(1)=XY-200
        SENT(2)=END1-200
        SENT(3)=XY
        SENT(4)=END1
        SENT(5)=XY+200
        SENT(6)=SENT(2)
        IF(DIREC.EQ.'V') GOTO 1
        SENT(2)=END1+200
        SENT(5)=SENT(1)
        SENT(6)=END1-200
C
```

```
 1      STATUS=VPLINE(DISPLAY,3,SENT)
 C
        SENT(1)=XY
        SENT(2)=END1
        SENT(3)=XY
        SENT(4)=END1-1500
        IF(DIREC.EQ.'V') GOTO 2
        SENT(3)=XY-1500
        SENT(4)=END1
 2      STATUS=VPLINE(DISPLAY,2,SENT)
 C
        IF(END2.EQ.0) RETURN
 C
        SENT(1)=XY-200
        SENT(2)=SENT(4)
        SENT(3)=XY+200
        SENT(4)=END1
        SENT(5)=END2-200
        SENT(6)=SENT(2)
        IF(DIREC.EQ.'V') GOTO 3
        SENT(1)=XY-1500
        SENT(2)=END1-200
        SENT(3)=XY
        SENT(4)=END1+200
        SENT(5)=SENT(1)
        SENT(6)=END2-200
 C
 3      STATUS=VCPPEL(DISPLAY,SENT)
        END

      SUBROUTINE INFORM (DISPLAY,N,ENDX,ENDY,*,*,*)
C*********************************************************************
C    A SUBROTINA ABAIXO FAZ A LEITURA DOS VALORES DE ENTRADA DO PROGRAMA
C*********************************************************************
      IMPLICIT INTEGER*2 (C-Z)
      CHARACTER CHARACT*6,INPUT(5),temp
      DIMENSION XYEND(2),EINT(5)
      COMMON/INFO/ECXY(2),/DADO3/DIGIT1
      EQUIVALENCE (CHARACT,INPUT)
C
15    STATUS=VSACOL(DISPLAY,3)
      STATUS=VSAUND(DISPLAY,1)
      STATUS=VSAPOS(DISPLAY,400,4000,XYEND(1),XYEND(2))
      XYEND(1)=ENDX
      XYEND(2)=ENDY
      CHARACT='  '
      GOTO (1,2,3,4) N
1     STATUS=VATXTS(DISPLAY,34,'Indique 1 valor entre 0 e 31:     ',
     .XYEND(1),XYEND(2))
      XYEND(1)=XYEND(1)-(2400+820)
      STATUS=VGTXTS(DISPLAY,400,3000,17,'(ESC) Abandona   ')
      CONT=3
      ASSIGN 15 TO REJEIT
      GOTO 17
C
2     STATUS=VATXTS(DISPLAY,38,'Indique 2 valores entre 0 e 31:      ',
     .XYEND(1),XYEND(2))
      XYEND(1)=XYEND(1)-(4040+820)
      STATUS=VGTXTS(DISPLAY,400,3000,17,'(ESC) Abandona   ')
```

```
4       CONT=6
        ASSIGN 15 TO REJEIT
        GOTO 17
C
3       CONT=2
        TEMP=CHAR(DIGIT1)
        ASSIGN 13 TO REJEIT
C
17      STATUS=VSACOL(DISPLAY,1)
        STATUS=VSAUND(DISPLAY,0)
        STATUS=VSAPOS(DISPLAY,XYEND(1),XYEND(2),LIMX,LIMY)
      I=1
5     STATUS=VRQSTR(DISPLAY,1,0,XYEND,INPUT(I))
      INT=ICHAR(INPUT(I))
      IF(INT.EQ.0) GOTO 5
      IF(INPUT(1).EQ.' ') RETURN 1
      IF(INPUT(I).EQ.' ') GOTO 6
C
C     VERIFICA SE OPCAO <ESC> (CANCELA)
C
      IF(INT.EQ.27.and.(n.eq.1.or.n.eq.2)) RETURN 2
C
C     APAGAR CARACTER PROVENIENTE DA SUBROTINA ESTATLOC E RETORNAR
C
      IF(INT.EQ.75.AND.N.EQ.3.AND.I.EQ.1) THEN
      STATUS=VSAPOS(DISPLAY,XYEND(1)-1638,XYEND(2),XYEND(1),XYEND(2))
      STATUS=VATXTS(DISPLAY,3,'   ',XYEND(1),XYEND(2))
      RETURN 3
      ENDIF
C
C     APAGAR CARACTERES A ESQUERDA
C
      IF(INT.EQ.75) THEN
      INPUT(I)=' '
      IF(I.NE.1) I=I-1
      INPUT(I)=' '
      STATUS=VQAPOS(DISPLAY,XOUT,YOUT)
      STATUS=VSAPOS(DISPLAY,XOUT-820,YOUT,XYEND(1),XYEND(2))
      STATUS=VATXTS(DISPLAY,1,' ',XYEND(1),XYEND(2))
      IF((XOUT-819).LT.LIMX) XOUT=XOUT+819
      STATUS=VSAPOS(DISPLAY,XOUT-819,YOUT,XYEND(1),XYEND(2))
      GOTO 5
      ENDIF
C
C     LER PROXIMO CARACTER
C
      STATUS=VATXTS(DISPLAY,1,INPUT(I),XYEND(1),XYEND(2))
      I=I+1
      IF(I.NE.(CONT+1)) GOTO 5
C
6       IF(INPUT(CONT).NE.' ') GOTO REJEIT
        IF(N.NE.3) GOTO 14
        INPUT(3)=INPUT(2)
        INPUT(2)=INPUT(1)
        INPUT(1)=temp
C
14      IAN=1
        IPO=2
        NEC=1
```

```
      CONT=0
18    DO 10 I=IAN,IPO
      EINT(I)=ICHAR(INPUT(I))
      IF(EINT(I).LT.48.OR.EINT(I).GT.57) GOTO REJEIT
      IF(INPUT(IPO).NE.' '.AND.INPUT(IPO).NE.',') GOTO 16
      ECIY(NEC)=EINT(IAN)-48
      GOTO 1000
16    EINT(I)=EINT(I)-48
10    CONTINUE
      I=2
      ECIY(NEC)=EINT(IAN)*10+EINT(IPO)
1000  IF(CONT.EQ.1) GOTO 1001
      IAN=I+2
      IPO=IAN+1
      NEC=2
      CONT=1
      IF(N.EQ.2.OR.N.EQ.4) GOTO 18
C
C     INFORMAR SOBRE VALORES DE ENTRADA FORA DA FAIXA [0..31]
C
1001  IF(ECIY(1).GT.31.OR.ECIY(2).GT.31) CALL MENSAGE(DISPLAY,4,*13)
      RETURN
13    RETURN 2
      END
```

```
            SUBROUTINE MINVERT (DISPLAY)
*****************************************************
*     REALIZA O MOVIMENTO DE ARQUIVOS DE IMAGENS
*****************************************************
C
C     REALIZA A TRANSFERENCIA ENTRE MIs
C
      IMPLICIT INTEGER*2(C-Z)
      CHARACTER NOME*14,STRING,STRIND
      COMMON IA(64,64),IB(64,64),IC(64,64),NOME,/COMUNIC/STRIND,STRING
     .      /SINAL/INDICA,INDICB
C
C     EFETUAR TRANSFERENCIA
C
      IF(STRIND.EQ.'1') INDICA=0
      IF(STRIND.EQ.'2') INDICB=0
C
      IF(STRING.EQ.'1'.AND.STRIND.EQ.'2') CALL INV1(IA,IB)
      IF(STRING.EQ.'1'.AND.STRIND.EQ.'3') CALL INV1(IA,IC)
      IF(STRING.EQ.'2'.AND.STRIND.EQ.'1') CALL INV1(IB,IA)
      IF(STRING.EQ.'2'.AND.STRIND.EQ.'3') CALL INV1(IB,IC)
      IF(STRING.EQ.'3'.AND.STRIND.EQ.'1') CALL INV1(IC,IA)
      IF(STRING.EQ.'3'.AND.STRIND.EQ.'2') CALL INV1(IC,IB)
C
      CALL MENSINF(DISPLAY,2)
      RETURN
      END
C
```

```
      SUBROUTINE INV1(IA,IB)
      IMPLICIT INTEGER*2 (C-Z)
      DIMENSION IA(64,64),IB(64,64)
C
      DO 10 I=1,64
      DO 10 J=1,64
10    IB(I,J)=IA(I,J)
      RETURN
```

```
      SUBROUTINE TRANSF(DISPLAY)
*****************************************************
*     APRESENTA MENU PARA MODULO DE TRANSFORMACOES
*     RADIOMETRICAS
*****************************************************
C
      IMPLICIT INTEGER*2 (C-Z)
      CHARACTER NOME*14, F(3)*33, SUBMENU*28,STRING,STRIND
C
      COMMON IA(64,64),IB(64,64),IC(64,64),NOME,/COMUNIC/STRING,STRIND
C
      DATA SUBMENU /'TRANSFORMACOES RADIOMETRICAS'/
      DATA F(1) /'F1 Equalizacao histogramica      '/
      DATA F(2) /'F2 Mapeamento dos niveis de cinza'/
      DATA F(3) /'F3 Display de imagens '/
C
C  APRESENTAR OPCOES DO SUBMENU: TRANSFORMACOES RADIOMETRICAS
C
10    STATUS=VCLRWK(DISPLAY)
      CALL MCURSOR(DISPLAY,SUBMENU,28)
C
      I=1
      DO 5 J=10,14,2
      STATUS=VCURAD(DISPLAY,J,10)
      STATUS=VCTXTS(DISPLAY,33,F(I))
      I=I+1
5     CONTINUE
C
C
      CALL FINAL(DISPLAY,1)
C
C  LER OPCAO DE ENTRADA
C
      STATUS=VRQCHC(DISPLAY,CHAIN,CAFIN)
      IF(CAFIN.EQ.1.AND.STATUS.GT.0) THEN
      CALL PISCA(10,10,F(1),5,DISPLAY,27,*10)
      IF(STRING.EQ.'1') CALL EQHIST (IA,IC,DISPLAY,1)
      IF(STRING.EQ.'2') CALL EQHIST (IB,IC,DISPLAY,2)
     ENDIF
C
      IF(CAFIN.EQ.2) CALL MAPDIR (DISPLAY)
C
      IF(CAFIN.EQ.3) THEN
      CALL PISCA(14,10,F(3),0,DISPLAY,21,*10)
      CALL IMAGEDIS(DISPLAY,*10)
     ENDIF
C
      IF(CAFIN.EQ.9) CALL AUXILIO(DISPLAY,4)
C
      IF(CAFIN.NE.10) GOTO 10
      RETURN
      END
```

```
      SUBROUTINE MAPDIR(DISPLAY)
*********************************************************
*     EXECUTA O MAPEAMENTO DIRETO DOS NIVEIS DE CINZA
*********************************************************
      IMPLICIT INTEGER*2 (C-Z)
      DIMENSION GXY(8),XY(6)
      CHARACTER NOME*14,TEXT(8)*33,SUBMENU*30,STRING,STRIND,MSG*29
      COMMON IA(64,64),IB(64,64),IC(64,64),NOME
      COMMON /COMUNIC/STRING,STRIND,/INFO/ECXY(2),/BLOC1/EXY(4),E1XY(4),
     .        E2XY(4)
C
      DATA SUBMENU /'MAPEAMENTO DOS NIVEIS DE CINZA'/
      DATA TEXT(1) /'F1 Fatiamento em dois niveis'/
      DATA TEXT(2) /'F2 Compressao'/
      DATA TEXT(3) /'F3 Compressao/expansao monotonica'/
      DATA TEXT(4) /'F4 Fatiamento por plano com fundo'/
      DATA TEXT(5) /'F5 Aumento linear do contraste'/
      DATA TEXT(6) /'F6 Fatiamento por plano'/
      DATA TEXT(7) /'F7 Inversao da escala de cinza'/
      DATA TEXT(8) /'F8 Dente de serra 3-ciclos'/
      DATA MSG /'F1-Copia F2-Repete F3-Retorna'/
C
      DATA X,Y,XY /18200,6000,4000,19000,4000,9000,14000,9000/
C
C     VALORES QUE REPRESENTAM A AREA DE APAGAMENTO DAS SETAS INDICATIVAS
C     NO GRAFICO
C
      E1XY(1)=XY(3)-200
      E1XY(2)=XY(4)-3000
      E1XY(3)=XY(5)+200
      E1XY(4)=XY(4)-1150
C
      E2XY(1)=XY(1)-3700
      E2XY(2)=XY(4)-200
      E2XY(3)=XY(1)-2000
      E2XY(4)=XY(2)+200
C
C     APRESENTAR OPCOES DO SUBMENU: MAPEAMENTO DIRETO
C
5     STATUS=VENCUR(DISPLAY)
      CALL MCURSOR(DISPLAY,SUBMENU,30)
C
      I=1
      DO 6 J=10,16,2
      STATUS=VCURAD(DISPLAY,J,10)
      STATUS=VCTXTS(DISPLAY,33,TEXT(I))
      STATUS=VCURAD(DISPLAY,J,46)
      STATUS=VCTXTS(DISPLAY,33,TEXT(I+1))
6     I=I+2
      CALL FINAL(DISPLAY,2)
C
C    AREA A SER APAGADA NO GRAFICO
C
      EXY(1)=XY(1)+100
      EXY(2)=XY(4)+200
      EXY(3)=XY(5)
      EXY(4)=XY(2)
C
C    LER OPCAO DE ENTRADA
```

```
C
10      STATUS=VROCHC(DISPLAY,CAFIN,CAFIN)
        IF(STATUS.EQ.0) CALL MENUP(DISPLAY)
        GOTO (110,210,310,410,510,610,710,810) CAFIN
        RETURN
C
C    SECAO PARA FATIAMENTO EM DOIS NIVEIS
C
110     CALL PISCA (10,10,TEXT(1),2,DISPLAY,28,*5)
        ASSIGN 120 TO RETORNO
        ASSIGN 115 TO NOMEIA
        GOTO 100
115     STATUS=VATXTS(DISPLAY,10,'Fatiamento',I,J)
120     CALL INFORM (DISPLAY,1,0,0,*120,*5,*5)
C
C    PLOTAR GRAFICO NA TELA
C
        GXY(1)=XY(3)
        GXY(2)=XY(4)
        GXY(3)=ECXY(1)*322+XY(1)
        GXY(4)=XY(4)
C
        GXY(5)=GXY(3)
        GXY(6)=XY(2)
        GXY(7)=XY(5)
        GXY(8)=GXY(6)
        STATUS=VPLINE(DISPLAY,4,GXY)
        CALL SETA(DISPLAY,GXY(3),XY(4)-1200,0,'V')
C
C    REALIZAR TRANSFORMACAO: FATIAMENTO [1]
C
        CALL DECIS (DISPLAY,*120,1)
C
        GOTO 500
C
C    SECAO PARA COMPRESSAO DE NIVEIS
C
210     CALL PISCA (10,46,TEXT(2),2,DISPLAY,13,*5)
        ASSIGN 215 TO NOMEIA
        ASSIGN 220 TO RETORNO
        GOTO 100
215     STATUS=VATXTS(DISPLAY,10,'Compressao',I,J)
220     CALL INFORM (DISPLAY,2,0,0,*220,*5,*5)
C
C    PLOTAR GRAFICO: COMPRESSAO
C
        GXY(1)=XY(3)
        GXY(2)=XY(4)+322*ECXY(1)
C
        GXY(3)=XY(5)
        GXY(4)=XY(4)+322*ECXY(2)
C
        STATUS=VPLINE(DISPLAY,2,GXY)
        CALL SETA(DISPLAY,XY(1)-2000,GXY(2),GXY(4),'H')
C
C    REALIZAR TRANSFORMACAO: COMPRESSAO [2]
C
        CALL DECIS (DISPLAY,*220,2)
C
```

```
      GOTO 500
C
C     SECAO PARA COMPRESSAO E EXPANSAO MONOTONICA
C
310   CALL PISCA (12,10,TEXT(3),2,DISPLAY,33,*5)
      ASSIGN 315 TO NOMEIA
      ASSIGN 320 TO RETORNO
      GOTO 100
315   STATUS=VATXTS(DISPLAY,19,'Compressao/Expansao',1,3)
C
C   PLOTAR GRAFICO
C
320   GXY(1)=XY(3)
      GXY(2)=XY(4)
      GXY(3)=XY(3)+10*322
      GXY(4)=XY(4)+5*322
      GXY(5)=XY(3)+21*322
      GXY(6)=XY(4)+27*322
      GXY(7)=XY(3)+31*322
      GXY(8)=XY(2)
      STATUS=VPLINE(DISPLAY,4,GXY)
C
      CALL DECIS (DISPLAY,*5,3)
C
      GOTO 500
C
C   SECAO PARA FATIAMENTO POR PLANO COM FUNDO
C
410   CALL PISCA (12,46,TEXT(4),2,DISPLAY,33,*5)
      ASSIGN 415 TO NOMEIA
      ASSIGN 420 TO RETORNO
      GOTO 100
415   STATUS=VATXTS(DISPLAY,14,'Plano c/ fundo',1,3)
420   CALL INFORM (DISPLAY,2,0,0,*420,*5,*5)
      IF(ECXY(2).LT.ECXY(1)) CALL MENSAGE(DISPLAY,5,*5)
C
C   PLOTAR GRAFICO: PLANO COM FUNDO
C
      GXY(1)=XY(3)
      GXY(2)=XY(4)
      GXY(3)=ECXY(1)*322+XY(3)
      GXY(4)=ECXY(1)*322+XY(4)
      GXY(5)=GXY(3)
      GXY(6)=XY(2)
      GXY(7)=ECXY(2)*322+XY(3)
      GXY(8)=GXY(6)
      STATUS=VPLINE(DISPLAY,4,GXY)
C
      CALL SETA(DISPLAY,GXY(3),XY(4)-1200,GXY(7),'V')
      GXY(1)=GXY(7)
      GXY(2)=GXY(8)
      GXY(3)=GXY(1)
      GXY(4)=ECXY(2)*322+XY(4)
      GXY(5)=XY(5)
      GXY(6)=XY(2)
      STATUS=VPLINE(DISPLAY,3,GXY)
C
      CALL DECIS(DISPLAY,*420,4)
      GOTO 500
C
```

```
C    SECAO PARA AUMENTO LINEAR DO CONTRASTE
C
510     CALL PISCA (14,10,TEXT(5),2,DISPLAY,30,*5)
        ASSIGN 515 TO NOMEIA
        ASSIGN 520 TO RETORNO
        GOTO 100
515     STATUS=VATXTS(DISPLAY,20,'Aumento de contraste',I,J)
C
520     CALL INFORM (DISPLAY,2,0,0,*520,*5,*5)
        IF(ECXY(2).LT.ECXY(1)) CALL MENSAGE(DISPLAY,5,*5)
C
C    PLOTAR GRAFICO:AUMENTO DE CONTRASTE
C
        GXY(1)=XY(3)
        GXY(2)=XY(4)
        GXY(3)=XY(3)+ECXY(1)*322
        GXY(4)=XY(4)
        GXY(5)=XY(3)+ECXY(2)*322
        GXY(6)=XY(2)
        GXY(7)=XY(5)
        GXY(8)=XY(2)
        STATUS=VPLINE(DISPLAY,4,GXY)
        CALL SETA(DISPLAY,GXY(3),XY(4)-1200,GXY(5),'V')
C
C    REALIZAR AUMENTO DE CONTRASTE (5)
C
        CALL DECIS(DISPLAY,*520,5)
        GOTO 500
C
C    SECAO PARA FATIAMENTO POR PLANO
C
610     CALL PISCA(14,46,TEXT(6),2,DISPLAY,23,*5)
        ASSIGN 615 TO NOMEIA
        ASSIGN 620 TO RETORNO
        GOTO 100
615     STATUS=VATXTS(DISPLAY,19,'Fatiamento p/ plano',I,J)
C
620     CALL INFORM (DISPLAY,2,0,0,*620,*5,*5)
        IF(ECXY(2).LT.ECXY(1)) CALL MENSAGE(DISPLAY,5,*5)
C
C    PLOTAR GRAFICO:FATIAMENTO POR PLANO
C
        GXY(1)=XY(3)
        GXY(2)=XY(4)
        GXY(3)=XY(3)+ECXY(1)*322
        GXY(4)=XY(4)
        GXY(5)=GXY(3)
        GXY(6)=XY(2)
        GXY(7)=XY(3)+ECXY(2)*322
        GXY(8)=XY(2)
        STATUS=VPLINE(DISPLAY,4,GXY)
        CALL SETA(DISPLAY,GXY(3),XY(4)-1200,GXY(7),'V')
C
        GXY(1)=GXY(7)
        GXY(2)=GXY(8)
        GXY(3)=GXY(7)
        GXY(4)=XY(4)
        GXY(5)=XY(5)
        GXY(6)=XY(6)
```

