UFF – CENTRO TECNOLÓGICO
INSTITUTO DE COMPUTAÇÃO
DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO

UNIVERSIDADE FEDERAL FLUMINENSE

Um estudo de técnicas para o desenvolvimento de sistemas de processamento de imagens digitais

Bruno B. Ribeiro

Monografia apresentada ao departamento de Ciência da Computação da Universidade Federal Fluminense como parte dos requisitos para obtenção do Grau de Bacharel em Ciência da Computação.

Banca Examinadora :

Aura Conci
Anna Dolejsi Santos
Luciana Ferraz Thomé

Departamento de Ciência da Computação

Niterói, 22 de agosto de 2002

Agradecimentos

Agradeço a minha orientadora, professora Aura Conci, pelo apoio e atenção dispensados para elaboração e desenvolvimento desse projeto. Agradeço a meus amigos Rodrigo Henriques, Renata Ataídes, Clarissa Cancherini, Maria Fernanda Terra, Tatiana Faro, Ana Glaidis, Hsia Min Wi e Cristiane Canettieri por me ajudarem a testar o programa. Agradeço a meus professores que foram fundamentais em minha caminhada e principalmente a meus pais, Margareth e Alberto, pelo apoio e confiança desde o início.

Resumo

Muitas pessoas não conhecem ou nunca utilizaram nenhuma forma de processamento de imagens digitais, mas ele está presente em muitos fatos ao nosso redor, desde revistas ou imagens para internet até métodos de análise geológica ou médica. Este projeto tem como objetivo de implementar diversas técnicas emostrar as possíveis utilidades da análise e melhoramento de imagens. Para isso serão estudados, implementados e utilizados diversos métodos de filtragem. Entre esses filtros alguns são muito conhecidos como os métodos de Laplace, Prewitt, Sobel e Roberts. Além desses outras funções também foram estudadas e implementadas, como a inversão de tons de uma imagem (ou seu negativo), transformação para tons de cinza, desmembramento e associaçãode canais de cores de imagem, adição de ruído e alterações através da função gamma.
Como resultado concreto desse trabalho, apresentamos um sistema de processamento de imagens totalmente aberto e implementado na linguagem c (visual c++ 6.0) que utiliza imagens no formato bitmap.

Abstract

Many people don’t know or have never used any kind of digital image processing, but it is present in many aspects in our environment, from simple magazines and Internet images to geological and medical analysis methods. The project aim is to show you possible utilities of image enhancement and analysis. To do it we will use several filters such as Laplace, Prewitt, Sobel and Roberts. In addiction we may apply other functions like negative image, noise add, turn grayscale, channel split and combing and gamma correction.

Índice

CAPITULO 1 – Introdução 7
CAPITULO 2 – Processamento de Imagens 10
2.1 – Considerações Iniciais 10
2.2 – Separação/Combinação de canais de cores (Channel Splitting/Combining) 12
2.3 – Inversão da Imagem (Negative) 13
2.4 – Transformação para tons de cinza (Gray Scale) 15
2.5 – Ajuste de contraste (Contrast) 16
2.6 – Correção Gamma (Gamma Correction) 17
2.7 – Conversão para True-Color (Increase Color Depth) 19
2.8 – Adição de ruídos (Add Noise) 19
2.9 – Filtros de Média 3x3/5x5/7x7 (Mean) 21
2.10 – Filtros de Mediana 3x3/5x5/7x7 (Median) 22
2.11 – Filtros Passa-Alta (Sharpening Balanced 1 e 2 e Unbalanced) 23
2.12 – Filtro Sobel Direcional Horizontal/Vertical (Sobel) 25
2.13 – Filtro Direcional de Roberts (Roberts) 26
2.14 – Filtros Passa-Alta de Prewitt (Prewitt) 28
2.15 – Filtrogem definida pelo usuário (User Defined Filter) 29
CAPITULO 3 – O programa implementado 31
3.1 – Introdução 31
3.2 - Menu File 32
3.3 - Menu Edit 33
3.4 - Menu View 34
3.5 - Menu Image 38
3.6 - Menu Filters 39
3.7 - Menu Windows 40
3.8 - Menu Help 40
CAPITULO 4 – Exemplos 42
CAPITULO 5 – Conclusão 49
Apêndice I – Formato de arquivo BITMAP 50
Referências : 54

Índice de Imagens

Erro! Nenhuma entrada de índice de figuras foi encontrada.