```
      STATUS=VPLINE(DISPLAY,3,GXY)
C
C    REALIZAR FATIAMENTO POR PLANO [6]
C
      CALL DECI6 (DISPLAY,*620,6)
      GOTO 500
C
710   CALL PISCA(16,10,TEXT(7),2,DISPLAY,30,*5)
      ASSIGN 720 TO RETORNO
      ASSIGN 715 TO NOME1A
      GOTO 100
715   STATUS=VATXTS(DISPLAY,8,'Inversao',1,J)
C
C     PLOTAR GRAFICO:INVERSAO DA ESCALA DE CINZA
C
720   GXY(1)=XY(1)
      GXY(2)=XY(2)
      GXY(3)=XY(5)
      GXY(4)=XY(6)
      STATUS=VPLINE(DISPLAY,2,GXY)
C
      CALL DECIS(DISPLAY,*5,7)
      GOTO 500
C
810   CALL PISCA(16,46,TEXT(8),2,DISPLAY,26,*5)
      ASSIGN 820 TO RETORNO
      ASSIGN 815 TO NOME1A
      GOTO 100
815   STATUS=VATXTS(DISPLAY,14,'Dente de serra',1,J)
C
C     PLOTAR GRAFICO: DENTE DE SERRA
C
820   GXY(1)=XY(3)
      GXY(2)=XY(4)
      GXY(3)=XY(3)+3220
      GXY(4)=XY(2)
      STATUS=VPLINE(DISPLAY,2,GXY)
C
      GXY(1)=GXY(3)
      GXY(3)=XY(3)+6440
      STATUS=VPLINE(DISPLAY,2,GXY)
C
      GXY(1)=GXY(3)
      GXY(3)=XY(5)
      GXY(4)=XY(2)
      STATUS=VPLINE(DISPLAY,2,GXY)
C
      CALL DECIS(DISPLAY,*5,8)
C
500   STATUS=VGTXTS(DISPLAY,2000,1000,29,MSG)
      STATUS=VRQCHC(DISPLAY,CHAIN,CAFIN)
C
      IF(CAFIN.EQ.1.AND.STATUS.GT.0) THEN
      STATUS=VGTXTS(DISPLAY,2000,1000,29,'
     .')
      STATUS=VGTXTS(DISPLAY,400,3000,17,'                    ')
      STATUS=VHDCPY(DISPLAY)
      STATUS=VGTXTS(DISPLAY,2000,1000,29,MSG)
     ENDIF
C
```

```
        IF(CAFIN.EQ.2) THEN
        STATUS=VCLARY(DISPLAY,EXY,1,1,1,4,0)
        STATUS=VCLARY(DISPLAY,E1XY,1,1,1,4,0)
        STATUS=VCLARY(DISPLAY,E2XY,1,1,1,4,0)
        STATUS=VGTXTS(DISPLAY,2000,1000,29,'
     .')
        STATUS=VGTXTS(DISPLAY,400,3000,17,'                    ')
        IF(STRING.EQ.'1') CALL IMAGEM(IA,NOME,DISPLAY,X,Y,0,1)
        IF(STRING.EQ.'2') CALL IMAGEM(IB,NOME,DISPLAY,X,Y,0,2)
        GOTO RETORNO
      ENDIF
        IF(CAFIN.NE.3) GOTO 500
        GOTO 5
        STOP
C
C     DESENHA MOLDURA NA TELA E ENTRA NO MODO GRAFICO
C
100     CALL MOLDURA(DISPLAY)
C
        IF(STRING.EQ.'1') CALL IMAGEM (IA,NOME,DISPLAY,X,Y,0,1)
        IF(STRING.EQ.'2') CALL IMAGEM (IB,NOME,DISPLAY,X,Y,0,2)
C
C    DESENHAR EIXOS PARA GRAFICO
C
        STATUS=VPLINE(DISPLAY,3,XY)
C
C    ESCREVER VALORES NOS EIXOS
C
        STATUS=VGTXTS(DISPLAY,XY(1),XY(4)-900,1,'0')
        STATUS=VGTXTS(DISPLAY,XY(5),XY(4)-900,2,'31')
        STATUS=VGTXTS(DISPLAY,XY(1)-1800,XY(2)-800,2,'31')
        STATUS=VSAPOS(DISPLAY,400,XY(2)+1000,1,J)
        GOTO NOMEIA
        END
C
C****************************************
        SUBROUTINE DECIS (DISPLAY,*,FUNC)
C****************************************
C
        IMPLICIT INTEGER*2 (C-Z)
        CHARACTER CHARIN,STRING,STRIND,NOME*14
        DIMENSION EIXY(2)
        COMMON IA(64,64),IB(64,64),IC(64,64),NOME,/BLOCI/EXY(4),
     .E1XY(4),E2XY(4),/COMUNIC/STRING,STRIND
5       STATUS=VSAPOS(DISPLAY,400,2950,XOUT,YOUT)
        STATUS=VATXTS(DISPLAY,19,'PROSSEGUIR(S/N)?    ',EIXY(1),EIXY(2))
        EIXY(1)=EIXY(1)-2460
        EIXY(2)=EIXY(2)+200
        CHARIN=' '
        STATUS=VRDSTR(DISPLAY,1,1,EIXY,CHARIN)
C
        IF(CHARIN.EQ.'N'.OR.CHARIN.EQ.'n') THEN
        STATUS=VCLARY(DISPLAY,EXY,1,1,1,4,0)
        STATUS=VGTXTS(DISPLAY,400,3000,17,'                    ')
C
C     APAGAR SETAS DA TELA
C
```

```
      STATUS=VCLARY(DISPLAY,E1XY,1,1,1,4,0)
      STATUS=VCLARY(DISPLAY,E2XY,1,1,1,4,0)
      RETURN 1
     ENDIF
      IF(CHARIN.NE.'S'.AND.CHARIN.NE.'s') GOTO 5
C
C    REALIZAR FUNCAO ESPECIFICA
C
      IF(STRING.EQ.'1') CALL REALIZ (DISPLAY,IA,IC,NOME,FUNC)
      IF(STRING.EQ.'2') CALL REALIZ (DISPLAY,IB,IC,NOME,FUNC)
      RETURN
      END
```

```
                    SUBROUTINE IMAGEM(IA,NOME,DISPLAY,X,Y,DM,DN)
************************************************************
* APRESENTA IMAGEM NA TELA. PICTOREA VERSAO 4 CORES
************************************************************
      IMPLICIT INTEGER*2(C-Z)
      CHARACTER*14 NOME , MSG*26,PELA(295),PELB(295)
      DIMENSION IA(64,64),XY(4)
      COMMON/SINAL/INDICA,INDICB
      Y1=Y+14000
      X1=X+3000
      IF(DN.EQ.0)GO TO 2
      IF(DM.EQ.0.OR.DM.EQ.1)GO TO 3
      X1=103*X
      Y1=103*Y
3     STATUS=VSAPOS(DISPLAY,X1,Y1,XOUT,YOUT)
      STATUS=VATXTS(DISPLAY,14,NOME,XOUT,YOUT)
2     X2=X
      XY(1)=X
      XY(2)=Y
      IF(INDICA.EQ.1.AND.DN.EQ.1) THEN
      STATUS=VPTPEL(DISPLAY,XY,PELA)
      GOTO 11
     ENDIF
      IF(INDICB.EQ.1.AND.DN.EQ.2) THEN
      STATUS=VPTPEL(DISPLAY,XY,PELB)
      GOTO 11
     ENDIF
      DO 10 I=1,64
      Y2=Y
      XY(1)=X2
      XY(3)=X2+210
      DO 5 J=64,1,-1
      XY(2)=Y2
      XY(4)=Y2+210
      IMAGE=IA(J,I)
      COR=0
      IF(IMAGE.GE.8.AND.IMAGE.LT.16) COR=2
      IF(IMAGE.GE.16.AND.IMAGE.LT.24) COR=3
      IF(IMAGE.GE.24) COR=1
      STATUS=VCLARY(DISPLAY,XY,1,1,1,4,COR)
      Y2=Y2+210
5     CONTINUE
      X2=X2+210
10    CONTINUE
      XY(1)=X
```

```
      XY(2)=Y
      XY(3)=X+13440
      XY(4)=Y+13440
      IF(INDICA.EQ.0.AND.DN.EQ.1) THEN
      STATUS=VGTPEL(DISPLAY,XY,PELA)
      INDICA=1
     ENDIF
      IF(INDICB.EQ.0.AND.DN.EQ.2) THEN
      STATUS=VGTPEL(DISPLAY,XY,PELB)
      INDICB=1
     ENDIF
11    IF(DN.EQ.0) RETURN
      CALL CRFIM(DISPLAY)
C
      END
C
C
      SUBROUTINE CRFIM(DISPLAY)
***********************************************
*     APRESENTA OPCOES DE COPIA,RETORNA E FIM
***********************************************
C
      IMPLICIT INTEGER*2 (C-Z)
C
      STATUS=VGTXTS(DISPLAY,6000,1000,26,'F1-Copia F2-Retorna F3-Fim')
15    STATUS=VRQCHC(DISPLAY,COUT,COUT)
      IF(COUT.EQ.1.AND.STATUS.GT.0) THEN
      STATUS=VGTXTS(DISPLAY,6000,1000,26,'                          ')
      STATUS=VHDCPY(DISPLAY)
      STATUS=VGTXTS(DISPLAY,6000,1000,26,'F1 Copia F2 Retorna F3 Fim')
     ENDIF
      IF(COUT.EQ.2) THEN
      STATUS=VENCUR(DISPLAY)
      RETURN
     ENDIF
      IF(COUT.NE.3) GOTO 15
      STATUS=VENCUR(DISPLAY)
      STATUS=VCRCOL(DISPLAY,1,0,X,Y)
      STATUS=VCLRWK(DISPLAY)
      STATUS=VCLSWK(DISPLAY)
      STOP
      END


            SUBROUTINE GRAFO2(NP,DISPLAY,X,Y,ID)
*************************************************************
* APRESENTA GRAFICOS NA TELA
*************************************************************
      IMPLICIT INTEGER*2 (C-Z)
      DIMENSION NP(ID),XY(4),COR(1)
      NPMAX=0
      DO 5 I=1,ID
      IF(NP(I).GT.NPMAX) NPMAX=NP(I)
5     CONTINUE
      XMAX=ID
      XMIN=1
      YMAX=31
      YMIN=0
C
C     LIMPA AREA RESERVADA PARA O GRAFICO
```

```
C
      COR(1)=0
      XY(1)=X
      XY(2)=Y
      XY(3)=X+11000
      XY(4)=Y+11000
      STATUS=VCLARY(DISPLAY,XY,1,1,1,4,COR)
C
C     DESENHA EIXOS PARA GRAFICO
C
      IF(ID.EQ.64) GO TO 7
      XMAX=ID-1
      XMIN=0
      YMAX=NPMAX
C
C     EIXO VERTICAL
7     XY(1)=X+4250
      XY(2)=Y+1000
      XY(3)=XY(1)
      XY(4)=Y+11000
      STATUS=VPLINE(DISPLAY,2,XY)
C
C     DESENHA EIXO HORIZONTAL
C
      XY(3)=X+11000
      XY(4)=XY(2)
      STATUS=VPLINE(DISPLAY,2,XY)
C
C     ESCREVE LIMITES NOS EIXOS
      XN=X
      YN=Y+10575
      CALL DSPNUM(0.,YMAX,XN,YN,0,DISPLAY)
      XN=X+1000
      YN=Y
      CALL DSPNUM(0.,XMIN,XN,YN,0,DISPLAY)
      XN=X+8000
      YN=Y
      CALL DSPNUM(0.,XMAX,XN,YN,0,DISPLAY)
C
C     TRACA GRAFICO
C
      XY(2)=Y+1000
      AA=210.*32./ID
      DO 10 I=1,ID
      XY(1)=I*AA+X+4380
      XY(3)=XY(1)
      XY(4)=(10000*NP(I))/NPMAX+Y+1000
      STATUS=VPLINE(DISPLAY,2,XY)
10    CONTINUE
      RETURN
      END

      SUBROUTINE DSPHIS(DISPLAY)
*****************************************************
*     APRESENTA MENU PARA FUNCOES DO MODULO DISPLAY
*     DE HISTOGRAMAS
*****************************************************
C
      IMPLICIT INTEGER*2 (C-Z)
      CHARACTER NOME*14,STRING,STRIND,SUBMENU*26,F(4)*39
      COMMON IA(64,64),IB(64,64),IC(64,64),NOME,/COMUNIC/STRING,STRIND
```

```
C
      DATA SUBMENU /'HISTOGRAMAS E ESTATISTICAS'/
      DATA F(1) /'F1 Display de histograma'/
      DATA F(2) /'F2 Display de perfil de linha ou coluna'/
      DATA F(3) /'F3 Estatisticas locais'/
      DATA F(4) /'F4 Verifica e/ou modifica pixels'/
C
C     APRESENTA OPCOES
C
5     STATUS=VCLRWK(DISPLAY)

      CALL MCURSOR(DISPLAY,SUBMENU,26)
C
      I=1
      DO 1 J=10,16,2
      STATUS=VCURAD(DISPLAY,J,10)
      STATUS=VCTXTS(DISPLAY,39,F(I))
1     I=I+1
C
      CALL FINAL(DISPLAY,1)
C
      STATUS=VRQCHC(DISPLAY,CHFIN,CHFIN)
C
      IF(CHFIN.EQ.1.AND.STATUS.GT.0) THEN
      CALL PISCA(10,10,F(1),1,DISPLAY,24,*5)
      IF(STRING.EQ.'1') CALL HISTO(IA,NOME,DISPLAY,1)
      IF(STRING.EQ.'2') CALL HISTO(IB,NOME,DISPLAY,2)
      IF(STRING.EQ.'3') CALL HISTO(IC,NOME,DISPLAY,3)
     ENDIF
C
      IF(CHFIN.EQ.2) THEN
      CALL PISCA(12,10,F(2),1,DISPLAY,39,*5)
      IF(STRING.EQ.'1') CALL PERFIL(IA,NOME,DISPLAY,1)
      IF(STRING.EQ.'2') CALL PERFIL(IB,NOME,DISPLAY,2)
      IF(STRING.EQ.'3') CALL PERFIL(IC,NOME,DISPLAY,3)
     ENDIF
C
      IF(CHFIN.EQ.3) THEN
      CALL PISCA (14,10,F(3),1,DISPLAY,22,*5)
      IF(STRING.EQ.'1') CALL ESTATLOC(IA,NOME,DISPLAY,1,1)
      IF(STRING.EQ.'2') CALL ESTATLOC(IB,NOME,DISPLAY,1,2)
      IF(STRING.EQ.'3') CALL ESTATLOC(IC,NOME,DISPLAY,1,3)
     ENDIF
C
      IF(CHFIN.EQ.4) THEN
      CALL PISCA (16,10,F(4),1,DISPLAY,32,*5)
      IF(STRING.EQ.'1') CALL ESTATLOC(IA,NOME,DISPLAY,2,1)
      IF(STRING.EQ.'2') CALL ESTATLOC(IB,NOME,DISPLAY,2,2)
      IF(STRING.EQ.'3') CALL ESTATLOC(IC,NOME,DISPLAY,2,3)
     ENDIF
      IF(CHFIN.EQ.9) CALL AUXIL10(DISPLAY,2)
C
      IF(CHFIN.NE.10) GOTO 5
      END
```

```
              SUBROUTINE PERFIL (IA,NOME,DISPLAY,DN)
************************************************
*     REALIZA PERFIL DE LINHA E DE COLUNA
************************************************
C
       IMPLICIT INTEGER*2 (C-Z)
       CHARACTER NOME*14,MSG1*34,MSG2*34,MSG3*34,TEXT*2
       DIMENSION IA(64,64),NP(64),COR(1),XY(4)
       DATA MSG1 /'F1-Linha F2-Coluna F3-Retorna'/
       DATA MSG2 /'Posicione o cursor e tecle <ENTER>'/
       DATA MSG3 /'F1-Copia F2-Repete F3-Retorna     '/
       DATA XIM,YIM,YIM2,YIM3,XIM2,XIM3,COR /3000,8000,21335,8105,16335,
     .3105,0/
       DATA XY /22350,9200,29000,19100/
C
C     DESENHA MOLDURA NA TELA E ENTRA NO MODO GRAFICO
C
       CALL MOLDURA(DISPLAY)
       CALL CONTA3(0,0,IA,NP)
C
       CALL IMAGEM (IA,NOME,DISPLAY,XIM,YIM,0,DN)
       CALL ESTAT (NP,DISPLAY,2900)
4      STATUS=VSAPOS(DISPLAY,4000,210,XOUT,YOUT)
       STATUS=VATXTS(DISPLAY,34,MSG1,XOUT,YOUT)
C
8      STATUS=VRBCHC(DISPLAY,CHIN,CHFIN)
       IF(STATUS.EQ.0) GOTO 8
       GOTO (10,10,35) CHFIN
       GOTO 8
C
C    EXECUTAR PERFIL DE LINHA DE VARREDURA
C    POSICIONAR CURSOR:
10     STATUS=VSAPOS(DISPLAY,4000,210,XOUT,YOUT)
       STATUS=VATXTS(DISPLAY,34,MSG2,XOUT,YOUT)
       STATUS=VSAPOS(DISPLAY,4000,1500,XOUT,YOUT)
       STATUS=VATXTS(DISPLAY,7,'CURSOR:',XOUT,YOUT)
       STATUS=VSAPOS(DISPLAY,10000,1500,XOUT,YOUT)
C
       IF(CHFIN.EQ.2) THEN
       STATUS=VATXTS(DISPLAY,10,'Coluna=    ',XOUT,YOUT)
       XIM1=XIM+6615
       YIM1=YIM-1000
       GOTO 20
      ENDIF
C
       STATUS=VATXTS(DISPLAY,10,' Linha=    ',XOUT,YOUT)
9      XIM1=XIM-1000
       YIM1=YIM+6615+210
C
20     STATUS=VDSPCR(DISPLAY,XIM1,YIM1)
C
C    ESCREVER VALOR NUMERICO DE L OU C (NUMERO DA LINHA OU COLUNA)
C
       IF(CHFIN.EQ.1) LC=65-((YIM1-YIM)/210 +1)
       IF(CHFIN.EQ.2) LC=(XIM1-XIM)/210 +1
C
       WRITE(TEXT,'(I2)') LC
       STATUS=VGTXTS(DISPLAY,XOUT-2500,YOUT+200,2,TEXT)
16     STATUS=VRDCKY(DISPLAY,1,DIR,KEY)
       IF(DIR.EQ.0) GOTO 30
```

```
      IF(CHFIN.EQ.1.AND.DIR.EQ.2) YIM1=YIM1-210
      IF(CHFIN.EQ.1.AND.DIR.EQ.8) YIM1=YIM1+210
      IF(CHFIN.EQ.2.AND.DIR.EQ.4) XIM1=XIM1-210
      IF(CHFIN.EQ.2.AND.DIR.EQ.6) XIM1=XIM1+210
      IF(DIR.EQ.1.OR.DIR.EQ.3.OR.DIR.EQ.7.OR.DIR.EQ.9) GOTO 16
C
C    CONTROLE DO CURSOR GRAFICO NO CAMPO DE IMAGEM DA TELA
C
      IF(CHFIN.EQ.1.AND.YIM1.GT.YIM2) YIM1=YIM3
      IF(CHFIN.EQ.1.AND.YIM1.LT.YIM3) YIM1=YIM2
      IF(CHFIN.EQ.2.AND.XIM1.GT.XIM2) XIM1=XIM3
      IF(CHFIN.EQ.2.AND.XIM1.LT.XIM3) XIM1=XIM2
C
      STATUS=VREMCR(DISPLAY)
      GOTO 20
C
C    RECEPCAO DE TECLA ENTER:
C
30    IF(CHFIN.EQ.1) CALL CONTA3 (LC,0,IA,NP)
      IF(CHFIN.EQ.2) CALL CONTA3 (0,LC,IA,NP)
      CALL GRAFO2 (NP,DISPLAY,18000,8000,64)
      STATUS=VSAPOS(DISPLAY,4000,210,XOUT,YOUT)
      STATUS=VATXTS(DISPLAY,34,MSG3,XOUT,YOUT)
5     STATUS=VRQCHC(DISPLAY,CHIN,CHFIN)
      IF(CHFIN.EQ.2) THEN
      STATUS=VREMCR(DISPLAY)
      STATUS=VCLARY(DISPLAY,XY,1,1,1,4,COR)
      STATUS=VSAPOS(DISPLAY,4000,210,XOUT,YOUT)
      STATUS=VATXTS(DISPLAY,20,'                    ',XOUT,YOUT)
      GOTO 4
     ENDIF
      IF(CHFIN.EQ.1.AND.STATUS.GT.0) THEN
      STATUS=VGTXTS(DISPLAY,4000,310,34,'
     .    ')
      STATUS=VHDCPY(DISPLAY)
      STATUS=VSAPOS(DISPLAY,4000,210,XOUT,YOUT)
      STATUS=VATXTS(DISPLAY,34,MSG3,XOUT,YOUT)
     ENDIF
      IF(CHFIN.NE.3) GOTO 5
35    STATUS=VENCUR(DISPLAY)
      RETURN
      END

      SUBROUTINE CONTA3(L,C,IA,NP)
*********************************************************************
* FORNECE INFORMACOES REFERENTES A DISTRIBUICAO DOS NIVEIS DE CINZA
* DOS PIXELS DA IMAGEM
*********************************************************************
C
      IMPLICIT INTEGER*2 (C-Z)
      DIMENSION IA(64,64),NP(64)
      DATA LIN,LFN,CIN,CFN /1,64,1,64/
      DO 10 I=1,64
10    NP(I)=0
      IF(L.EQ.0 .AND. C.EQ.0) GO TO 30
      IF(L.EQ.0) GO TO 20
      LIN=L
      LFN=L
20    IF(C.EQ.0) GO TO 21
      CIN=C
```

```
      CFN=C
21    KS=1
      DO 25 I=LIN,LFN
      DO 25 J=CIN,CFN
      NP(KS)=IA(I,J)
      KS=KS+1
25    CONTINUE
      RETURN
C
30    DO 35 I=1,64
      DO 35 J=1,64
      KS=IA(I,J)+1
      NP(KS)=NP(KS)+1
35    CONTINUE
      RETURN
      END


      SUBROUTINE ESTATLOC(IA,NOME,DISPLAY,FLAG,DN)
C*****************************************************
C     REALIZA A FUNCAO 'ESTATISTICAS LOCAIS'
C*****************************************************
      IMPLICIT INTEGER*2 (C-Z)
      REAL AREA,AREA1
      DIMENSION XYC(4),XYB(4),IA(64,64),NP(32),XYEND(4),LIMP(4),
     .          MIAPAG(4)
      CHARACTER NOME*14,MSG*29,MSG1*32,TEXT*2,OPCAD,JANELA(290)
      COMMON /WIDE/ LINFER,LINSUP,COLINF,COLSUP,/INFO/ECXY(2),
     -       /PONTO/LINHA,COLUNA,XCUR,YCUR,NUML,NUMC,/DADO3/E1NT,
     -       /PASSO1/PASSO,/SINAL/INDICA,INDICB,/PRAESTATLOC/AREA,
     -       /WINDOW/JANELA
      DATA XIM,YIM /1600,8000/
      DATA MSG /'F1-Copia F2-Repete F3-Retorna'/
      DATA MSG1/'F4-Verifica e/ou modifica pixels'/
C
C     DEFINE CARACTERISTICAS DA JANELA (BAR)
C
      STATUS=VSFCOL(DISPLAY,0)
      STATUS=VSFINT(DISPLAY,0)
C
      GOTO (1,2) FLAG
1     CALL MOLDURA(DISPLAY)
      CALL IMAGEM(IA,NOME,DISPLAY,XIM,YIM,0,DN)
      MIAPAG(1)=XIM+64*210+50
      MIAPAG(2)=YIM-200
      MIAPAG(3)=MIAPAG(1)+100
      MIAPAG(4)=YIM+64*210+100
      STATUS=VCLARY(DISPLAY,MIAPAG,1,1,1,4,0)
C
      STATUS=VGTXTS(DISPLAY,2000,1000,30,'Janela: F1-4*4 F2-8*8 F3-16*16
     .')
      STATUS=VRQCHC(DISPLAY,CHIN,CHFIN)
      TF=4
      AREA=4.*4.
      IF(CHFIN.EQ.2) THEN
      TF=8
      AREA=8.*8.
     ENDIF
      IF(CHFIN.EQ.3) THEN
```

```
        TF=16
        AREA=16.*16.
      ENDIF
        AREA1=AREA
C
        STATUS=VGTXTS(DISPLAY,2000,3000,13,'Deslocamento?')
        STATUS=VGTXTS(DISPLAY,2000,2000,16,'F1-Ponto a ponto')
        STATUS=VGTXTS(DISPLAY,2000,1000,30,'F2-Comprimento da janela
     .')
        STATUS=VRQCHC(DISPLAY,CHIN,CHFIN)
        DESLOC=1
        IF(CHFIN.EQ.2) DESLOC=TF
C
        ASSIGN 31 TO RETORNO
C
3       STATUS=VGTXTS(DISPLAY,2000,3000,34,'Posicione a janela e tecle (EN
     .TER)')
        STATUS=VGTXTS(DISPLAY,2000,2000,22,'F1-Esquerda F2-Direita')
        STATUS=VGTXTS(DISPLAY,2000,1000,33,'F3-P/cima   F4-P/baixo F5-Te
     .rmina')
C
C     DEFINE AREA DA JANELA 5*5
C
        XYB(1)=XIM
        XYB(2)=YIM+13440-TF*210
        XYB(3)=XIM+TF*210
        XYB(4)=YIM+13440
C
        GOTO RETORNO
C
C     DESLOCAR JANELA NA IMAGEM:
C
31      ASSIGN 51 TO RETORNO
        PASSO=0
11      CALL MOVE (xim,yim,TF,XYB,DESLOC,0,DIR,DISPLAY)
C
        IF(DIR.EQ.5) GOTO 70
C
C     APRESENTAR HISTOGRAMA LOCAL:
C
50      LINFER=65-(XYB(4)-YIM)/210 +.5
        LINSUP=LINFER+(TF-1)
        COLSUP=(XYB(3)-XIM)/210 +.5
        COLINF=COLSUP-(TF-1)
        GOTO RETORNO
C
51      CALL CONTA3 (-1,0,1A,NP)
        CALL GRAFO2 (NP,DISPLAY,20000,8000,32)
        AREA=AREA1
        CALL ESTAT(NP,DISPLAY,4000)
C
        GOTO 11
C
70      LIMP(1)=2000
        LIMP(2)=200
        LIMP(3)=30000
        LIMP(4)=4000
        STATUS=VCLARY(DISPLAY,LIMP,1,1,1,4,0)
```

```
      STATUS=VGTXTS(DISPLAY,2000,2000,29,MSG)
      STATUS=VGTXTS(DISPLAY,2000,1000,32,MSG1)
71    STATUS=VRQCHC(DISPLAY,CHIN,CHIN)
      IF(CHIN.EQ.1.AND.STATUS.GT.0) THEN
      STATUS=VGTXTS(DISPLAY,2000,2000,29,'
     .)
      STATUS=VGTXTS(DISPLAY,2000,1000,32,'
     .  ')
      STATUS=VHDCPY(DISPLAY)
      STATUS=VGTXTS(DISPLAY,2000,2000,29,MSG)
      STATUS=VGTXTS(DISPLAY,2000,1000,32,MSG1)
      GOTO 71
     ENDIF
      IF(CHIN.NE.3) GOTO 72
      STATUS=VENCUR(DISPLAY)
      RETURN
72    IF(CHIN.EQ.4) GOTO 2
      IF(CHIN.NE.2) GOTO 71
      STATUS=VCLRWK(DISPLAY)
      GOTO 1
C
C     TRECHO DO PROGRAMA QUE VERIFICA E/OU MODIFICA PIXELS (2):
C
2     CALL MOLDURA(DISPLAY)
      CALL IMAGEM (IA,NOME,DISPLAY,XIM,YIM,0,DN)
      MIAPAG(1)=XIM+64*210+50
      MIAPAG(2)=YIM-200
      MIAPAG(3)=MIAPAG(1)+100
      MIAPAG(4)=YIM+64*210+100
      STATUS=VCLARY(DISPLAY,MIAPAG,1,1,1,4,0)
C
C     COLOCAR TEXTO PARA INTERFACE
C
      TF=5
      DESLOC=5
C
      STATUS=VGTXTS(DISPLAY,2000,4000,10,'Janela:5*5')
C
      ASSIGN 21 TO RETORNO
      GOTO 3
C
21    PASSO=0
      CALL MOVE (XIM,YIM,TF,XYB,DESLOC,0,DIR,DISPLAY)
      IF(DIR.EQ.5) GOTO 95
C
C     APRESENTAR PIXELS NA TELA:
C
      ASSIGN 52 TO RETORNO
      GOTO 50
C
52    STATUS=VGTXTS(DISPLAY,18000,20000,17,'Modifique pixel e')
      STATUS=VGTXTS(DISPLAY,18000,19000,13,'tecle (ENTER)')
      CALL MOSTRA (IA,DISPLAY)
C
      NUML=LINFER+2
      NUMC=COLINF+2
      WRITE(TEXT,'(I2)') NUML
      STATUS=VGTXTS(DISPLAY,19080,7000,7,'Linha =')
      STATUS=VGTXTS(DISPLAY,25680,7000,2,TEXT)
```

```
      WRITE(TEXT,'(I2)') NUMC
      STATUS=VGTXTS(DISPLAY,19080,6000,7,'Coluna=')
      STATUS=VGTXTS(DISPLAY,25680,6000,2,TEXT)
C
C     POSICAO INICIAL DO CURSOR:
C
      XCUR=24330
      YCUR=13455
      LINHA=3
      COLUNA=3
C
C     APRESENTAR CURSOR NO PONTO XCUR E YCUR
C
60    STATUS=VSFCOL(DISPLAY,1)
      XYC(1)=XCUR-600
      XYC(2)=YCUR-750
      XYC(3)=XCUR+1250
      XYC(4)=YCUR+600
      STATUS=VBAR(DISPLAY,XYC)
      STATUS=VSFCOL(DISPLAY,0)
C
C     ESPERA A RECEPCAO DE UMA TECLA VALIDA:
C
61    OPCAO=' '
      STATUS=VRQSTR(DISPLAY,1,0,XYEND,OPCAO)
      EINT=ICHAR(OPCAO)
      IF(EINT.EQ.0) GOTO 61
      IF(EINT.EQ.63) GOTO 95
      STATUS=VBAR(DISPLAY,XYC)
      IF(OPCAO.EQ.' ') GOTO 52
C
C     ESPERA TECLA DE MOVIMENTO DE CURSOR
C
      IF(EINT.EQ.72.OR.EINT.EQ.75.OR.EINT.EQ.77.OR.EINT.EQ.80) CALL
     . MOVE1 (EINT,DISPLAY,*60)
C
C     ESPERA TECLA DE FUNCAO ESPECIAL(MOVIMENTO DA JANELA)
C
      IF(EINT.GT.58.AND.EINT.LT.63) THEN
      LIMP(1)=16500
      LIMP(2)=5500
      LIMP(3)=32600
      LIMP(4)=21000
      STATUS=VCLARY(DISPLAY,LIMP,1,1,1,4,0)
      CALL MOVE(XIM,YIM,TF,XYB,DESLOC,1,EINT,DISPLAY)
      IF(EINT.EQ.5) GOTO 95
      GOTO 50
     ENDIF
C
C     EFETUAR MODIFICACAO DOS VALORES DOS PIXELS
C
      IF(EINT.GT.47.AND.EINT.LT.58) THEN
C
C     APRESENTA PRIMEIRO DIGITO DO DADO DE ENTRADA:
C
      STATUS=VGTXTS(DISPLAY,XCUR-410,YCUR-400,3,'    ')
      STATUS=VSAPOS(DISPLAY,XCUR-410,YCUR-580,XCUR1,YCUR1)
      STATUS=VATXTS(DISPLAY,1,OPCAO,XCUR1,YCUR1)
C
```

```
C     APRESENTA SEGUNDO DIGITO DO DADO DE ENTRADA:
C
        CALL INFORM(DISPLAY,3,XCUR1,YCUR1+100,*85,*80,*61)
        GOTO 90
80      IF(ECXY(1).GT.31) GOTO 97
        STATUS=VGTXTS(DISPLAY,XCUR-410,YCUR-400,3,'   ')
        WRITE(TEXT,'(I2)') IA(NUML,NUMC)
        STATUS=VGTXTS(DISPLAY,XCUR-410,YCUR-450,2,TEXT)
        GOTO 60
85      ECXY(1)=EINT-48
C
C     ARMAZENAR NOVO DADO NA MEMORIA DE IMAGEM
C
90      IA(NUML,NUMC)=ECXY(1)
C
        STATUS=VPTPEL(DISPLAY,XYB,JANELA)
C
C     APRESENTAR NOVO VALOR DO PIXEL NA TELA
C
        XYEND(1)=(NUMC-1)*210+XIM
        XYEND(2)=(64-NUML)*210+YIM
        XYEND(3)=XYEND(1)+210
        XYEND(4)=XYEND(2)+210
        CALL IMAGEP(IA(NUML,NUMC),XYEND,DISPLAY)
C
C     ARMAZENAR AREA MODIFICADA EM JANELA
C
        STATUS=VGTPEL(DISPLAY,XYB,JANELA)
        STATUS=VSWRMD(DISPLAY,12)
        STATUS=VBAR(DISPLAY,XYB)
        STATUS=VSWRMD(DISPLAY,4)
C
        IF(DN.EQ.1) INDICA=0
        IF(DN.EQ.2) INDICB=0
      ENDIF
        GOTO 60
95      CALL CRRET(DISPLAY,*2,2000,2000)
        RETURN
C
97      STATUS=VENCUR(DISPLAY)
        RETURN
        END

              SUBROUTINE MOVE (XIM,YIM,TF,XYB,DESLOC,FLAG,DIR,DISPLAY)
C*********************************************************
C     REALIZA O MOVIMENTO DA JANELA NA IMAGEM
C*********************************************************
        IMPLICIT INTEGER*2 (C-Z)
        DIMENSION XYB(4),LIMP(4)
        CHARACTER JANELA(290)
        COMMON/PASSO1/PASSO,/WINDOW/JANELA
C
        DIR=DIR-58
        FLAG1=FLAG
        FLAG2=FLAG
        DIR1=1
C
C     ESPECIFICA PASSAGEM NA SUBROTINA MOVE
C
```

```
      IF(PASSO.EQ.1) STATUS=VPTPEL(DISPLAY,XYB,JANELA)
      PASSO=1
C
30    STATUS=VSWRMD(DISPLAY,12)
      STATUS=VGTPEL(DISPLAY,XYB,JANELA)
C
C     DESENHAR JANELA
C
      STATUS=VBAR(DISPLAY,XYB)
C
C     LER MOVIMENTO DO CURSOR
C
      IF(FLAB1.EQ.1) GOTO 42
40    STATUS=VRQCHC(DISPLAY,DIR,DIR)
      IF(DIR.GT.5) GOTO 40
      DIR1=STATUS
C
C     APAGAR AREA QUE APRESENTA EFEITOS DE ZOOM E REDUCAO
C
      IF(FLAG2.NE.2.OR.DIR.EQ.5) GOTO 42
      LIMP(1)=17000
      LIMP(2)=6000
      LIMP(3)=31000
      LIMP(4)=21000
      STATUS=VCLARY(DISPLAY,LIMP,1,1,1,4,0)
C
C     RESTAURA AREA ARMAZENADA:
C
42    FLAG1=0
      STATUS=VSWRMD(DISPLAY,4)
      IF(DIR.EQ.5.OR.(DIR.EQ.1).AND.DIR1.EQ.0) RETURN
      STATUS=VPTPEL(DISPLAY,XYB,JANELA)
C
C     DESLOCAR JANELA DE ACORDO COM A INDICACAO DO CURSOR
C
      IF(DIR.EQ.2) THEN
      XYB(1)=XYB(1)+DESLOC*210
      XYB(3)=XYB(3)+DESLOC*210
      IF(XYB(3).LT.(XIM+13440+DESLOC*210)) GOTO 30
      XYB(1)=XIM
      XYB(3)=XIM+TF*210
C
      ELSE IF(DIR.EQ.1) THEN
      XYB(1)=XYB(1)-DESLOC*210
      XYB(3)=XYB(3)-DESLOC*210
      IF(XYB(1).GT.(XIM-DESLOC*210)) GOTO 30
      XYB(1)=XIM+13440-TF*210
      XYB(3)=XIM+13440
C
      ELSE IF(DIR.EQ.4) THEN
      XYB(2)=XYB(2)-DESLOC*210
      XYB(4)=XYB(4)-DESLOC*210
      IF(XYB(2).GT.(YIM-DESLOC*210)) GOTO 30
      XYB(2)=YIM+13440-TF*210
      XYB(4)=YIM+13440
C
      ELSE IF(DIR.EQ.3) THEN
      XYB(2)=XYB(2)+DESLOC*210
      XYB(4)=XYB(4)+DESLOC*210
```

```
      IF(XYB(4).LT.(YIM+13440+DESLOC*210)) GOTO 30
      XYB(2)=YIM
      XYB(4)=YIM+TF*210
     ENDIF
C
      GOTO 30
C
      END

      SUBROUTINE MOVE1 (DIR,DISPLAY,*)
C*****************************************
C     DESLOCA CURSOR NA MATRIZ DE PIXELS
C*****************************************
      IMPLICIT INTEGER*2 (C-Z)
      CHARACTER TEXT*2
      COMMON /PONTO/LINHA,COLUNA,XCUR,YCUR,NUML,NUMC,
     .      /WIDE/LINFER,LINSUP,COLINF,COLSUP
C
      MARCB=5
      MARCE=1
      MARCD=5
      MARCC=1
      IF(COLSUP.GT.64) MARCD=4
      IF(LINSUP.GT.64) MARCB=4
      IF(COLINF.LT.1)  MARCE=2
      IF(LINFER.LT.1)  MARCC=2
C
C     PARA BAIXO
C
      IF(DIR.EQ.80) THEN
      IF((LINHA+1).GT.MARCB) RETURN 1
      YCUR=YCUR-2000
      LINHA=LINHA+1
     NUML=NUML+1
     ENDIF
C
C     ESQUERDA
C
      IF(DIR.EQ.75) THEN
      IF((COLUNA-1).LT.MARCE) RETURN 1
      XCUR=XCUR-2460
      COLUNA=COLUNA-1
      NUMC=NUMC-1
C
C     DIREITA
C
      ELSE IF(DIR.EQ.77) THEN
      IF((COLUNA+1).GT.MARCD) RETURN 1
      XCUR=XCUR+2460
      COLUNA=COLUNA+1
      NUMC=NUMC+1
C
C     PARA CIMA
C
      ELSE IF(DIR.EQ.72) THEN
      IF((LINHA-1).LT.MARCC) RETURN 1
      YCUR=YCUR+2000
      LINHA=LINHA-1
      NUML=NUML-1
     ENDIF
```

```
C
C     ATUALIZAR LINHA(NUML) E COLUNA(NUMC) DA MATRIZ DE PIXELS
C
        WRITE(TEXT,'(I2)') NUML
        STATUS=VGTXTS(DISPLAY,25680,7000,2,TEXT)
        WRITE(TEXT,'(I2)') NUMC
        STATUS=VGTXTS(DISPLAY,25680,6000,2,TEXT)
C
        RETURN 1
        END
C
C
        SUBROUTINE MOSTRA (IA,DISPLAY)
C*****************************************
C     APRESENTA OS NIVEIS DE CINZA NA TELA
C*****************************************
        IMPLICIT INTEGER*2 (C-Z)
        DIMENSION IA(64,64)
        CHARACTER CHARACT(25)*2
        COMMON /WIDE/ LINFER,LINSUP,COLINF,COLSUP
C
        K=1
        DO 101 I=LINFER,LINSUP
        DO 101 J=COLINF,COLSUP
        WRITE(CHARACT(K),'(I2)') IA(I,J)
      IF(I.LT.1.OR.I.GT.64.OR.J.LT.1.OR.J.GT.64) CHARACT(K)='*'
101     K=K+1
C
C     MOSTRA PIXELS
C
        J=17000
        K=1
        DO 5 NL=1,5
        DO 10 I=19000,28840,2460
        STATUS=VGTXTS(DISPLAY,I,J,2,CHARACT(K))
        K=K+1
10      CONTINUE
        J=J-2000
5       CONTINUE
        END

              SUBROUTINE FILTRO (DISPLAY)
*********************************************************
*     REALIZA FILTRAGEM ESPACIAL
*********************************************************
C
        IMPLICIT INTEGER*2 (C-Z)
        CHARACTER NOME*14,STRING,STRIND,SUBMENU*17,F(4)*21
        COMMON IA(64,64),IB(64,64),IC(64,64),NOME,/COMUNIC/STRING,STRIND
        DATA SUBMENU /'FILTROS ESPACIAIS'/
        DATA F(1) /'F1 Convolucao'/
        DATA F(2) /'F2 Detecao de bordas'/
        DATA F(3) /'F3 Suavizacao'/
        DATA F(4) /'F4 Display de imagens'/

C
C     APRESENTA OPCOES DO SUBMENU
C
10      STATUS=VCLRWK(DISPLAY)
        CALL MCURSOR(DISPLAY,SUBMENU,17)
C
```

```
      I=1
      DO 6 J=10,16,2
      STATUS=VCURAD(DISPLAY,J,10)
      STATUS=VCTXTS(DISPLAY,21,F(I))
6     I=I+1
C
      CALL FINAL(DISPLAY,1)
C
C     LER DPCAD DE ENTRADA
C
      STATUS=VRQCHC(DISPLAY,CHAIN,CHAIN)
      IF(CHAIN.EQ.1.AND.STATUS.GT.0) THEN
      CALL PISCA(10,10,F(1),5,DISPLAY,13,*10)
      IF(STRING.EQ.'1') CALL MASCARA(IA,IC,DISPLAY,1)
      IF(STRING.EQ.'2') CALL MASCARA(IB,IC,DISPLAY,2)
     ENDIF
      IF(CHAIN.EQ.2) CALL DETECAD(DISPLAY)
      IF(CHAIN.EQ.3) CALL SUAVI(DISPLAY)
      IF(CHAIN.EQ.4) THEN
      CALL PISCA(16,10,F(4),0,DISPLAY,21,*10)
      CALL IMAGEDIS(DISPLAY,*10)
     ENDIF
C
      IF(CHAIN.EQ.9) CALL AUXILIO(DISPLAY,5)
C
      IF(CHAIN.NE.10) GOTO 10
C
      RETURN
      END

            SUBROUTINE MEDIAFIL (IA,IC,DISPLAY,FLAG)
*************************************************************************
*     REALIZA A FILTRAGEM ESPACIAL UTILIZANDO O FILTRO DA
*     MEDIA (JANELA 3*3)
*************************************************************************
C
      IMPLICIT INTEGER*2 (C-Z)
      DIMENSION IA(64,64),IC(64,64),XY(4)
      DATA PXIM,PYIM/17500,3500/
C
      STATUS=VGTXTS(DISPLAY,6000,20810,26,'Filtro da Media:Janela 3*3')
C
      CALL IMAGEM(IA,'Original      ',DISPLAY,2000,3900,0,FLAG)
      STATUS=VGTXTS(DISPLAY,PXIM+2000,PYIM+14700,10,'Processada')
C
      XY(2)=PYIM+13440+210
      DO 10 X=2,63
      XY(1)=PXIM-210
      XY(2)=XY(2)-210
      XY(4)=XY(2)+210
      DO 10 Y=2,63
      XY(1)=XY(1)+210
      XY(3)=XY(1)+210
      IC(X,Y)=(IA(X-1,Y-1)+IA(X-1,Y)+IA(X-1,Y+1)+IA(X,Y-1)+IA(X,Y)+
     .         IA(X,Y+1)+IA(X+1,Y-1)+IA(X+1,Y)+IA(X+1,Y+1))/9
      CALL IMAGEP(IC(X,Y),XY,DISPLAY)
10    CONTINUE
C
      CALL COMPLET(IC,IA,1,DISPLAY,FLAG)
C
      RETURN
      END
```

```
                SUBROUTINE ORDEMFIL (IA,IC,DISPLAY,FLAG)
*****************************************************************
*     REALIZA A FILTRAGEM ESPACIAL UTILIZANDO OS FILTROS DA ORDEM
*     (JANELAS 3*3 E 5*5)
*****************************************************************
C
        IMPLICIT INTEGER*2 (C-Z)
        CHARACTER TAMANHO*10,TEXT*2
        DIMENSION IA(64,64),IC(64,64),vetor(25),XY(4)
        COMMON/PRASOK/TAMANHO
        DATA PXIM,PYIM /17500,3500/
C
        CALL ENTRE(DISPLAY,E,KIM,1,*100)
        STATUS=VSAPOS(DISPLAY,5200,20810,I,J)
        STATUS=VATXTS(DISPLAY,16,'Filtro da Ordem ',I,J)
        WRITE(TEXT,'(I2)') KIM
        STATUS=VATXTS(DISPLAY,2,TEXT,I,J)
        STATUS=VATXTS(DISPLAY,1,':',I,J)
        STATUS=VATXTS(DISPLAY,10,TAMANHO,I,J)
        YPOS=3900
        IF(E.EQ.5) YPOS=4110
        CALL IMAGEM(IA,'Original       ',DISPLAY,2000,YPOS,0,FLAG)
        STATUS=VGTXTS(DISPLAY,PXIM+2000,PYIM+14700,10,'Processada')
C
        XY(2)=PYIM+13440+210
        PONTOIJ=1
        DIM=2
        IJFIM=62
        IF(E.NE.5) GOTO 2
        IJFIM=60
        DIM=4
        PONTOIJ=2
2       DO 5 I=1,IJFIM
        XY(1)=PXIM-210
        XY(2)=XY(2)-210
        XY(4)=XY(2)+210
        DO 5 J=1,IJFIM
        XY(1)=XY(1)+210
        XY(3)=XY(1)+210
        IW=I+DIM
        JW=J+DIM
        K=1
        DO 10 IF=I,IW
        DO 10 JF=J,JW
        VETOR(K)=IA(IF,JF)
        K=K+1
10      CONTINUE
        MEDIAND=CRECNT(VETOR,E**2,KIM)
C
C     ARMAZENAR VALOR MEDIANO NO PONTO CENTRAL DA JANELA
C
        IC(I+PONTOIJ,J+PONTOIJ)=MEDIAND
        CALL IMAGEP(MEDIAND,XY,DISPLAY)
C
5       CONTINUE
C
        E=E/2
        CALL COMPLET(IC,IA,E,DISPLAY,FLAG)
```

```
C
100     RETURN
        END
C
C
        INTEGER*2 FUNCTION CRECNT(VETOR,INDE,INDK)
        IMPLICIT INTEGER*2 (C-Z)
        DIMENSION VETOR(*)
        COMMON /ORD/ORDEM(25)
C
        DO 5 K=1,INDE
        VALMIN=MIN(32,VETOR(1))
        DO 10 I=2,INDE
        VALMIN=MIN(VALMIN,VETOR(I))
10      CONTINUE
        ORDEM(K)=VALMIN
C
C     COLOCAR VALOR 32(FLAG) NA POSICAO DO VALOR MINIMO EM VETOR
C
        DO 20 I=1,INDE
        IF(VALMIN.NE.VETOR(I)) GOTO 20
        VETOR(I)=32
        GOTO 5
20      CONTINUE
5       CONTINUE
C
C     IDENTIFICAR VALOR MEDIANO
C
        CRECNT=ORDEM(INDK)
C
        RETURN
        END
```

```
                    SUBROUTINE GRADIENT (DISPLAY)
**************************************************
*     REALIZA A OPERACAO GRADIENTE DE ROBERTS
**************************************************
C
        IMPLICIT INTEGER*2 (C-Z)
        CHARACTER NOME*14,STRING,STRIND,SUBMENU*20,F(3)*32
        COMMON IA(64,64),IB(64,64),IC(64,64),NOME,/COMUNIC/ STRING,STRIND
C
        DATA SUBMENU /'GRADIENTE DE ROBERTS'/
        DATA F(1) /'F1 Aplicacao direta do gradiente'/
        DATA F(2) /'F2 Gradiente com fundo definido'/
        DATA F(3) /'F3 Imagem gradiente binaria'/
C
C     APRESENTA OPCOES
C
10      STATUS=VCLRWK(DISPLAY)
        CALL MCURSOR(DISPLAY,SUBMENU,20)
C
        I=1
        DO 6 J=10,14,2
        STATUS=VCURAD(DISPLAY,J,10)
        STATUS=VCTXTS(DISPLAY,32,F(I))
6       I=I+1
C
        CALL FINAL(DISPLAY,2)
C
```

```
C     LER OPCAO DE ENTRADA
C
        STATUS=VRQCHC(DISPLAY,CHAIN,CHAIN)
        IF(STATUS.EQ.0) CALL MENUP(DISPLAY)
        IF(CHAIN.EQ.1) THEN
        CALL PISCA(10,10,F(1),5,DISPLAY,32,#10)
        IF(STRING.EQ.'1') CALL PPONTOG(IA,IC,DISPLAY,1)
        IF(STRING.EQ.'2') CALL PPONTOG(IB,IC,DISPLAY,2)
      ENDIF
C
        IF(CHAIN.EQ.2) THEN
        CALL PISCA(12,10,F(2),5,DISPLAY,31,#10)
        IF(STRING.EQ.'1') CALL COMFUNDO(IA,IC,DISPLAY,1)
        IF(STRING.EQ.'2') CALL COMFUNDO(IB,IC,DISPLAY,2)
      ENDIF
C
        IF(CHAIN.EQ.3) THEN
        CALL PISCA(14,10,F(3),5,DISPLAY,27,#10)
        IF(STRING.EQ.'1') CALL BINARIA(IA,IC,DISPLAY,1)
        IF(STRING.EQ.'2') CALL BINARIA(IB,IC,DISPLAY,2)
      ENDIF
C       STATUS=VENCUR(DISPLAY)
C
        IF(CHAIN.NE.10) GOTO 10
C
        END

              SUBROUTINE DETECAO(DISPLAY)
*****************************************************************
*     CHAMA ROTINAS QUE REALIZAM DETECAO DE BORDAS
*****************************************************************
C
        IMPLICIT INTEGER#2(C-Z)
        CHARACTER NOME#14,SUBMENU#17,STRING,STRIND,F(4)#25
        COMMON IA(64,64),IB(64,64),IC(64,64),NOME,/COMUNIC/STRING,STRIND
C
        DATA SUBMENU /'DETECAO DE BORDAS'/
        DATA F(1) /'F1 Gradiente de Roberts'/
        DATA F(2) /'F2 Operador de Sobel'/
        DATA F(3) /'F3 Operador de Prewitt'/
        DATA F(4) /'F4 Operadores direcionais'/
C
C     APRESENTA OPCOES
C
10      STATUS=VCLRWK(DISPLAY)
        CALL MCURSOR(DISPLAY,SUBMENU,17)
C
        I=1
        DO 6 J=10,16,2
        STATUS=VCURAD(DISPLAY,J,10)
        STATUS=VCTXTS(DISPLAY,25,F(I))
6       I=I+1
C
c       STATUS=VCURAD(DISPLAY,24,2)
c       STATUS=VCTXTS(DISPLAY,14,'OPCAO (F1..F5)')
        CALL FINAL(DISPLAY,2)
C
C     LER OPCAO
C
        STATUS=VRQCHC(DISPLAY,CHAIN,CHAIN)
```

```
        IF(STATUS.EQ.0) CALL MENUP(DISPLAY)
        IF(CHAIN.EQ.1) CALL GRADIENT(DISPLAY)
        IF(CHAIN.EQ.2) THEN
        CALL PISCA (12,10,F(2),5,DISPLAY,20,*10)
        IF(STRING.EQ.'1') CALL SOBEL(IA,IC,DISPLAY,1)
        IF(STRING.EQ.'2') CALL SOBEL(IB,IC,DISPLAY,2)
      ENDIF
        IF(CHAIN.EQ.3) THEN
        CALL PISCA (14,10,F(3),5,DISPLAY,22,*10)
        IF(STRING.EQ.'1') CALL PREWITT(IA,IC,DISPLAY,1)
        IF(STRING.EQ.'2') CALL PREWITT(IB,IC,DISPLAY,2)
      ENDIF
        IF(CHAIN.EQ.4) CALL DIREC1(DISPLAY)
        IF(CHAIN.NE.10) GOTO 10
C
        RETURN
        END
C-----
        SUBROUTINE DIREC1(DISPLAY)
C*************************************************************
C     CHAMA ROTINAS QUE REALIZAM AS OPERACOES DIRECIONAIS
C*************************************************************
C
        IMPLICIT INTEGER*2(C-Z)
        CHARACTER NOME*14,STRING,STRIND,SUBMENU*22,F(4)*34
        COMMON IA(64,64),IB(64,64),IC(64,64),NOME,/COMUNIC/STRING,STRIND
        DATA SUBMENU /'OPERADORES DIRECIONAIS'/
        DATA F(1) /'F1 Mascaras direcionais de Prewitt'/
        DATA F(2) /'F2 Mascaras direcionais de Kirsch'/
        DATA F(3) /'F3 Mascaras simples de 3 niveis'/
        DATA F(4) /'F4 Mascaras simples de 5 niveis'/
C
C     APRESENTA OPCOES
C
10      STATUS=VCLRWK(DISPLAY)
        CALL MCURSOR(DISPLAY,SUBMENU,22)

C
        I=1
        DO 6 J=10,16,2
        STATUS=VCURAD(DISPLAY,J,10)
        STATUS=VCTXTS(DISPLAY,34,F(I))
6       I=I+1
C
        CALL FINAL(DISPLAY,2)
C
C     LER OPCAO
C
        STATUS=VRQCHC(DISPLAY,CHAIN,CHAIN)
        IF(STATUS.EQ.0) CALL MENUP(DISPLAY)
        IF(CHAIN.EQ.1) THEN
        CALL PISCA (10,10,F(1),5,DISPLAY,34,*10)
        IF(STRING.EQ.'1') CALL PREWFIL(IA,IC,DISPLAY,1)
        IF(STRING.EQ.'2') CALL PREWFIL(IB,IC,DISPLAY,2)
      ENDIF
        IF(CHAIN.EQ.2) THEN
        CALL PISCA (12,10,F(2),5,DISPLAY,33,*10)
        IF(STRING.EQ.'1') CALL KIRSFIL(IA,IC,DISPLAY,1)
        IF(STRING.EQ.'2') CALL KIRSFIL(IB,IC,DISPLAY,2)
      ENDIF
```

```
        IF(CHAIN.EQ.3) THEN
        CALL PISCA(14,10,F(3),5,DISPLAY,31,*10)
        IF(STRING.EQ.'1') CALL MASC3FIL(IA,IC,DISPLAY,1)
        IF(STRING.EQ.'2') CALL MASC3FIL(IB,IC,DISPLAY,2)
      ENDIF
        IF(CHAIN.EQ.4) THEN
        CALL PISCA(16,10,F(4),5,DISPLAY,31,*10)
        IF(STRING.EQ.'1') CALL MASC5FIL(IA,IC,DISPLAY,1)
        IF(STRING.EQ.'2') CALL MASC5FIL(IB,IC,DISPLAY,2)
      ENDIF
C       STATUS=VENCUR(DISPLAY)
        IF(CHAIN.NE.10) GOTO 10
C
        RETURN
        END

       SUBROUTINE SUAVI(DISPLAY)
*****************************************************************
*     CHAMA ROTINAS QUE EXECUTAM OS PROCEDIMENTOS DE SUAVIZACAO
*     ESPACIAL
*****************************************************************
C
        IMPLICIT INTEGER*2 (C-Z)
        CHARACTER NOME*14,STRING,STRIND,SUBMENU*10,F(6)*70
        COMMON IA(64,64),IB(64,64),IC(64,64),NOME,/COMUNIC/STRING,STRIND
        DATA SUBMENU /'SUAVIZACAO'/
        DATA F(1) /'F1 Filtro da Media'/
        DATA F(2) /'F2 Filtro da Ordem'/
        DATA F(3) /'F3 Suaviz. com Vizinhanca Selecionada por Variancia'/
        DATA F(4) /'F4 Suaviz. com Vizinhanca Selecionada por Soma de Dife
     -rencas Absolutas'/
        DATA F(5) /'F5 Filtro da Media com os k-vizinhos mais proximos'/
        DATA F(6) /'F6 Filtro Sigma'/
C
C     APRESENTA OPCOES
C
10      STATUS=VCLRWK(DISPLAY)
        CALL MCURSOR(DISPLAY,SUBMENU,10)
C
        I=1
        DO 6 J=10,20,2
        STATUS=VCURAD(DISPLAY,J,10)
        STATUS=VCTXTS(DISPLAY,70,F(I))
6       I=I+1
C
        CALL FINAL(DISPLAY,2)
C
C     LER OPCAO DE ENTRADA
C
        STATUS=VRQCHC(DISPLAY,CHAIN,CHAIN)
        IF(STATUS.EQ.0) CALL MENUP(DISPLAY)
        IF(CHAIN.EQ.1) THEN
        CALL PISCA(10,10,F(1),5,DISPLAY,18,*10)
        CALL MOLDURA(DISPLAY)
        IF(STRING.EQ.'1') CALL MEDIAFIL(IA,IC,DISPLAY,1)
        IF(STRING.EQ.'2') CALL MEDIAFIL(IB,IC,DISPLAY,2)
      ENDIF
        IF(CHAIN.EQ.2) THEN
        CALL PISCA (12,10,F(2),5,DISPLAY,18,*10)
```

```
        IF(STRING.EQ.'1') CALL ORDEMFIL(IA,IC,DISPLAY,1)
        IF(STRING.EQ.'2') CALL ORDEMFIL(IB,IC,DISPLAY,2)
      ENDIF
        IF(CHAIN.EQ.3) THEN
        CALL PISCA (14,10,F(3),5,DISPLAY,51,#10)
        CALL MOLDURA(DISPLAY)
        IF(STRING.EQ.'1') CALL SVSVFIL(IA,IC,DISPLAY,1)
        IF(STRING.EQ.'2') CALL SVSVFIL(IB,IC,DISPLAY,2)
      ENDIF
        IF(CHAIN.EQ.4) THEN
        CALL PISCA(16,10,F(4),5,DISPLAY,70,#10)
        CALL MOLDURA(DISPLAY)
        IF(STRING.EQ.'1') CALL SSDAFIL(IA,IC,DISPLAY,1)
        IF(STRING.EQ.'2') CALL SSDAFIL(IB,IC,DISPLAY,2)
      ENDIF
        IF(CHAIN.EQ.5) THEN
        CALL PISCA (18,10,F(5),5,DISPLAY,50,#10)
        IF(STRING.EQ.'1') CALL KVIZFIL(IA,IC,DISPLAY,1)
        IF(STRING.EQ.'2') CALL KVIZFIL(IB,IC,DISPLAY,2)
      ENDIF
        IF(CHAIN.EQ.6) THEN
        CALL PISCA (20,10,F(6),5,DISPLAY,15,#10)
        IF(STRING.EQ.'1') CALL SIGMA(IA,IC,DISPLAY,1)
        IF(STRING.EQ.'2') CALL SIGMA(IB,IC,DISPLAY,2)
      ENDIF
        IF(CHAIN.NE.10) GOTO 10
C
        RETURN
        END

              SUBROUTINE COMPLET(IC,IA,FLAG,DISPLAY,FLAG1)
*************************************************************************
*     PREENCHE A MEMORIA DE IMAGEM MI3 APOS UMA OPERACAO DE FILTRAGEM
*************************************************************************
C
        IMPLICIT INTEGER*2 (C-Z)
        DIMENSION IC(64,64),IA(64,64),XY(4)
C
        PXIM=17500
        PYIM=3710
        GOTO (1,2) FLAG
C
C     COMPLETAR PARTE DA COLUNA 2 DA MATRIZ DE IMAGEM
C
2       XY(1)=PXIM-210
        XY(3)=XY(1)+210
        XY(2)=PYIM+13440-210+210
        DO 5 I=3,62
        XY(2)=XY(2)-210
        XY(4)=XY(2)+210
        IC(I,2)=IC(I,3)
        CALL IMAGEP(IC(I,2),XY,DISPLAY)
5       CONTINUE
C
C     COMPLETAR PARTE DA COLUNA 63 DA MATRIZ
C
        XY(1)=PXIM+13440-210-210-210-210
        XY(3)=XY(1)+210
        XY(2)=PYIM+13440-210+210
```

```
      DO 10 I=3,62
      XY(2)=XY(2)-210
      XY(4)=XY(2)+210
      IC(I,63)=IC(I,62)
      CALL IMAGEP(IC(I,63),XY,DISPLAY)
10    CONTINUE
C
C     COMPLETAR TODA A LINHA 2 DA MATRIZ
C
      XY(1)=PXIM-210-210
      XY(2)=PYIM+13440-210+210
      XY(4)=XY(2)+210
      DO 15 I=2,63
      XY(1)=XY(1)+210
      XY(3)=XY(1)+210
      IC(2,I)=IC(3,I)
      CALL IMAGEP(IC(2,I),XY,DISPLAY)
15    CONTINUE
C
C     COMPLETAR TODA A LINHA 63 DA MATRIZ
C
      XY(1)=PXIM-210-210
      XY(2)=PYIM+210+210
      XY(4)=XY(2)+210
      DO 20 I=2,63
      XY(1)=XY(1)+210
      XY(3)=XY(1)+210
      IC(63,I)=IC(62,I)
      CALL IMAGEP(IC(63,I),XY,DISPLAY)
20    CONTINUE
C
C
C     COMPLETAR PARTE DA COLUNA 1 DA MATRIZ
C
1     IF(FLAG.NE.2) GOTO 21
      PXIM=PXIM-210
      PYIM=PYIM+210
21    XY(1)=PXIM-210
      XY(3)=XY(1)+210
      XY(2)=PYIM+13440-210+210
      DO 25 I=2,63
      XY(2)=XY(2)-210
      XY(4)=XY(2)+210
      IC(I,1)=IC(I,2)
      CALL IMAGEP(IC(I,1),XY,DISPLAY)
25    CONTINUE
C
C     COMPLETAR PARTE DA COLUNA 64 DA MATRIZ
C
      XY(1)=PXIM+13440-210-210
      XY(3)=XY(1)+210
      XY(2)=PYIM+13440-210+210
      DO 30 I=2,63
      XY(2)=XY(2)-210
      XY(4)=XY(2)+210
      IC(I,64)=IC(I,63)
      CALL IMAGEP(IC(I,64),XY,DISPLAY)
30    CONTINUE
C
```

```
C     COMPLETAR TODA A LINHA 1 DA MATRIZ
C
        XY(1)=PXIM-210-210
        XY(2)=PYIM+13440-210+210
        XY(4)=XY(2)+210
        DO 35 I=1,64
        XY(1)=XY(1)+210
        XY(3)=XY(1)+210
        IC(1,I)=IC(2,I)
        CALL IMAGEP(IC(1,I),XY,DISPLAY)
35      CONTINUE
C
C     COMPLETAR TODA A LINHA 64 DA MATRIZ
C
        XY(1)=PXIM-210-210
        XY(2)=PYIM+210
        XY(4)=XY(2)+210
        DO 40 I=1,64
        XY(1)=XY(1)+210
        XY(3)=XY(1)+210
        IC(64,I)=IC(63,I)
        CALL IMAGEP(IC(64,I),XY,DISPLAY)
40      CONTINUE
        CALL CRFIM(DISPLAY)
        RETURN

        END

              SUBROUTINE SVSVFIL (IA,IC,DISPLAY,FLAG)
*************************************************************************
*     REALIZA A FILTRAGEM ESPACIAL UTILIZANDO O FILTRO DE SUAVIZACAO
*     COM VIZINHANCA SELECIONADA POR VARIANCIA (Nagao e Matsuyama)
*************************************************************************
C
        IMPLICIT INTEGER*2 (C-Z)
        DIMENSION JANELA(5,5),IA(64,64),IC(64,64),XY(4)
        DATA PXIM,PYIM /17500,3500/
C
        STATUS=VGTXTS(DISPLAY,10000,20810,11,'Filtro SVSV')
        CALL IMAGEM(IA,'Original      ',DISPLAY,2000,4110,0,FLAG)
        STATUS=VGTXTS(DISPLAY,PXIM+2000,PYIM+14700,10,'Processada')
C
C     COMPOR MATRIZ_JANELA A PARTIR DA MEMORIA DE IMAGEM
C
        XY(2)=PYIM+13440+210
        DO 5 I=1,60
        XY(1)=PXIM-210
        XY(2)=XY(2)-210
        XY(4)=XY(2)+210
        DO 5 J=1,60
        XY(1)=XY(1)+210
        XY(3)=XY(1)+210
        IW=I+4
        JW=J+4
        LINHA=0
C
C     DEFINE TAMANHO DA JANELA: 5*5
C
        DO 10 IF=I,IW
        LINHA=LINHA+1
        COLUNA=0
```

```
      DO 10 JF=J,JW
      COLUNA=COLUNA+1
10    JANELA(LINHA,COLUNA)=IA(IF,JF)
C
C     CHAMAR FUNCTION QUE REALIZA A FILTRAGEM
C
      PONTOP=SVSV(JANELA)
C
C     ARMAZENAR VALOR NA MEMORIA DE IMAGEM (MI3)
C
      IC(I+2,J+2)=PONTOP
      CALL IMAGEP(PONTOP,XY,DISPLAY)
C
5     CONTINUE
C
      CALL COMPLET(IC,IA,2,DISPLAY,FLAG)
C
      RETURN
      END
C
C
C
      INTEGER*2 FUNCTION SVSV(JANELA)
*********************************************************************
*     REALIZA A FILTRAGEM DE SUAVIZACAO SVSV E RETORNA COM O VALOR DO
*     PONTO CENTRAL DA JANELA 5*5 A SER SUBSTITUIDO NA IMAGEM
*********************************************************************
C
      IMPLICIT INTEGER*2 (C-Z)
      DIMENSION JANELA(5,5),VIZINHA(9,9),CTES(9,6),VARIANCIA(9),MEDIA(9)
      DATA CTES /2*1,2*3,1,2*2,3,2,1,3,1,3,2,1,3,2*2,2*1,3*3,2*2,
     .           3,0,2*3,1,5,2,2*3,2,0,2*3,2*5,3,2*4,3,0,1,5,2*3,
     .           4,2*3,4,0/
C
C     FORMAR TABELA COM OS PIXELS DE CADA VIZINHANCA DA JANELA
C
      DO 1 NLINHA=1,9
      COLUNA=0
      IF=CTES(NLINHA,1)
      JF=CTES(NLINHA,2)
      IS=IF+2
      JS=JF+2
C
      DO 5 I=IF,IS
      DO 5 J=JF,JS
      IF((I.EQ.CTES(NLINHA,3).AND.J.EQ.CTES(NLINHA,4)).OR.
     .   (I.EQ.CTES(NLINHA,5).AND.J.EQ.CTES(NLINHA,6))) GOTO 5
      COLUNA=COLUNA+1
      VIZINHA(NLINHA,COLUNA)=JANELA(I,J)
5     CONTINUE
C
C     CALCULAR A MEDIA DOS PIXELS
C
      IMEDIA=0
      IF=7
      IF(NLINHA.EQ.9) IF=9
      DO 10 I=1,IF
10    IMEDIA=IMEDIA+VIZINHA(NLINHA,I)
      MEDIA(NLINHA)=IMEDIA/IF +.5
```

```
C
C     CALCULO DA VARIANCIA
C
        VARIANCIA(NLINHA)=0
        DO 15 I=1,IF
15      VARIANCIA(NLINHA)=VARIANCIA(NLINHA)+(VIZINHA(NLINHA,I)-
                        MEDIA(NLINHA))**2
C
1       CONTINUE
C
C     ENCONTRAR A VIZINHANCA QUE APRESENTA A MENOR VARIANCIA
C
        NVIZIN=MENOR(VARIANCIA)
C
C     CALCULAR A MEDIA DA VIZINHANCA COM MENOR DISPERSAO
C
        SVSV=MEDIA(NVIZIN)
C
        RETURN
        END
              SUBROUTINE SSDAFIL(IA,IC,DISPLAY,FLAG)
*************************************************************************
*     REALIZA A FILTRAGEM ESPACIAL UTILISANDO O FILTRO DE SUAVIZACAO
*     COM VIZINHANCA SELECIONADA POR SOMA DE DIFERENCAS ABSOLUTAS
*************************************************************************
        IMPLICIT INTEGER*2 (C-Z)
        DIMENSION JANELA(5,5),IA(64,64),IC(64,64),XY(4)
        DATA PXIM,PYIM /17500,3500/
C
        STATUS=VGTXTS(DISPLAY,10000,20810,11,'Filtro SSDA')
        CALL IMAGEM(IA,'Original       ',DISPLAY,2000,4110,0,FLAG)
        STATUS=VGTXTS(DISPLAY,PXIM+2000,PYIM+14700,10,'Processada')
C
C     COMPOR MATRIZ_JANELA A PARTIR DA MEMORIA DE IMAGEM:
        XY(2)=PYIM+13440+210
        DO 5 I=1,60
        XY(1)=PXIM-210
        XY(2)=XY(2)-210
        XY(4)=XY(2)+210
        DO 5 J=1,60
        XY(1)=XY(1)+210
        XY(3)=XY(1)+210
        IW=I+4
        JW=J+4
        LINHA=0
C
C     DEFINE TAMANHO DA JANELA: 5*5
C
        DO 10 IF=I,IW
        LINHA=LINHA+1
        COLUNA=0
        DO 10 JF=J,JW
        COLUNA=COLUNA+1
10      JANELA(LINHA,COLUNA)=IA(IF,JF)
C
C     CHAMAR FUNCTION QUE REALIZA A FILTRAGEM
C
        PONTOP=SSDA(JANELA)
C
C     ARMAZENAR VALOR NA MEMORIA DE IMAGEM (MI3)
C
```

```
      IC(I+2,J+2)=PONTOP
      CALL IMAGEP(PONTOP,XY,DISPLAY)
C
5     CONTINUE
C   COMPLETAR MEMORIA IC
C
      CALL COMPLET(IC,IA,2,DISPLAY,FLAG)
C
      RETURN
      END
C
C
C
      INTEGER*2 FUNCTION SSDA(JANELA)
*************************************************************************
*     REALIZA A FILTRAGEM DE SUAVIZACAO SSDA E RETORNA COM O VALOR DO
*      PONTO CENTRAL DA JANELA 5*5 A SER SUBSTITUIDO NA IMAGEM
*************************************************************************
C
      IMPLICIT INTEGER*2 (C-Z)
      DIMENSION JANELA(5,5),ELEM9(9,9),INDICE(9)

C
C     FORMAR TABELA COM OS PIXELS DE CADA VIZINHANCA
C
      LINHA=1
      DO 5 I=1,3
      DO 5 J=1,3
      IW=I+2
      JW=J+2
      COLUNA=0
      DO 10 IF=I,IW
      DO 10 JF=J,JW
      COLUNA=COLUNA+1
10    ELEM9(LINHA,COLUNA)=JANELA(IF,JF)
      LINHA=LINHA+1
5     CONTINUE
C
C     CALCULAR O INDICE DE HOMOGENEIDADE DE CADA VIZINHANCA
C
      J=JANELA(3,3)
      DO 15 LINHA=1,9
      SSDA=0
      DO 20 I=1,9
      SSDA=SSDA+IABS(ELEM9(LINHA,I)-J)
20    CONTINUE
      INDICE(LINHA)=SSDA
15    CONTINUE
C
C     DETERMINAR A VIZINHANCA QUE APRESENTA O MENOR INDICE DE
C     HOMOGENEIDADE
C
      NVIZIN=MENOR(INDICE)
C
C     CALCULAR A MEDIA DOS PIXELS CUJA VIZINHANCA APRESENTA
C     O MENOR INDICE DE HOMOGENEIDADE
C
      SSDA=0
```

```
      DO 35 J=1,9
35    SSDA=SSDA+ELEM9(NVIZIN,J)
      SSDA=SSDA/9 +.5
C
      RETURN
      END
C
C
      INTEGER*2 FUNCTION MENOR(INDICE)
*****************************************************************
*     DETERMINA QUAL DAS VIZINHANCAS APRESENTA MAIOR INDICE DE
*     HOMOGENEIDADE OU A MENOR VARIANCIA
*****************************************************************
C
      IMPLICIT INTEGER*2 (C-Z)
      DIMENSION INDICE(9)
C
      COMPARE=INDICE(1)
      MENOR=1
      DO 30 I=2,9
      IF(INDICE(I).GE.COMPARE) GOTO 30
      MENOR=I
      COMPARE=INDICE(I)
30    CONTINUE
C
      RETURN
      END

      SUBROUTINE MASC3FIL (IA,IC,DISPLAY,FLAG)
*****************************************************************
*     REALIZA A OPERACAO DIRECIONAL UTILIZANDO AS MASCARAS DE
*     3-NIVEIS
*****************************************************************
C
      IMPLICIT INTEGER*2 (C-Z)
      DIMENSION MASC3(4,9),IA(64,64),IC(64,64)
C
      DATA MASC3 /3*1,0,2*1,0,-1,1,0,2*-1,0,3*1,5*0,4*-1,0,2*1,
     .           2*-1,0,1,3*-1,0/
C
C    REALIZAR PROCEDIMENTO
C
      CALL MOLDURA(DISPLAY)
      STATUS=V8TXTS(DISPLAY,10000,20810,17,'Mascaras 3-niveis')
      CALL OPDIREC(MASC3,IA,IC,4,DISPLAY,FLAG)
C
      RETURN
      END

      SUBROUTINE MASC5FIL (IA,IC,DISPLAY,FLAG)
*****************************************************************
*     REALIZA A OPERACAO DIRECIONAL UTILIZANDO A MASCARA
*     5-NIVEIS
*****************************************************************
C
      IMPLICIT INTEGER*2 (C-Z)
      DIMENSION MASC5(4,9),IA(64,64),IC(64,64)
C
      DATA MASC5 /1,2,1,0,2,1,0,-1,1,0,-1,-2,0,1,2,1,5*0,-1,-2,2*-1,
     .           0,1,2,-2,-1,0,1,-1,-2,-1,0/
C
```

```
      CALL MOLDURA(DISPLAY)
      STATUS=V6TXTS(DISPLAY,10000,20810,17,'Mascaras 5-niveis')
      CALL OPDIREC(MASC5,IA,IC,4,DISPLAY,FLAG)
C
      RETURN
      END
            SUBROUTINE KIRSFIL (IA,IC,DISPLAY,FLAG)
*****************************************************************
*     REALIZA A OPERACAO DIRECIONAL UTILIZANDO AS MASCARAS DE
*     KIRSCH
*****************************************************************
C
      IMPLICIT INTEGER*2 (C-Z)
      DIMENSION MASCKIR(8,9),IA(64,64),IC(64,64)
      DATA MASCKIR /3*5,5*-3,2*5,5*-3,2*5,5*-3,2*5,-3,3*5,4*-3,8*0,
     .     5*-3,3*5,2*-3,3*5,6*-3,3*5,6*-3,3*5,-3/
C
      CALL MOLDURA(DISPLAY)
      STATUS=V6TXTS(DISPLAY,10000,20810,18,'Mascaras de Kirsch')
      CALL OPDIREC (MASCKIR,IA,IC,8,DISPLAY,FLAG)
C
      RETURN
      END
            SUBROUTINE PREWFIL(IA,IC,DISPLAY,FLAG)
*****************************************************************
*     REALIZA A OPERACAO DIRECIONAL UTILIZANDO AS MASCARAS
*     DE PREWITT
*****************************************************************
C
      IMPLICIT INTEGER*2 (C-Z)
      DIMENSION MASCPREW (8,9),IA(64,64),IC(64,64)
C
      DATA MASCPREW /4*1,3*-1,4*1,3*-1,4*1,3*-1,8*1,3*-1,8*-2,1,3*-1,
     .              4*1,-1,5*1,4*-1,5*1,4*-1,5*1/
C
      CALL MOLDURA(DISPLAY)
      STATUS=V6TXTS(DISPLAY,10000,20810,19,'Mascaras de Prewitt')
      CALL OPDIREC(MASCPREW,IA,IC,8,DISPLAY,FLAG)
C
      RETURN
      END
            SUBROUTINE SOBEL (IA,IC,DISPLAY,FLAG)
*****************************************************************
*     APLICA AS DUAS MASCARAS DOS OPERADORES DE SOBEL
*     PARA A DETERMINACAO DO GRADIENTE DA IMAGEM
*****************************************************************
C
      IMPLICIT INTEGER*2(C-Z)
      DIMENSION MSOBEL(2,9),IA(64,64),IC(64,64)
C
      DATA MSOBEL /2*1,2,0,1,-1,0,2,3*0,-2,-1,1,-2,0,2*-1/
C
      CALL MOLDURA(DISPLAY)
      STATUS=V6TXTS(DISPLAY,10000,20810,17,'Operador de Sobel')

      CALL SOBPREW (IA,IC,MSOBEL,DISPLAY,FLAG)
C
      RETURN
      END
C
```

```
C
      SUBROUTINE SOBPREW (IA,IC,SP,DISPLAY,FLAG)
*************************************************
*     REALIZA OPERACOES DE SOBEL E PREWITT
*************************************************
C
      IMPLICIT INTEGER*2 (C-Z)
      DIMENSION ACONVOL(2),SP(2,9),IA(64,64),IC(64,64),XY(4)
      DATA PXIM,PYIM /17500,3500/
C
      CALL IMAGEM(IA,'Original       ',DISPLAY,2000,3900,0,FLAG)
      STATUS=VGTXTS(DISPLAY,PXIM+2000,PYIM+14700,10,'Processada')
C
      XY(2)=PYIM+13440+210
      DO 5 X=2,63
      XY(1)=PXIM-210
      XY(2)=XY(2)-210
      XY(4)=XY(2)+210
      DO 5 Y=2,63
      XY(1)=XY(1)+210
      XY(3)=XY(1)+210
      DO 10 K=1,2
      ACONVOL(K)=IA(X-1,Y-1)*SP(K,9)+IA(X-1,Y)*SP(K,8)+
     .           IA(X-1,Y+1)*SP(K,7)+IA(X,Y-1)*SP(K,6)+IA(X,Y)*SP(K,5)+
     .           IA(X,Y+1)*SP(K,4)+IA(X+1,Y-1)*SP(K,3)+
     .           IA(X+1,Y)*SP(K,2)+IA(X+1,Y+1)*SP(K,1)
10    CONTINUE
C
C     DETERMINAR A MAGNITUDE DO GRADIENTE
C
      PONTOP=SQRT(ACONVOL(1)**2+ACONVOL(2)**2)
C
      IF(PONTOP.GT.31) PONTOP=31
      IC(X,Y)=PONTOP
C
      CALL IMAGEP(PONTOP,XY,DISPLAY)
C
5     CONTINUE
C
C     COMPLETAR MEMORIA DE IMAGEM
C
      CALL COMPLET (IC,IA,1,DISPLAY,FLAG)
C
      RETURN
      END

                              SUBROUTINE PREWITT (IA,IC,DISPLAY,FLAG)
*************************************************************
*     APLICA AS DUSA MASCARAS DE PREWITT PARA DETERMINAR O
*     GRADIENTE DA IMAGEM
*************************************************************
C
      IMPLICIT INTEGER*2 (C-Z)
      DIMENSION MPREWITT(2,9),IA(64,64),IC(64,64)
C
      DATA MPREWITT /2*1,1,0,1,-1,0,1,3*0,2*-1,1,-1,0,2*-1/
C
      CALL MOLDURA(DISPLAY)
      STATUS=VGTXTS(DISPLAY,10000,20810,19,'Operador de Prewitt')
      CALL SOBPREW (IA,IC,MPREWITT,DISPLAY,FLAG)
```

```
C
      RETURN
      END

            SUBROUTINE EQHIST (IA,IC,DISPLAY,FLAG)
C*************************************************************
C
C     REALIZA A EQUALIZACAO HISTOGRAMICA EM UMA IMAGEM
C
C*************************************************************
C
      IMPLICIT INTEGER*2 (C-Z)
      CHARACTER STRING*2
      REAL SUM
      DIMENSION IA(64,64),IC(64,64),NP(32),NS(32),XY(4)
      DATA PXIM,PYIM/17500,3400/
C
      CALL MOLDURA(DISPLAY)
      STATUS=VGTXTS(DISPLAY,7000,20810,24,'Equalizacao Histogramica')
      CALL IMAGEM(IA,'Original      ',DISPLAY,2000,3600,0,FLAG)
      STATUS=VGTXTS(DISPLAY,PXIM+2000,PYIM+14400,10,'Processada')
C
15    CALL CONTA3(-2,0,IA,NP)
      DO 20 K=1,32
      NS(K)=SUM(K,NP)*0.0076 +0.5
20    CONTINUE
C
      XY(2)=PYIM+13440+210
      DO 40 J=1,64
      XY(1)=PXIM-210
      XY(2)=XY(2)-210
      XY(4)=XY(2)+210
      DO 40 K=1,64
      XY(1)=XY(1)+210
      XY(3)=XY(1)+210
      IC(J,K)=NS(IA(J,K)+1)
      CALL IMAGEP(IC(J,K),XY,DISPLAY)
40    CONTINUE
      CALL CRFIM(DISPLAY)
      RETURN
      END
C
      REAL*4 FUNCTION SUM (K,NP)
      INTEGER*2 K,NP,J
      DIMENSION NP(32)
      SUM=0.
      DO 10 J=1,K
      SUM=SUM+NP(J)
10    CONTINUE
      RETURN
      END
      SUBROUTINE REALIZ (DISPLAY,IA,IC,NOME,FUNC)
C*************************************************************
C    A SUBROTINA ABAIXO REALIZA AS FUNCOES REFERENTES AO MAPEAMENTO
C    DIRETO DOS NIVEIS DE CINZA
C*************************************************************
C
      IMPLICIT INTEGER*2 (C-Z)
```

```
      CHARACTER NOME*14
      DIMENSION IA(64,64),IC(64,64),ICAIX(32),MIAPAB(4)
      COMMON/INFO/ECXY(2)
      DATA X,Y/18200,6000/
      INTV=ECXY(1)
      INT1=ECXY(2)
      GOTO (1,2,3,4,5,6,7,8) FUNC
1     DO 10 I=1,64
      DO 10 J=1,64
      IC(I,J)=0
      IF(IA(I,J).GE.ECXY(1)) IC(I,J)=31
10    CONTINUE
      GOTO 100
C
2     INTV=ECXY(2)-ECXY(1)
      INT1=ECXY(1)
      DO 20 I=0,31
      ICAIX(I)=.5+INT1+I*INTV/31
20    CONTINUE
      GOTO 99
C
3     DO 30 I=0,10
30    ICAIX(I)=I/2
      DO 31 I=11,21
31    ICAIX(I)=(2*I-15)
      DO 32 I=22,31
32    ICAIX(I)=I/2 +15
      GOTO 99
C
4     DO 40 I=0,INTV
40    ICAIX(I)=I
      DO 41 I=INT1,31
41    ICAIX(I)=I
      INTV=INTV+1
      INT1=INT1+1
      DO 42 I=INTV,INT1
42    ICAIX(I)=31
      GOTO 99
C
5     DO 50 I=0,INTV
50    ICAIX(I)=0
      DO 51 I=INT1,31
51    ICAIX(I)=31
      INTV=INTV+1
      INT1=INT1-1
      DO 52 I=INTV,INT1
52    ICAIX(I)=(I-ECXY(1))*31/(ECXY(2)-ECXY(1))
      GOTO 99
C
6     DO 60 I=INTV,INT1
60    ICAIX(I)=31
      INTV=INTV-1
      INT1=INT1+1
      DO 61 I=0,INTV
61    ICAIX(I)=0
      DO 62 I=INT1,31
62    ICAIX(I)=0
      GOTO 99
C
```

```
7       DO 70 I=0,31
70      ICAIX(I)=31-I
        GOTO 99
C
8       DO 80 I=0,10
80      ICAIX(I)=3*I
        DO 81 I=11,20
81      ICAIX(I)=3*(I-10)
        DO 82 I=21,31
82      ICAIX(I)=3*(I-20)
C
C
99      DO 21 I=1,64
        DO 21 J=1,64
21      IC(I,J)=ICAIX(IA(I,J))
100     CALL IMAGEM (IC,NOME,DISPLAY,X,Y,0,3)
        MIAPAG(1)=X+64*210+50
        MIAPAG(2)=Y-200
        MIAPAG(3)=MIAPAG(1)+200
        MIAPAG(4)=Y+64*210+100
        STATUS=VCLARY(DISPLAY,MIAPAG,1,1,1,4,0)
        RETURN
        END
              SUBROUTINE OPDIREC (MASC,IA,IC,NMASC,DISPLAY,FLAG)
************************************************
*     REALIZA OPERACOES DIRECIONAIS
************************************************
C
        IMPLICIT INTEGER*2 (C-Z)
        DIMENSION IA(64,64),IC(64,64),MASC(NMASC,9),CONVOL(9),XY(4)
        DATA PXIM,PYIM /17500,3500/
C
        CALL IMAGEM(IA,'Original      ',DISPLAY,2000,3900,0,FLAG)
        STATUS=VGTXTS(DISPLAY,PXIM+2000,PYIM+14700,10,'Processada')
C
        XY(2)=PYIM+13440+210
        DO 5 X=2,63
        XY(1)=PXIM-210
        XY(2)=XY(2)-210
        XY(4)=XY(2)+210
        DO 5 Y=2,63
        XY(1)=XY(1)+210
        XY(3)=XY(1)+210
C
        DO 10 K=1,NMASC
        CONVOL(K)=IABS(IA(X-1,Y-1)*MASC(K,9)+IA(X-1,Y)*MASC(K,8)+
     .  IA(X-1,Y+1)*MASC(K,7)+IA(X,Y-1)*MASC(K,6)+IA(X,Y)*MASC(K,5)+
     .  IA(X,Y+1)*MASC(K,4)+IA(X+1,Y-1)*MASC(K,3)+IA(X+1,Y)*MASC(K,2)+
     .  IA(X+1,Y+1)*MASC(K,1))
C
10      CONTINUE
C
C     DETERMINAR A MAIOR RESPOSTA ENTRE AS MASCARAS DE CONVOLUCAO
C
        PONTOP=CONVOL(1)
        DO 15 K=2,NMASC
        IF(PONTOP.LT.CONVOL(K)) PONTOP=CONVOL(K)
15      CONTINUE
C
```

```
C     SUBSTITUIR A RESPOSTA MAXIMA NA MEMORIA DE IMAGEM (MI3)
C
        IF(PONTOP.GT.31) PONTOP=31
C
        IC(X,Y)=PONTOP
C
C     REPRESENTAR PIXEL NA TELA
C
        CALL IMAGEP(PONTOP,XY,DISPLAY)
C
5       CONTINUE
C
C     COMPLETAR MEMORIA DE IMAGEM
C
        CALL COMPLET(IC,IA,1,DISPLAY,FLAG)
C
        RETURN
        END
        SUBROUTINE HISTO(IA,NOME,DISPLAY,DN)
*********************************************************
* CHAMA SUBROTINAS ASSOCIADAS AO HISTOGRAMA DAS IMAGENS
*********************************************************
        IMPLICIT INTEGER*2 (C-Z)
        IMPLICIT REAL (A-B)
        CHARACTER*14 NOME
        DIMENSION IA(64,64),NP(64)
        CALL MOLDURA(DISPLAY)
        CALL IMAGEM(IA,NOME,DISPLAY,6000,6000,0,DN)
        CALL CONTA3(0,0,IA,NP)
        CALL GRAFO2(NP,DISPLAY,20000,8000,32)
        CALL ESTAT(NP,DISPLAY,2000)
        CALL CRFIM(DISPLAY)
        RETURN
        END

       SUBROUTINE VMDP(NP,AMED,ADP,MAX,MIN)
*********************************************************
*     CALCULA O VALOR MEDIO E O DESVIO PADRAO
*********************************************************
C
        IMPLICIT INTEGER*2 (C-Z)
        IMPLICIT REAL (A-B)
        DIMENSION NP(32)
        COMMON/PRAESTATLOC/AREA
C
        IF(AREA.EQ.0.) AREA=4096.
        AMED=0.
        ADP =0.0
        MAX =0
        MIN =31
        DO 10 I=1,32
        AMED=AMED+NP(I)*(I-1.)
10      CONTINUE
        AMED=AMED/AREA
        DO 20 I=1,32
        AA=ABS(I-1.-AMED)
        ADP=ADP+NP(I)*(AA**2.)
20      CONTINUE
        ADP=ADP/AREA
```

```
      ADP=SQRT(ADP)
      T1=0
      T2=0
      DO 30 I=1,32
      IF(NP(I).EQ.0 .OR.T1.EQ.1) GO TO 25
      MIN=I-1
      T1=1
25    IF(NP(33-I).EQ.0 .OR. T2.EQ.1) GO TO 30
      MAX=32-I
      T2=1
30    CONTINUE
      AREA=0.
      RETURN
      END
```

```
      SUBROUTINE DSP(Ib,DISPLAY,*)
*****************************************
*     REALIZA A IMPRESSAO DAS IMAGENS
*****************************************
C
C     OBS: ESTA SUBROTINAA E UMA VERSAO DO PROGRAMA FORNECIDO POR
C     GONZALES NO LIVRO DIGITAL IMAGE PROCESSING
C
      IMPLICIT INTEGER*2 (I-N,V)
      INTEGER*2 EINT,RETORNO
      integer*2 lev(32),blank(5),DISPLAY,STATUS
      character*1 line(128,5),gray(32,5),OPCAO(4)*15,TIPO(2)*8,ESCOLHA
      DIMENSION IA(64,64),IB(64,64),XYEND(2)
c
c     especifica caracteres para nivel de cinza
c
      DATA GRAY /6*'M',5*'H','X','H','X','O',
     1 'Z','W','M','N','O','S','=','I','*','+',
     2 '+','=',':','-','.','-',' ',6*'W',3*'#',
     3 '*','+','+',4*'-',5*' ','=',' ',' ',' ','-',
     4 3*' ','-',3*' ',4*'#','O','O','+','-',
     5 24*' ','O','O','O',29*' ','+',31*' '/
C
      OPCAO(1)='Linear'
      OPCAO(2)='Raiz quadratica'
      OPCAO(3)='Logaritmica'
      OPCAO(4)='Absorcao'
      TIPO(1)='Positiva'
      TIPO(2)='Negativa'
C
      DO 1 I=1,64
      DO 1 J=1,64
1     IA(I,J)=IB(65-J,I)
      IL=0
      IH=31
      nx=64
      ny=64
11    STATUS=VCURAD(DISPLAY,25,1)
      STATUS=VEREOL(DISPLAY)
      STATUS=VCURAD(DISPLAY,23,2)
      STATUS=VCTXTS(DISPLAY,63,'Escolha escala de translacao do nivel de
     . cinza                  ')
      STATUS=VCURAD(DISPLAY,24,2)
      STATUS=VCTXTS(DISPLAY,55,'F1-Linear F2-Raiz quadratica F3-Logarit
     .ica F4-Absorcao')
```

```
      INF=59
      NSUP=62
      ASSIGN 3 TO RETORNO
      GOTO 5
3     LAW=EINT-58
      STATUS=VCURAD(DISPLAY,23,2)
      STATUS=VCTXTS(DISPLAY,42,'ESCALA DE TRANSLACAO DO NIVEL DE CINZA:
     .')
      STATUS=VCTXTS(DISPLAY,15,OPCAO(LAW))
      STATUS=VCURAD(DISPLAY,24,2)
      STATUS=VCTXTS(DISPLAY,55,'Escolha tipo de imagem
     .              ')
      STATUS=VCURAD(DISPLAY,25,2)
      STATUS=VCTXTS(DISPLAY,23,'F1-Positiva F2-Negativa')
      INF=59
      NSUP=60
      ASSIGN 4 TO RETORNO
      GOTO 5
4     NEG=EINT-58
      STATUS=VCURAD(DISPLAY,25,2)
      STATUS=VEREOL(DISPLAY)
      STATUS=VCURAD(DISPLAY,24,2)
      STATUS=VCTXTS(DISPLAY,16,'TIPO DE IMAGEM: ')
      STATUS=VCTXTS(DISPLAY,8,TIPO(NEG))
      STATUS=VCURAD(DISPLAY,25,2)
      STATUS=VCTXTS(DISPLAY,23,'Iniciar impressao(S/N)?')
50    ESCOLHA=' '
      STATUS=VSMSTR(DISPLAY,1,0,XYEND,ESCOLHA)
      EINT=ICHAR(ESCOLHA)
      IF(EINT.EQ.27) RETURN 1
      IF(EINT.EQ.0.OR.ESCOLHA.NE.'S'.AND.ESCOLHA.NE.'s'.AND.ESCOLHA.NE.
     .'N'.AND.ESCOLHA.NE.'n') GOTO 50
      STATUS=VCTXTS(DISPLAY,1,ESCOLHA)
      IF(ESCOLHA.NE.'S'.AND.ESCOLHA.NE.'s') GOTO 11
C
C     REALIZR IMPRESSAO
C
      STATUS=VCURAD(DISPLAY,1,2)
      STATUS=VCTXTS(DISPLAY,20,'AGUARDE...IMPRIMINDO')
C
      gn=32.
      fl=IL
      fh=IH
      if((fh-fl).gt.0.0) go to 100
      t=fl
      fl=fh
      fh=t
100   range=(fh-fl+1)/gn
      aa=(sqrt(fh)-sqrt(fl))/gn
      ee=(fh-fl)/alog(gn+1.0)
      t=amax1(fl,1.0)
      ss=-(1.0/gn)*alog(fh/t)
c
c     o vetor lev e' computado a continuacao
c
      do 160 i=1,32
      go to (110,120,130,140),law
110   flev=fl+(i-1)*range+0.5
      go to 150
```

```
120     flev=(sqrt(fl)+(i-1)*aa)**2+0.5
        go to 150
130     flev=fl+ee*alog(float(i))+0.5
        go to 150
140     flev=fh*exp(ss*(gn-i))+0.5
150     lev(i)=flev
160     continue
        do 180 i=1,nx
        do 180 j=1,ny
        klt=1
        do 170 k=1,32
        if(ia(i,j).ge.lev(k)) klt=k
170     continue
        ib(i,j)=klt
180     continue
c
c       impressao
c
        OPEN(7,FILE='LPT1')
C       write(7,1)
        ix=nx
        iy=2*ny
        IR=1
        do 210 i=1,ix
        do 190 j=1,5
        blank(j)=0
190     continue
        do 200 k=2,iy,2
        j=k/2
        NG=ib(i,j)
        if(neg.eq.2) ng=33-ng
        do 200 l=1,5
        line(k-1,l)=gray(ng,l)
        line(k,l)=gray(ng,l)
        if(ng.ne.32)blank(l)=1
200     continue
        write(7,9)
        do 210 l=1,5
        if(blank(l).eq.0) go to 210
        write(7,10)(line(m,l),m=1,iy)
        IR=IR+1
210     continue
9       format(1H )
10      format(1H+,3x,128a1)
        close(7)
      DO 1000 I=1,64
      DO 1000 J=1,64
1000  IB(I,J)=IA(J,65-I)
        return
C
5       ESCOLHA=' '
        STATUS=VRQSTR(DISPLAY,1,0,XYEND,ESCOLHA)
        EINT=ICHAR(ESCOLHA)
        IF(EINT.EQ.27) RETURN 1
        IF(EINT.LT.INF.OR.EINT.GT.NSUP) GOTO 5
        GOTO RETORNO
C
        end
```

```
      SUBROUTINE CVI(DATA,S)
*************************************************************
* CONVERTE DADOS DE ENTRADA NO MODO GRAFICO PARA
* VALORES INTEIROS
*************************************************************
C
      IMPLICIT INTEGER*2 (C-Z)
      CHARACTER*1 S(5)
      DIMENSION X(5)
      DO 10 I=1,5
      S(I)=' '
10    CONTINUE
      X(1)=DATA/10000
      IF(X(1).GT.9) GO TO 1
      IF(X(1).NE.0) CALL TROCA(X(1),S,1)
      X(2)=DATA/1000-X(1)*10
      IF(X(1).NE.0 .OR. X(2).NE.0) CALL TROCA(X(2),S,2)
      X(3)=DATA/100-X(1)*100-X(2)*10
      IF(X(1).NE.0 .OR. X(2).NE.0 .OR. X(3).NE.0) CALL TROCA(X(3),S,3)
      X(4)=DATA/10-X(1)*1000-X(2)*100-X(3)*10
      IF(X(1).NE.0.OR.X(2).NE.0.OR.X(3).NE.0.OR.X(4).NE.0)
     1 CALL TROCA(X(4),S,4)
      X(5)=DATA-X(1)*10000-X(2)*1000-X(3)*100-X(4)*10
      CALL TROCA(X(5),S,5)
      RETURN
1     DO 20 I=1,5
      S(I)='*'
20    CONTINUE
      RETURN
      END



      SUBROUTINE DSPNUM(ADATA,DATA,X,Y,T,DISPLAY)
*************************************************************
* MOSTRA VALORES NUMERICOS DOS GRAFICOS APRESENTADOS
* NO SISTEMA
*************************************************************
C
      IMPLICIT INTEGER*2 (C-Z)
      REAL ADATA
      CHARACTER*1 STRG(5)
      IF(T.EQ.1) CALL CVR(ADATA,STRG)
      IF(T.EQ.0) CALL CVI(DATA,STRG)
      STATUS=VSAPOS(DISPLAY,X,Y,X,Y)
      IF(STATUS.EQ.-1) RETURN
      DO 10 I=1,5
      STATUS=VATXTS(DISPLAY,1,STRG(I),X,Y)
      IF(STATUS.EQ.-1) RETURN
10    CONTINUE
      RETURN
      END
```

```
      SUBROUTINE ESTAT(NP,DISPLAY,Y)
*****************************************************
* APRESENTA DADOS ESTATISTICOS FORNECIDOS PELO SISTEMA
*****************************************************
C
      IMPLICIT INTEGER*2 (C-Z)
      IMPLICIT REAL (A-B)
      CHARACTER*18 NOME
      DIMENSION NP(32),COR(3),XY(4)
      DATA COR /2,3,3/
      DATA NOME /'DADOS ESTATISTICOS'/
      CALL VMDP(NP,AMED,ADP,MAX,MIN)
      XY(1)=6383
      XY(3)=26383
      XY(2)=Y
      XY(4)=Y+3500
      STATUS=VCLARY(DISPLAY,XY,1,1,3,4,COR)
      X1=2721+XY(1)
      Y1=Y+2400
      STATUS=VSAPOS(DISPLAY,X1,Y1,XOUT,YOUT)
      STATUS=VATXTS(DISPLAY,18,NOME,XOUT,YOUT)
      Y1=Y1-1100
      STATUS=VSAPOS(DISPLAY,7000,Y1,XOUT,YOUT)
      STATUS=VATXTS(DISPLAY,6,'MEDIA:',XOUT,YOUT)
      CALL DSPNUM(AMED,0,XOUT,YOUT,1,DISPLAY)
      X1=XOUT+850
      STATUS=VSAPOS(DISPLAY,X1,YOUT,XOUT,YOUT)
      STATUS=VATXTS(DISPLAY,6,'D PAD:',XOUT,YOUT)
      CALL DSPNUM(ADP,0,XOUT,YOUT,1,DISPLAY)
      Y1=YOUT-850
      STATUS=VSAPOS(DISPLAY,7000,Y1,XOUT,YOUT)
      STATUS=VATXTS(DISPLAY,6,'MIN  :',XOUT,YOUT)
      CALL DSPNUM(0.,MIN,XOUT,YOUT,0,DISPLAY)
      X1=XOUT+850
      STATUS=VSAPOS(DISPLAY,X1,YOUT,XOUT,YOUT)
      STATUS=VATXTS(DISPLAY,6,'MAX  :',XOUT,YOUT)
      CALL DSPNUM(0.,MAX,XOUT,YOUT,0,DISPLAY)
      RETURN
      END

      SUBROUTINE CVR(ADATA,S)
*****************************************************
* CONVERTE DADOS DE ENTRADA NO MODO GRAFICO PARA VALORES
* REAIS
*****************************************************
C
      IMPLICIT INTEGER*2 (C-Z)
      REAL ADATA,E
      CHARACTER*1 S(5)
      DIMENSION X(4),IND(8,5)
      DATA IND /5,4,3,2,1,1,1,1,1,1,1,2,3,4,5,2,2,2,3,3,4,5,
     1  0,3,3,4,4,4,5,0,0,4,5,5,5,5,0,0,0/
      DO 5 I=1,5
      S(I)='0'
5     CONTINUE
      S(4)='.'
      K=55
      DO 10 I=1,8
      IN=I
```

```
      E=I-4.
      X(1)=ADATA*(10.**E)
      IF(X(1).NE.0) GO TO 1
10    CONTINUE
      RETURN
1     IF(X(1).GT.9) GO TO 20
      I=IND(IN,1)
      CALL TROCA(K,S,I)
      I=IND(IN,2)
      CALL TROCA(X(1),S,I)
      IF(IN.EQ.8) RETURN
      E=IN-3.
      X(2)=(ADATA*(10.**E))-X(1)*10
      I=IND(IN,3)
      CALL TROCA(X(2),S,I)
      IF(IN.EQ.7) RETURN
      E=IN-2.
      X(3)=(ADATA*(10.**E))-X(1)*100-X(2)*10
      I=IND(IN,4)
      CALL TROCA(X(3),S,I)
      IF(IN.EQ.6) RETURN
      E=IN-1.
      X(4)=(ADATA*(10.**E))-X(1)*1000-X(2)*100-X(3)*10
      I=IND(IN,5)
      CALL TROCA(X(4),S,I)
      RETURN
20    DO 25 I=1,5
      S(I)='*'
25    CONTINUE
      RETURN
      END


      SUBROUTINE TROCA(X,S,I)
*****************************************************************
* CONVERTE TODO O CAMPO DOS DADOS DE ENTRADA NO MODO GRAFICO
* PARA VALORES NUMERICOS
*****************************************************************
C
      IMPLICIT INTEGER*2 (C-Z)
      CHARACTER*1 S(5)
      IF(X.EQ.0) S(I)='0'
      IF(X.EQ.1) S(I)='1'
      IF(X.EQ.2) S(I)='2'
      IF(X.EQ.3) S(I)='3'
      IF(X.EQ.4) S(I)='4'
      IF(X.EQ.5) S(I)='5'
      IF(X.EQ.6) S(I)='6'
      IF(X.EQ.7) S(I)='7'
      IF(X.EQ.8) S(I)='8'
      IF(X.EQ.9) S(I)='9'
      IF(X.GT.9) S(I)='.'
      RETURN
      END
```

```
      INTEGER*2 FUNCTION SPDIC1(I4)
*********************************************************
*     FUNCTION DE AUXILIO AO USUARIO: CARREGA IMAGEM 64*64
*********************************************************
C
      IMPLICIT INTEGER*2 (C-Z)
      CHARACTER NOME*14,VOLTA
      LOGICAL EX
      DIMENSION WORKIN(19),WORKOUT(66),IA(64,64),EXE(2)
C
      DATA WORKIN /11*1,68,73,83,80,76,65,89,32/
C
      STATUS=VOPNWK(WORKIN,DISPLAY,WORKOUT)
      STATUS=VCRCOL(DISPLAY,1,4,I,J)
1     STATUS=VENCUR(DISPLAY)
C
      STATUS=VCURAD(DISPLAY,8,10)
      STATUS=VCTXTS(DISPLAY,60,'PICTOREA-UMA FERRAMENTA DE ENSINO PARA T
     .RATAMENTO DE IMAGENS')
      STATUS=VCURAD(DISPLAY,12,30)
      STATUS=VCTXTS(DISPLAY,19,'PROGRAMA DO USUARIO')
      STATUS=VCURAD(DISPLAY,14,24)
      STATUS=VCTXTS(DISPLAY,31,'*** CARREGAMENTO DE IMAGENS ***')
      STATUS=VRVON(DISPLAY)
      STATUS=VCURAD(DISPLAY,24,10)
      STATUS=VCTXTS(DISPLAY,42,'DIGITE O NOME DO ARQUIVO A SER CARREGADO
     .: ')
      READ(*,'(1A14)',ERR=1000) NOME
C
      STATUS=VRVOFF(DISPLAY)
C
C     REALIZAR CARREGAMENTO
C
      INQUIRE(FILE=NOME,EXIST=EX)
      IF(.NOT.EX.OR.NOME.EQ.' ') GOTO 1000
      OPEN(1,FILE=NOME,ACCESS='DIRECT',ERR=1000,FORM='FORMATTED',
     .RECL=128)
      STATUS=VCURAD(DISPLAY,2,2)
      STATUS=VCTXTS(DISPLAY,23,'AGUARDE... CARREGANDO: ')
      STATUS=VCTXTS(DISPLAY,14,NOME)
      DO 20 I=1,64
      READ(1,'(64I2)',REC=I)(IA(I,J),J=1,64)
20    CONTINUE
      CLOSE(1)
C
      STATUS=VCURAD(DISPLAY,2,2)
      STATUS=VEREOL(DISPLAY)
      STATUS=VCTXTS(DISPLAY,27,'*** Fim de CARREGAMENTO ***')
      STATUS=VCURAD(DISPLAY,3,2)
      STATUS=VCRCOL(DISPLAY,1,2,I,J)
      STATUS=VCTXTS(DISPLAY,38,'Tecle <ENTER> para continuar execucao.')
      STATUS=VCRCOL(DISPLAY,1,4,I,J)
      ASSIGN 30 TO RETORNO
C
25    VOLTA=' '
      STATUS=VRQSTR(DISPLAY,1,0,EXE,VOLTA)
      IF(VOLTA.NE.' ') GOTO 25
      GOTO RETORNO
C
```

```
1000    STATUS=VCURAD(DISPLAY,2,2)
        STATUS=VCTXTS(DISPLAY,48,'*** ERRO NA ESPECIFICACAO DO NOME DO ARQ
     .UIVO ***')
        STATUS=VCURAD(DISPLAY,3,2)
        STATUS=VCTXTS(DISPLAY,27,'Tecle <ENTER> para repetir.')
        ASSIGN 1 TO RETORNO
        GOTO 25
C
30      SPDICI=0
        STATUS=VCRCOL(DISPLAY,1,0,1,J)
        STATUS=VCLRWK(DISPLAY)
        STATUS=VCLSWK(DISPLAY)
        RETURN
        END
        INTEGER*2 FUNCTION SPDIAI(IA)
*************************************************************
*     FUNCTION DE AUXILIO AO USUARIO: ARMAZENA IMAGEM 64*64
*************************************************************
C
        IMPLICIT INTEGER*2 (C-Z)
        CHARACTER NOME*14,VOLTA
        DIMENSION WORKIN(19),WORKOUT(66),IA(64,64),EXE(2)
C
        DATA WORKIN /11*1,68,73,83,80,76,65,89,32/
C
        STATUS=VOPNWK(WORKIN,DISPLAY,WORKOUT)
        STATUS=VCRCOL(DISPLAY,1,4,1,J)
        STATUS=VENCUR(DISPLAY)
C
        STATUS=VCURAD(DISPLAY,8,10)
        STATUS=VCTXTS(DISPLAY,60,'PICTOREA-UMA FERRAMENTA DE ENSINO PARA T
     .RATAMENTO DE IMAGENS')
        STATUS=VCURAD(DISPLAY,12,30)
        STATUS=VCTXTS(DISPLAY,19,'PROGRAMA DO USUARIO')
        STATUS=VCURAD(DISPLAY,14,24)
        STATUS=VCTXTS(DISPLAY,32,'*** ARMAZENAMENTO DE IMAGENS ***')
        STATUS=VCURAD(DISPLAY,24,10)
        STATUS=VRVON(DISPLAY)
        STATUS=VCTXTS(DISPLAY,43,'DIGITE O NOME DO ARQUIVO A SER ARMAZENAD
     .O: ')
1       READ(*,'(1A14)') NOME
C
        STATUS=VRVOFF(DISPLAY)
C
C     REALIZAR CARREGAMENTO
C
        IF(NOME.EQ.' ') GOTO 1
        OPEN(1,FILE=NOME,ACCESS='DIRECT',FORM='FORMATTED',
     .RECL=128,STATUS='NEW')
        STATUS=VCURAD(DISPLAY,2,2)
        STATUS=VCTXTS(DISPLAY,24,'AGUARDE... ARMAZENANDO: ')
        STATUS=VCTXTS(DISPLAY,14,NOME)
        DO 20 I=1,64
        WRITE(1,'(64I2)',REC=I)(IA(I,J),J=1,64)
20      CONTINUE
        CLOSE(1)
C
        STATUS=VCURAD(DISPLAY,2,2)
        STATUS=VEREOL(DISPLAY)
```

```
      STATUS=VCTXTS(DISPLAY,28,'*** Fim de ARMAZENAMENTO ***')
      STATUS=VCURAD(DISPLAY,3,2)
      STATUS=VCRCOL(DISPLAY,1,2,1,J)
      STATUS=VCTXTS(DISPLAY,38,'Tecle (ENTER) para continuar execucao.')
      STATUS=VCRCOL(DISPLAY,1,4,1,J)
C
25    VOLTA=' '
      STATUS=VRQSTR(DISPLAY,1,0,EXE,VOLTA)
      IF(VOLTA.NE.' ') GOTO 25
C
      SPDIAI=0
      STATUS=VCRCOL(DISPLAY,1,0,1,J)
      STATUS=VCLRWK(DISPLAY)
      STATUS=VCLSWK(DISPLAY)
      RETURN
      END
```

```
      program fourier

      implicit integer*2 (c-z)
      complex f(64), fimag(64,64)
      dimension ia(64,64), workin(19), workout(66)
      data workin /11*1,68,73,83,80,76,65,89,32/
c ler imagem

      i= spdici(ia)

      status=vopnwk(workin,display,workout)
      status=vcurad(display,3,2)
      status=vctxts(display,22,'*** Executando FFT ***')

c translacao da imagem

      do 50 i=1,64
      do 50 j=1,64
50    ia(i,j)=ia(i,j)*(-1)**(i+j)

      do 2 i=1,64
      do 3 j=1,64
3     f(j)=ia(i,j)
      call fft(f,6)
      do 4 k=1,64
4     fimag(i,k)=64.0*f(k)
2     continue

      do 5 j=1,64
      do 6 i=1,64
6     f(i)=fimag(i,j)
      call fft(f,6)
      do 7 k=1,64
7     fimag(k,j)=f(k)
5     continue
```

```
      do 8 i=1,64
      do 8 j=1,64
8     ia(i,j)=8*nint(alog(1. + anint(cabs(fimag(i,j)))))

      status=vencur(display)
      status=vclswk(display)

c armazenar imagem

      i= spdiai(ia)

      stop
      end
*****************************

      SUBROUTINE FFT(F, LN)

      COMPLEX F(64), U, W, T, CMPLX
      PI=3.141593
      N=2**LN
      NV2=N/2
      NM1=N-1
      J=1
      DO 3 I=1,NM1
      IF(I. GE. J) GOTO 1
      T=F(J)
      F(J)=F(I)
      F(I)=T
1     K=NV2
2     IF(K. GE. J) GOTO 3
      J=J-K
      K=K/2
      GOTO 2
3     J=J+K
      DO 5 L=1,LN
      LE=2**L
      LE1=LE/2
      U=(1.0,0.0)
      W=CMPLX(COS(PI/LE1), -SIN(PI/LE1))
      DO 5 J=1, LE1
      DO 4 I=J,N,LE
      IP=I+LE1
      T=F(IP)*U
      F(IP)=F(I)-T
4     F(I)=F(I)+T
5     U=U*W
      DO 6 I=1,N
6     F(I)=F(I)/FLOAT(N)
      RETURN
      END
```

```
      program  walsh

      implicit integer*2 (c-z)
      real*4 walsh(64), walshimag(64,64)
      dimension ia(64,64), workin(19), workout(66)
      data workin /11*1,68,73,83,80,76,65,89,32/



      i=spdici(ia)

      status=vopnwk(workin,display,workout)
      status=vcurad(display,3,2)
      status=vctxts(display,22,'*** Executando FWT ***')

c  translacao da imagem
      do 50 i=1,64
      do 50 j=1,64
50    ia(i,j)=ia(i,j)*(-1)**(i+j)

      do 12 i=1,64
      do 13 j=1,64
 13   walsh(j)=ia(i,j)
      call fwt(walsh,6)
      do 14 k=1,64
 14   walshimag(i,k)=64.0*walsh(k)
 12   continue

      do 15 j=1,64
      do 16 i=1,64
 16   walsh(i)=walshimag(i,j)
      call fwt(walsh,6)
      do 17 k=1,64
 17   walshimag(k,j)=walsh(k)
 15   continue

      do 18 i=1,64
      do 18 j=1,64
      ia(i,j)=abs(walshimag(i,j))
      if(ia(i,j). GT. 31) ia(i,j) = 31
18    continue

      status=vencur(display)
      status=vclswk(display)

c armazenar imagem

      i=spdiai(ia)

      stop
      end


      SUBROUTINE FWT(F,LN)
*******************************************
* Executa a transformada rapida de Walsh
*******************************************

      REAL*4 F(1024), T
```

```
      N=2**LN
      NV2=N/2
      NM1=N-1
      J=1
      DO 3 I=1,NM1
      IF(I. GE. J) GOTO 1
      T=F(J)
      F(J)=F(I)
      F(I)=T
1     K=NV2
2     IF(K. GE. J) GOTO 3
      J=J-K
      K=K/2
      GOTO 2
3     J=J+K
      DO 5 L=1,LN
      LE=2**L
      LE1=LE/2
      DO 5 J=1,LE1
      DO 4 I=J,N,LE
      IP=I+LE1
      T=F(IP)
      F(IP)=F(I)-T
4     F(I)=F(I)+T
5     CONTINUE
      DO 6 I=1,N
6     F(I)=F(I)/FLOAT(N)
      RETURN
```

PICTŌREA

PROGRAMAS FONTES MODIFICADOS NA VERSĀO EM 16 CORES

```
            PROGRAM PICTOREA
C
      IMPLICIT INTEGER*2 (C-Z)
      DIMENSION WORKIN(19),WORKOUT(66),RGBI(3),RGBREAL(3)
      COMMON /PRAESTATLOC/AREA,/PRAVOLTA/GRAFICO
C
      DATA WORKIN /11*1,68,73,83,80,76,65,89,32/
C
C     ABRIR ESTACAO DE TRABALHO:
C
      STATUS=VOPNWK(WORKIN,DISPLAY,WORKOUT)
C
      STATUS=VENCUR(DISPLAY)
      STATUS=VCRCOL(DISPLAY,1,4,FORE,BACK)
      STATUS=VENCUR(DISPLAY)
C
C     DEFINE CORES DE 8 A 15 DIFERENTES DA COR BRANCA
C
      RGBI(1)=600
      RGBI(2)=600
      RGBI(3)=0
      STATUS=VSCOLR(DISPLAY,8,RGBI,RGBREAL)
      RGBI(1)=600
      RGBI(2)=600
      RGBI(3)=600
      STATUS=VSCOLR(DISPLAY,9,RGBI,RGBREAL)
      RGBI(1)=400
      RGBI(2)=400
      RGBI(3)=400
      STATUS=VSCOLR(DISPLAY,10,RGBI,RGBREAL)
      RGBI(1)=400
      RGBI(2)=400
      RGBI(3)=1000
      STATUS=VSCOLR(DISPLAY,11,RGBI,RGBREAL)
      RGBI(1)=400
      RGBI(2)=1000
      RGBI(3)=400
      STATUS=VSCOLR(DISPLAY,12,RGBI,RGBREAL)
      RGBI(1)=400
      RGBI(2)=1000
      RGBI(3)=1000
      STATUS=VSCOLR(DISPLAY,13,RGBI,RGBREAL)
      RGBI(1)=1000
      RGBI(2)=400
      RGBI(3)=400
      STATUS=VSCOLR(DISPLAY,14,RGBI,RGBREAL)
      RGBI(1)=1000
      RGBI(2)=400
      RGBI(3)=1000
      STATUS=VSCOLR(DISPLAY,15,RGBI,RGBREAL)
C
C     INICIALIZA VARIAVEIS DE SINALIZACAO
C
      INDICA=0
      INDICB=0
      AREA=0.
      GRAFICO=0
C
      CALL MENUP(DISPLAY)
      END
```

```
      SUBROUTINE IMAGEM(IA,NOME,DISPLAY,X,Y,DM,DN)
*****************************************************************
*     REALIZA O DISPLAY DE IMAGENS: VERSAO 16 CORES
*****************************************************************
      IMPLICIT INTEGER*2(C-Z)
      CHARACTER*14 NOME , MSG*26,PELA(720),PELB(720)
      DIMENSION IA(64,64),XY(4)
      COMMON/SINAL/INDICA,INDICB

      Y1=Y+14000
      X1=X+3000
      IF(DN.EQ.0)GO TO 2
      IF(DM.EQ.0.OR.DM.EQ.1)GO TO 3
      X1=103*X
      Y1=103*Y
3     STATUS=VSAPOS(DISPLAY,X1,Y1,XOUT,YOUT)
      STATUS=VATXTS(DISPLAY,14,NOME,XOUT,YOUT)
2     X2=X
      XY(1)=X
      XY(2)=Y
      IF(INDICA.EQ.1.AND.DN.EQ.1) THEN
      STATUS=VPTPEL(DISPLAY,XY,PELA)
      GOTO 11
     ENDIF
      IF(INDICB.EQ.1.AND.DN.EQ.2) THEN
      STATUS=VPTPEL(DISPLAY,XY,PELB)
      GOTO 11
     ENDIF
      DO 10 I=1,64
      Y2=Y
      XY(1)=X2
      XY(3)=X2+210
      DO 5 J=64,1,-1
      XY(2)=Y2
      XY(4)=Y2+210
      IMAGE=IA(J,I)
      COR=0
      IF(IMAGE.GE.2.AND.IMAGE.LT.4) COR=10
      IF(IMAGE.GE.4.AND.IMAGE.LT.6) COR=4
      IF(IMAGE.GE.6.AND.IMAGE.LT.8) COR=3
      IF(IMAGE.GE.8.AND.IMAGE.LT.10) COR=6
      IF(IMAGE.GE.10.AND.IMAGE.LT.12) COR=2
      IF(IMAGE.GE.12.AND.IMAGE.LT.14) COR=7
      IF(IMAGE.GE.14.AND.IMAGE.LT.16) COR=8
      IF(IMAGE.GE.16.AND.IMAGE.LT.18) COR=9
      IF(IMAGE.GE.18.AND.IMAGE.LT.20) COR=11
      IF(IMAGE.GE.20.AND.IMAGE.LT.22) COR=12
      IF(IMAGE.GE.22.AND.IMAGE.LT.24) COR=13
      IF(IMAGE.GE.24.AND.IMAGE.LT.26) COR=14
      IF(IMAGE.GE.26.AND.IMAGE.LT.28) COR=15
      IF(IMAGE.GE.28.AND.IMAGE.LT.30) COR=5
      IF(IMAGE.GE.30) COR=1
      STATUS=VCLARY(DISPLAY,XY,1,1,1,4,COR)
      Y2=Y2+210
5     CONTINUE
      X2=X2+210
10    CONTINUE
      XY(1)=X
      XY(2)=Y
```

```
      XY(3)=X+13440
      XY(4)=Y+13440
      IF(INDICA.EQ.0.AND.DN.EQ.1) THEN
      STATUS=VGTPEL(DISPLAY,XY,PELA)
      INDICA=1
     ENDIF
      IF(INDICB.EQ.0.AND.DN.EQ.2) THEN
      STATUS=VGTPEL(DISPLAY,XY,PELB)
      INDICB=1
     ENDIF
11    IF(DN.EQ.0) RETURN
      CALL CRFIM(DISPLAY)
C
      END
C
C
      SUBROUTINE CRFIM(DISPLAY)
*************************************************
*     APRESENTA OPCOES DE COPIA,RETORNA E FIM
*************************************************
C
      IMPLICIT INTEGER*2 (C-Z)
C
      STATUS=VGTXTS(DISPLAY,6000,1000,26,'F1-Copia F2-Retorna F3-Fim')
15    STATUS=VRQCHC(DISPLAY,COUT,COUT)
      IF(COUT.EQ.1.AND.STATUS.GT.0) THEN
      STATUS=VGTXTS(DISPLAY,6000,1000,26,'                          ')
      STATUS=VHDCPY(DISPLAY)
      STATUS=VGTXTS(DISPLAY,6000,1000,26,'F1 Copia F2 Retorna F3 Fim')
     ENDIF
      IF(COUT.EQ.2) THEN
      STATUS=VENCUR(DISPLAY)
      RETURN
     ENDIF
      IF(COUT.NE.3) GOTO 15
      STATUS=VENCUR(DISPLAY)
      STATUS=VCRCOL(DISPLAY,1,0,X,Y)
      STATUS=VCLRWK(DISPLAY)
      STATUS=VCLSWK(DISPLAY)
      STOP
      END

      SUBROUTINE MOVE (XIM,YIM,TF,XYB,DESLOC,FLAG,DIR,DISPLAY)
C*****************************************************
C     REALIZA O MOVIMENTO DA JANELA NA IMAGEM
C*****************************************************
      IMPLICIT INTEGER*2 (C-Z)
      DIMENSION XYB(4),LIMP(4)
      CHARACTER JANELA(720)
      COMMON/PASSO1/PASSO,/WINDOW/JANELA
C
      DIR=DIR-58
      FLAG1=FLAG
      FLAG2=FLAG
      DIR1=1
C
C     ESPECIFICA PASSAGEM NA SUBROTINA MOVE
C
```

```
      IF(PASSO.EQ.1) STATUS=VPTPEL(DISPLAY,XYB,JANELA)
      PASSO=1
C
30    STATUS=VSWRMD(DISPLAY,12)
      STATUS=VGTPEL(DISPLAY,XYB,JANELA)
C
C     DESENHAR JANELA
C
      STATUS=VBAR(DISPLAY,XYB)
C
C     LER MOVIMENTO DO CURSOR
C
      IF(FLAG1.EQ.1) GOTO 42
40    STATUS=VRQCHC(DISPLAY,DIR,DIR)
      IF(DIR.GT.5) GOTO 40
      DIR1=STATUS
C
C     APAGAR AREA QUE APRESENTA EFEITOS DE ZOOM E REDUCAO
C
      IF(FLAG2.NE.2.OR.DIR.EQ.5) GOTO 42
      LIMP(1)=17000
      LIMP(2)=6000
      LIMP(3)=31000
      LIMP(4)=21000
      STATUS=VCLARY(DISPLAY,LIMP,1,1,1,4,0)
C
C     RESTAURA AREA ARMAZENADA:
C
42    FLAG1=0
      STATUS=VSWRMD(DISPLAY,4)
      IF(DIR.EQ.5.OR.(DIR.EQ.1).AND.DIR1.EQ.0) RETURN
      STATUS=VPTPEL(DISPLAY,XYB,JANELA)
C
C     DESLOCAR JANELA DE ACORDO COM A INDICACAO DO CURSOR
C
      IF(DIR.EQ.2) THEN
      XYB(1)=XYB(1)+DESLOC*210
      XYB(3)=XYB(3)+DESLOC*210
      IF(XYB(3).LT.(XIM+13440+DESLOC*210)) GOTO 30
      XYB(1)=XIM
      XYB(3)=XIM+TF*210
C
      ELSE IF(DIR.EQ.1) THEN
      XYB(1)=XYB(1)-DESLOC*210
      XYB(3)=XYB(3)-DESLOC*210
      IF(XYB(1).GT.(XIM-DESLOC*210)) GOTO 30
      XYB(1)=XIM+13440-TF*210
      XYB(3)=XIM+13440
C
      ELSE IF(DIR.EQ.4) THEN
      XYB(2)=XYB(2)-DESLOC*210
      XYB(4)=XYB(4)-DESLOC*210
      IF(XYB(2).GT.(YIM-DESLOC*210)) GOTO 30
      XYB(2)=YIM+13440-TF*210
      XYB(4)=YIM+13440
C
      ELSE IF(DIR.EQ.3) THEN
      XYB(2)=XYB(2)+DESLOC*210
```

```
          XYB(4)=XYB(4)+DESLOC*210
          IF(XYB(4).LT.(YIM+13440+DESLOC*210)) GOTO 30
          XYB(2)=YIM
          XYB(4)=YIM+TF*210
        ENDIF
C
          GOTO 30
C
          END


C       SUBROUTINE ESTATLOC(IA,NOME,DISPLAY,FLAG,DN)
C*****************************************************
C     REALIZA A FUNCAO 'ESTATISTICAS LOCAIS'
C*****************************************************
          IMPLICIT INTEGER*2 (C-Z)
          REAL AREA,AREA1
          DIMENSION XYC(4),XYB(4),IA(64,64),NP(32),XYEND(4),LIMP(4),
     .            MIAPAG(4)
          CHARACTER NOME*14,MSG*29,MSG1*32,TEXT*2,OPCAO,JANELA(614)
          COMMON /WIDE/ LINFER,LINSUP,COLINF,COLSUP,/INFO/ECXY(2),
     -          /PONTO/LINHA,COLUNA,XCUR,YCUR,NUML,NUMC,/DADO3/EINT,
     -          /PASSO1/PASSO,/SINAL/INDICA,INDICB,/PRAESTATLOC/AREA,
     -          /WINDOW/JANELA
          DATA XIM,YIM /1600,8000/
          DATA MSG /'F1-Copia F2-Repete F3-Retorna'/
          DATA MSG1/'F4-Verifica e/ou modifica pixels'/
C
C     DEFINE CARACTERISTICAS DA JANELA (BAR)
C
          STATUS=VSFCOL(DISPLAY,0)
          STATUS=VSFINT(DISPLAY,0)
C
          GOTO (1,2) FLAG
1         CALL MOLDURA(DISPLAY)
          CALL IMAGEM(IA,NOME,DISPLAY,XIM,YIM,0,DN)
          MIAPAG(1)=XIM+64*210+50
          MIAPAG(2)=YIM-200
          MIAPAG(3)=MIAPAG(1)+100
          MIAPAG(4)=YIM+64*210+100
          STATUS=VCLARY(DISPLAY,MIAPAG,1,1,1,4,0)
C
          STATUS=VGTXTS(DISPLAY,2000,1000,30,'Janela: F1-4*4 F2-8*8 F3-16*16
     .')
          STATUS=VRQCHC(DISPLAY,CHIN,CHFIN)
          TF=4
          AREA=4.*4.
          IF(CHFIN.EQ.2) THEN
          TF=8
          AREA=8.*8.
        ENDIF
          IF(CHFIN.EQ.3) THEN
          TF=16
```

```
        AREA=16.*16.
      ENDIF
        AREA1=AREA
C
        STATUS=VGTXTS(DISPLAY,2000,3000,13,'Deslocamento?')
        STATUS=VGTXTS(DISPLAY,2000,2000,16,'F1-Ponto a ponto')
        STATUS=VGTXTS(DISPLAY,2000,1000,30,'F2-Comprimento da janela
     .')
        STATUS=VRQCHC(DISPLAY,CHIN,CHFIN)
        DESLOC=1
        IF(CHFIN.EQ.2) DESLOC=TF
C
        ASSIGN 31 TO RETORNO
C
3       STATUS=VGTXTS(DISPLAY,2000,3000,34,'Posicione a janela e tecle <EN
     .TER>')
        STATUS=VGTXTS(DISPLAY,2000,2000,22,'F1-Esquerda F2-Direita')
        STATUS=VGTXTS(DISPLAY,2000,1000,33,'F3-P/cima   F4-P/baixo F5-Te
     .rmina')
C
C     DEFINE AREA DA JANELA 5*5
C
        XYB(1)=XIM
        XYB(2)=YIM+13440-TF*210
        XYB(3)=XIM+TF*210
        XYB(4)=YIM+13440
C
        GOTO RETORNO
C
C     DESLOCAR JANELA NA IMAGEM:
C
31      ASSIGN 51 TO RETORNO
        PASSO=0
11      CALL MOVE (xim,yim,TF,XYB,DESLOC,0,DIR,DISPLAY)
C
        IF(DIR.EQ.5) GOTO 70
C
C     APRESENTAR HISTOGRAMA LOCAL:
C
50      LINFER=65-(XYB(4)-YIM)/210 +.5
        LINSUP=LINFER+(TF-1)
        COLSUP=(XYB(3)-XIM)/210 +.5
        COLINF=COLSUP-(TF-1)
        GOTO RETORNO
C
51      CALL CONTA3 (-1,0,IA,NP)
        CALL GRAFO2 (NP,DISPLAY,20000,8000,32)
        AREA=AREA1
        CALL ESTAT(NP,DISPLAY,4000)
C
        GOTO 11
C
70      LIMP(1)=2000
        LIMP(2)=200
        LIMP(3)=30000
        LIMP(4)=4000
        STATUS=VCLARY(DISPLAY,LIMP,1,1,1,4,0)
        STATUS=VGTXTS(DISPLAY,2000,2000,29,MSG)
        STATUS=VGTXTS(DISPLAY,2000,1000,32,MSG1)
```

```
71      STATUS=VRQCHC(DISPLAY,CHIN,CHIN)
        IF(CHIN.EQ.1.AND.STATUS.GT.0) THEN
        STATUS=VGTXTS(DISPLAY,2000,2000,29,'
     .)
        STATUS=VGTXTS(DISPLAY,2000,1000,32,'
     .   ')
        STATUS=VHDCPY(DISPLAY)
        STATUS=VGTXTS(DISPLAY,2000,2000,29,MSG)
        STATUS=VGTXTS(DISPLAY,2000,1000,32,MSG1)
        GOTO 71
       ENDIF
        IF(CHIN.NE.3) GOTO 72
        STATUS=VENCUR(DISPLAY)
        RETURN
72      IF(CHIN.EQ.4) GOTO 2
        IF(CHIN.NE.2) GOTO 71
        STATUS=VCLRWK(DISPLAY)
        GOTO 1
C
C     TRECHO DO PROGRAMA QUE VERIFICA E/OU MODIFICA PIXELS (2):
C
2       CALL MOLDURA(DISPLAY)
        CALL IMAGEM (IA,NOME,DISPLAY,XIM,YIM,0,DN)
        MIAPAG(1)=XIM+64*210+50
        MIAPAG(2)=YIM-200
        MIAPAG(3)=MIAPAG(1)+100
        MIAPAG(4)=YIM+64*210+100
        STATUS=VCLARY(DISPLAY,MIAPAG,1,1,1,4,0)
C
C     COLOCAR TEXTO PARA INTERFACE
C
        TF=5
        DESLOC=5
C
        STATUS=VGTXTS(DISPLAY,2000,4000,10,'Janela:5*5')
C
        ASSIGN 21 TO RETORNO
        GOTO 3
C
21      PASSO=0
        CALL MOVE (XIM,YIM,TF,IYB,DESLOC,0,DIR,DISPLAY)
        IF(DIR.EQ.5) GOTO 95
C
C     APRESENTAR PIXELS NA TELA:
C
        ASSIGN 52 TO RETORNO
        GOTO 50
C
52      STATUS=VGTXTS(DISPLAY,18000,20000,17,'Modifique pixel e')
        STATUS=VGTXTS(DISPLAY,18000,19000,13,'tecle (ENTER)')
        CALL MOSTRA (IA,DISPLAY)
C
        NUML=LINFER+2
        NUMC=COLINF+2
        WRITE(TEXT,'(I2)') NUML
        STATUS=VGTXTS(DISPLAY,19080,7000,7,'Linha =')
        STATUS=VGTXTS(DISPLAY,25680,7000,2,TEXT)
        WRITE(TEXT,'(I2)') NUMC
        STATUS=VGTXTS(DISPLAY,19080,6000,7,'Coluna=')
```

```
      STATUS=VGTXTS(DISPLAY,25680,6000,2,TEXT)
C
C     POSICAO INICIAL DO CURSOR:
C
      XCUR=24330
      YCUR=13455
      LINHA=3
      COLUNA=3
C
C     APRESENTAR CURSOR NO PONTO XCUR E YCUR
C
60    STATUS=VSFCOL(DISPLAY,1)
      XYC(1)=XCUR-600
      XYC(2)=YCUR-750
      XYC(3)=XCUR+1250
      XYC(4)=YCUR+600
      STATUS=VBAR(DISPLAY,XYC)
      STATUS=VSFCOL(DISPLAY,0)
C
C     ESPERA A RECEPCAO DE UMA TECLA VALIDA:
C
61    OPCAO=' '
      STATUS=VRQSTR(DISPLAY,1,0,XYEND,OPCAO)
      EINT=ICHAR(OPCAO)
      IF(EINT.EQ.0) GOTO 61
      IF(EINT.EQ.63) GOTO 95
      STATUS=VBAR(DISPLAY,XYC)
      IF(OPCAO.EQ.' ') GOTO 52
C
C     ESPERA TECLA DE MOVIMENTO DE CURSOR
C
      IF(EINT.EQ.72.OR.EINT.EQ.75.OR.EINT.EQ.77.OR.EINT.EQ.80) CALL
     . MOVE1 (EINT,DISPLAY,*60)
C
C     ESPERA TECLA DE FUNCAO ESPECIAL(MOVIMENTO DA JANELA)
C
      IF(EINT.GT.58.AND.EINT.LT.63) THEN
      LIMP(1)=16500
      LIMP(2)=5500
      LIMP(3)=32600
      LIMP(4)=21000
      STATUS=VCLARY(DISPLAY,LIMP,1,1,1,4,0)
      CALL MOVE(XIM,YIM,TF,XYB,DESLOC,1,EINT,DISPLAY)
      IF(EINT.EQ.5) GOTO 95
      GOTO 50
     ENDIF
C
C     EFETUAR MODIFICACAO DOS VALORES DOS PIXELS
C
      IF(EINT.GT.47.AND.EINT.LT.58) THEN
C
C     APRESENTA PRIMEIRO DIGITO DO DADO DE ENTRADA:
C
      STATUS=VGTXTS(DISPLAY,XCUR-410,YCUR-400,3,'   ')
      STATUS=VSAPOS(DISPLAY,XCUR-410,YCUR-580,XCUR1,YCUR1)
      STATUS=VATXTS(DISPLAY,1,OPCAO,XCUR1,YCUR1)
C
C     APRESENTA SEGUNDO DIGITO DO DADO DE ENTRADA:
C
```

```
      CALL INFORM(DISPLAY,3,XCUR1,YCUR1+100,*85,*80,*61)
      GOTO 90
80    IF(ECXY(1).GT.31) GOTO 97
      STATUS=VGTXTS(DISPLAY,XCUR-410,YCUR-400,3,'   ')
      WRITE(TEXT,'(I2)') IA(NUML,NUMC)
      STATUS=VGTXTS(DISPLAY,XCUR-410,YCUR-450,2,TEXT)
      GOTO 60
85    ECXY(1)=EINT-48
C
C     ARMAZENAR NOVO DADO NA MEMORIA DE IMAGEM
C
90    IA(NUML,NUMC)=ECXY(1)
C
      STATUS=VPTPEL(DISPLAY,XYB,JANELA)
C
C     APRESENTAR NOVO VALOR DO PIXEL NA TELA
C
      XYEND(1)=(NUMC-1)*210+XIM
      XYEND(2)=(64-NUML)*210+YIM
      XYEND(3)=XYEND(1)+210
      XYEND(4)=XYEND(2)+210
      CALL IMAGEP(IA(NUML,NUMC),XYEND,DISPLAY)
C
C     ARMAZENAR AREA MODIFICADA EM JANELA
C
      STATUS=VGTPEL(DISPLAY,XYB,JANELA)
      STATUS=VSWRMD(DISPLAY,12)
      STATUS=VBAR(DISPLAY,XYB)
      STATUS=VSWRMD(DISPLAY,4)
C
      IF(DN.EQ.1) INDICA=0
      IF(DN.EQ.2) INDICB=0
     ENDIF
      GOTO 60
95    CALL CRRET(DISPLAY,*2,2000,2000)
      RETURN
C
97    STATUS=VENCUR(DISPLAY)
      RETURN
      END

      SUBROUTINE PPONTOS(IA,IC,DISPLAY,FLAG)
C*************************************************************
C     REALIZA A DETECAO DE BORDAS UTILIZANDO O GRADIENTE DE
C     ROBERTS
C*************************************************************
C
      IMPLICIT INTEGER*2 (C-Z)
      DIMENSION IA(64,64),IC(64,64),XY(4)
      DATA PXIM,PYIM /17500,3400/
C
      CALL MOLDURA(DISPLAY)
      STATUS=VGTXTS(DISPLAY,5500,20810,29,'Aplicacao direta do gradien
     .te')
      CALL IMAGEM(IA,'Original      ',DISPLAY,2000,3600,0,FLAG)
      STATUS=VGTXTS(DISPLAY,PXIM+2000,PYIM+14400,10,'Processada')
C
      XY(2)=PYIM+13440+210
      DO 5 X=1,63
      XY(1)=PXIM-210
```

```
      XY(2)=XY(2)-210
      XY(4)=XY(2)+210
      DO 5 Y=1,63
      XY(1)=XY(1)+210
      XY(3)=XY(1)+210
      IC(X,Y)=IMAGRAD(X,Y,IA)
      CALL IMAGEP(IC(X,Y),XY,DISPLAY)
5     CONTINUE
C
C     COMPLETAR MEMORIA DE IMAGEM
C
      CALL COMPGRAD(PXIM,PYIM,XY,IA,IC,DISPLAY,FLAG)
C
      RETURN
      END
C
C
      SUBROUTINE IMAGEP(PONTOP,XY,DISPLAY)
*************************************************************
*     APRESENTA IMAGEM NA TELA DE ACORDO COM OS VALORES
*     DE PONTOP
*************************************************************
C
      IMPLICIT INTEGER*2 (C-Z)
      DIMENSION XY(4)
C

      COR=0
      IF(PONTOP.GE.2.AND.PONTOP.LT.4) COR=10
      IF(PONTOP.GE.4.AND.PONTOP.LT.6) COR=4
      IF(PONTOP.GE.6.AND.PONTOP.LT.8) COR=3
      IF(PONTOP.GE.8.AND.PONTOP.LT.10) COR=6
      IF(PONTOP.GE.10.AND.PONTOP.LT.12) COR=2
      IF(PONTOP.GE.12.AND.PONTOP.LT.14) COR=7
      IF(PONTOP.GE.14.AND.PONTOP.LT.16) COR=8
      IF(PONTOP.GE.16.AND.PONTOP.LT.18) COR=9
      IF(PONTOP.GE.18.AND.PONTOP.LT.20) COR=11
      IF(PONTOP.GE.20.AND.PONTOP.LT.22) COR=12
      IF(PONTOP.GE.22.AND.PONTOP.LT.24) COR=13
      IF(PONTOP.GE.24.AND.PONTOP.LT.26) COR=14
      IF(PONTOP.GE.26.AND.PONTOP.LT.28) COR=15
      IF(PONTOP.GE.28.AND.PONTOP.LT.30) COR=5
      IF(PONTOP.GE.30) COR=1
C
      STATUS=VCLARY(DISPLAY,XY,1,1,1,4,COR)
C
      RETURN
      END
C
C
      INTEGER*2 FUNCTION IMAGRAD(X,Y,IA)
*************************************
*     REALIZA O CALCULO DO GRADIENTE
*************************************
C
      IMPLICIT INTEGER*2 (C-Z)
      DIMENSION IA(64,64)
      IMAGRAD=IABS(IA(X,Y)-IA(X+1,Y+1))+IABS(IA(X+1,Y)-IA(X,Y+1))
      IF(IMAGRAD.GT.31) IMAGRAD=31
C
```

```
      RETURN
      END
C
C
      SUBROUTINE COMPGRAD (PXIM,PYIM,XY,IA,IC,DISPLAY,FLAG)
*****************************************************************
*     COMPLETA MEMORIA DE IMAGEM PARA GRADIENTE
*****************************************************************
C
      IMPLICIT INTEGER*2 (C-Z)
      DIMENSION IA(64,64),IC(64,64),XY(4)
C

      XY(1)=XY(1)+210
      XY(3)=XY(1)+210
      XY(2)=PYIM+13440+210
      DO 10 X=1,64
      XY(2)=XY(2)-210
      XY(4)=XY(2)+210
      IC(X,64)=IC(X,63)
      CALL IMAGEP(IC(X,64),XY,DISPLAY)
10    CONTINUE
C
      XY(2)=XY(2)-210+210
      XY(1)=PXIM-210
      DO 15 Y=1,64
      XY(1)=XY(1)+210
      XY(3)=XY(1)+210
      IC(64,Y)=IC(63,Y)
      CALL IMAGEP(IC(64,Y),XY,DISPLAY)
15    CONTINUE
C
      CALL CRFIM(DISPLAY)
C
      END

      SUBROUTINE SETA (DISPLAY,XY,END1,END2,DIREC)
*****************************************************************
*     INDICA POSICAO NAS COORDENADAS X,Y DE UM GRAFICO
*     ATRAVES DE SETAS
*****************************************************************
C
      IMPLICIT INTEGER*2 (C-Z)
      DIMENSION SENT(6)
      CHARACTER DIREC
C
      SENT(1)=XY-200
      SENT(2)=END1-200
      SENT(3)=XY
      SENT(4)=END1
      SENT(5)=XY+200
      SENT(6)=SENT(2)
      IF(DIREC.EQ.'V') GOTO 1
      SENT(2)=END1+200
      SENT(5)=SENT(1)
      SENT(6)=END1-200
C
1     STATUS=VPLINE(DISPLAY,3,SENT)
C
```

```
      SENT(1)=XY
      SENT(2)=END1
      SENT(3)=XY
      SENT(4)=END1-1500
      IF(DIREC.EQ.'V') GOTO 2
      SENT(3)=XY-1500
      SENT(4)=END1
2     STATUS=VPLINE(DISPLAY,2,SENT)
C
      IF(END2.EQ.0) RETURN
C
      SENT(1)=XY-200
      SENT(2)=SENT(4)
      SENT(3)=XY+200
      SENT(4)=END1
      SENT(5)=END2-200
      SENT(6)=SENT(2)
      IF(DIREC.EQ.'V') GOTO 3
      SENT(1)=XY-2000
      SENT(2)=END1-200
      SENT(3)=XY+500
      SENT(4)=END1+200
      SENT(5)=SENT(1)
      SENT(6)=END2-200
C
3     STATUS=VCPPEL(DISPLAY,SENT)
      END
            INTEGER*2 FUNCTION SPDIA1(IA)
*********************************************************
*     FUNCTION DE AUXILIO AO USUARIO: ARMAZENA IMAGEM 64*64
*********************************************************
C
      IMPLICIT INTEGER*2 (C-Z)
      CHARACTER NOME*14,VOLTA
      DIMENSION WORKIN(19),WORKOUT(66),IA(64,64),EXE(2)
C
      DATA WORKIN /11*1,68,73,83,80,76,65,89,32/
C
      STATUS=VOPNWK(WORKIN,DISPLAY,WORKOUT)
      STATUS=VENCUR(DISPLAY)
      STATUS=VCRCOL(DISPLAY,1,4,I,J)
      STATUS=VCLRWK(DISPLAY)
C
      STATUS=VCURAD(DISPLAY,8,10)
      STATUS=VCTXTS(DISPLAY,60,'PICTOREA-UMA FERRAMENTA DE ENSINO PARA T
     .RATAMENTO DE IMAGENS')
      STATUS=VCURAD(DISPLAY,12,30)
      STATUS=VCTXTS(DISPLAY,19,'PROGRAMA DO USUARIO')
      STATUS=VCURAD(DISPLAY,14,24)
      STATUS=VCTXTS(DISPLAY,32,'*** ARMAZENAMENTO DE IMAGENS ***')
      STATUS=VCURAD(DISPLAY,24,10)
      STATUS=VRVON(DISPLAY)
      STATUS=VCTXTS(DISPLAY,43,'DIGITE O NOME DO ARQUIVO A SER ARMAZENAD
     .O: ')
1     READ(*,'(1A14)') NOME
C
      STATUS=VRVOFF(DISPLAY)
C
C     REALIZAR CARREGAMENTO
```

```
C
      IF(NOME.EQ.' ') GOTO 1
      OPEN(1,FILE=NOME,ACCESS='DIRECT',FORM='FORMATTED',
     .RECL=128,STATUS='NEW')
      STATUS=VCURAD(DISPLAY,2,2)
      STATUS=VCTXTS(DISPLAY,24,'AGUARDE... ARMAZENANDO: ')
      STATUS=VCTXTS(DISPLAY,14,NOME)
      DO 20 I=1,64
      WRITE(1,'(64I2)',REC=I)(IA(I,J),J=1,64)
20    CONTINUE
      CLOSE(1)
C
      STATUS=VCURAD(DISPLAY,2,2)
      STATUS=VEREOL(DISPLAY)
      STATUS=VCTXTS(DISPLAY,28,'*** Fim de ARMAZENAMENTO ***')
      STATUS=VCURAD(DISPLAY,3,2)
      STATUS=VCRCOL(DISPLAY,1,2,I,J)
      STATUS=VCTXTS(DISPLAY,38,'Tecle <ENTER> para continuar execucao.')
      STATUS=VCRCOL(DISPLAY,1,4,I,J)
C
25    VOLTA=' '
      STATUS=VRQSTR(DISPLAY,1,0,EXE,VOLTA)
      IF(VOLTA.NE.' ') GOTO 25
C
      SPDIA1=0
      STATUS=VCRCOL(DISPLAY,1,0,I,J)
      STATUS=VCLRWK(DISPLAY)
      STATUS=VCLSWK(DISPLAY)
      RETURN
      END
      INTEGER*2 FUNCTION SPDIC1(IA)
*************************************************************
*     FUNCTION DE AUXILIO AO USUARIO: CARREGA IMAGEM 64*64
*************************************************************
C
      IMPLICIT INTEGER*2 (C-Z)
      CHARACTER NOME*14,VOLTA
      LOGICAL EX
      DIMENSION WORKIN(19),WORKOUT(66),IA(64,64),EXE(2)
C
      DATA WORKIN /11*1,68,73,83,80,76,65,89,32/
C
      STATUS=VOPNWK(WORKIN,DISPLAY,WORKOUT)
      STATUS=VENCUR(DISPLAY)
      STATUS=VCRCOL(DISPLAY,1,4,I,J)
1     STATUS=VENCUR(DISPLAY)
C
      STATUS=VCURAD(DISPLAY,8,10)
      STATUS=VCTXTS(DISPLAY,60,'PICTOREA-UMA FERRAMENTA DE ENSINO PARA T
     .RATAMENTO DE IMAGENS')
      STATUS=VCURAD(DISPLAY,12,30)
      STATUS=VCTXTS(DISPLAY,19,'PROGRAMA DO USUARIO')
      STATUS=VCURAD(DISPLAY,14,24)
      STATUS=VCTXTS(DISPLAY,31,'*** CARREGAMENTO DE IMAGENS ***')
      STATUS=VRVON(DISPLAY)
      STATUS=VCURAD(DISPLAY,24,10)
      STATUS=VCTXTS(DISPLAY,42,'DIGITE O NOME DO ARQUIVO A SER CARREGADO
     .: ')
      READ(*,'(1A14)',ERR=1000) NOME
C
```

```
      STATUS=VRVOFF(DISPLAY)
C
C     REALIZAR CARREGAMENTO
C
      INQUIRE(FILE=NOME,EXIST=EX)
      IF(.NOT.EX.OR.NOME.EQ.' ') GOTO 1000
      OPEN(1,FILE=NOME,ACCESS='DIRECT',ERR=1000,FORM='FORMATTED',
     .RECL=128)
      STATUS=VCURAD(DISPLAY,2,2)
      STATUS=VCTXTS(DISPLAY,23,'AGUARDE... CARREGANDO: ')
      STATUS=VCTXTS(DISPLAY,14,NOME)
      DO 20 I=1,64
      READ(1,'(64I2)',REC=I)(IA(I,J),J=1,64)
20    CONTINUE
      CLOSE(1)
C
      STATUS=VCURAD(DISPLAY,2,2)
      STATUS=VEREOL(DISPLAY)
      STATUS=VCTXTS(DISPLAY,27,'*** Fim de CARREGAMENTO ***')
      STATUS=VCURAD(DISPLAY,3,2)
      STATUS=VCRCOL(DISPLAY,1,2,I,J)
      STATUS=VCTXTS(DISPLAY,38,'Tecle <ENTER> para continuar execucao.')
      STATUS=VCRCOL(DISPLAY,1,4,I,J)
      ASSIGN 30 TO RETORNO
C
25    VOLTA=' '
      STATUS=VRQSTR(DISPLAY,1,0,EXE,VOLTA)
      IF(VOLTA.NE.' ') GOTO 25
      GOTO RETORNO
C
1000  STATUS=VCURAD(DISPLAY,2,2)
      STATUS=VCTXTS(DISPLAY,48,'*** ERRO NA ESPECIFICACAO DO NOME DO ARQ
     .UIVO ***')
      STATUS=VCURAD(DISPLAY,3,2)
      STATUS=VCTXTS(DISPLAY,27,'Tecle <ENTER> para repetir.')
      ASSIGN 1 TO RETORNO
      GOTO 25
C
30    SPDICI=0
      STATUS=VCRCOL(DISPLAY,1,0,I,J)
      STATUS=VCLRWK(DISPLAY)
      STATUS=VCLSWK(DISPLAY)
      RETURN
      END
```

REFERÊNCIAS BIBLIOGRÁFICAS

[1] WANG, David C.C. et alii. Digital image enhancement: A survey, in: Computer Vision, Graphics, and Image Processing 24, 1983, pp.363-381.

[2] MASCARENHAS, Nélson D.A. & VELASCO, Flávio R.D. Processamento Digital de Imagens. São Paulo, Quarta Escola de Computação, 1984.

[3] ROSENFELD, A. & KAK, A.C. Digital Picture Processing. New York, Academic Press, 1976.

[4] ARAÚJO, A. de A. Filtros Espaciais: Estudo comparativo e aplicação em segmentação e classificação de imagens. Tese de doutorado, Universidade Federal da Paraíba, 1987.

[5] ARAÚJO, A. de A. et alii. PICTÓREA: Um sistema didático de processamento de imagens. Anais, VIII Congresso da SBC, Salvador, BA, 1987 pp.398-400

[6] LEITE, N.J. et alii. PICTÓREA: Uma ferramenta de ensino portátil para tratamento de imagens. Anais, VI Simpósio Brasileiro de Telecomunicações - SBT, Campina Grande, PB, 1988, pp.231-237.

[7] GONZALEZ, R.G. & WINTZ, Paul. Digital Image Processing. Massachussets Addison-Wesley Company, 1977.

[8] WATTS, Richard A. Introducing Interactive Computing. Manchester, NCC Publications, 1984

[9] Graphics Development Toolkit, IBM - Personal Programming Family, 1984

[10] PRATT, W.K. Digital Image Processing. New York, John Wiley and Sons, 1978.

[11] DUDA, R.O. & HART, P.E. Pattern Classification and Scene Analysis, New York, John Wiley and Sons, 1973.

[12] PREWITT, J.M. Object enhancement and extraction, in B.S. Lipkin and A. Rosenfeld, Eds., Picture Processing and Psychopictorics, New York, Academic Press, 1970.

[13] KIRSCH, R. Computer determination of the constituint structure of biological imagens, in Computer and Biomedical Research 4, 1971 pp.315-328.

[14] ROBINSON, G.S. Edge detection by compass gradient masks, in Computer Graphics and Image Processing 6, 1977, pp.582-588.

[15] HEYGSTER, G. Rank filters in Digital imagem processing, in Computer Graphics and Image Processing 19, 1982, pp.148-164.

[16] NAGAO, M. & MATSUYAMA, T. Edge preserving smoothing, in Computer Graphics and Image Processing 9, 1978, pp.349-407.

[17] ARAÚJO, A. de A. Sum of absolute grey level differences: an edge-preserving smoothing approach, in Electronics Letters 21, 1985, pp.1219-1220.

[18] ARAÚJO, A. de A. Sim of absolute difference values smoothing: evaluation and application, in European Signal Processing Conference, 1986, pp.773-776.

[19] DAVIS, L.S. & ROSENFELD, A. Noise cleaning by iterated averaging, in IEEE T. on Systems, Man, and Cybernetics 8, 1978, pp.705-710.

[20] LEE, J.S. Digital image smoothing and the sigma filter, in Computer Graphics and Image Processing 24, 1983, pp.255-269.

[21] NAKAGAWA, N. & ROSENFELD. A note on the use of local min and max operations in digital picture processing, in IEEE T. on Systems, Man, and Cybernetics 8, 1978, 632-635.