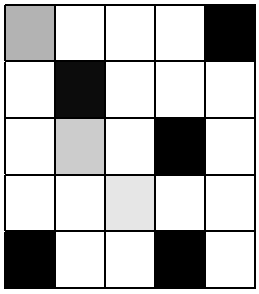
CAPITULO 1 – Introdução

A área de processamento digital de imagens tem crescido consideravelmente nas últimas décadas, com a utilização de imagens e gráficos em uma grande variedade de aplicações. A tecnologia de construção de computadores também tem se aprimorado, possibilitando a utilização de sistemas mais eficientes e baratos e com grandes áreas de armazenamento, condições essenciais ao processamento deste tipo de dado. Em muitas áreas o processamento de imagens digitais é essencial, tais como: reconhecimento de padrões (indústria), medicina, agricultura, pesquisas espaciais, metrologia, etc.

Matematicamente falando uma imagem pode ser representada por uma função bidimensional z = F(x,y) definida sobre certa região de um plano.

Uma imagem é representada através de um conjunto de valores , onde cada valor é um número que descreve os atributos de um *pixel* na imagem[JAI_89].

A figura 1.1 representa uma imagem em tons de cinza cuja variação desses tons vai de 0 a 255. Sua matriz correspondente apresenta os valores de z= F(x,y), onde cada quadrado da imagem representa um pixel.



127				255
	252			
	120		255	
		100		
253			250	

Figura 1.1 - Representação da imagem na forma de matriz.

Através do processamento de imagens podemos analisar e modificar imagens com o objetivo de extrair informações, reconhecer, comparar, classificar elementos que a compõem, transformar a imagem de forma a aumentar seu contraste, realçar bordas e corrigir alguma imperfeição, como ruído por exemplo.

Existem três tipos de operações sobre as imagens :

Operações Pontuais – onde o novo valor de um pixel depende apenas de uma transformação realizada em cima de seu valor inicial[IAN_98].

Operações Locais – ao contrário das operações pontuais, nas operações locais o novo valor de um pixel é determinado não apenas a partir de operações em cima de si mesmo, mas também utilizando os pixel vizinhos[IAN_98].

Operações Globais – o novo valor do pixel não depende apenas dele e nem de seus vizinhos, mas sim da imagem inteira[IAN_98].

O *Histograma de Intensidades* indica a quantidade de pontos existentes para cada cor presente na imagem. Desta forma o histograma possui informação global que é muito utilizada para diversos melhoramentos em imagens digitais, tal como contraste[GON_87]. A figura 1.2 apresenta uma imagem em tons de cinza e o seu histograma.

Figura 1.2 - Representação do histograma da imagem em tons de cinza.

Os algoritmos clássicos de processamento de imagens efetuam a *convolução* entre imagens, sendo utilizado para realizar filtragem espacial e procurar peculiaridades nas imagens.Convolução é um exemplo de operação local.

As operações de convolução substituem o valor de um ponto pelo valor obtido através de uma operação linear do ponto e seus vizinhos, sendo que são a eles atribuídos pesos. Estes pesos são colocados em uma matriz, usualmente de dimensão pequena, chamada de *máscara* ou *kernel*[GON_87]. Por exemplo a matriz abaixo representa um kernel 3x3.

$$K = \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Através de modificações no histograma ou da aplicação de filtros de convolução adequados muitos efeitos podem ser efetuados na imagem, alguns deles serão vistos no capítulo 2, onde veremos alguns exemplos de processos realizados utilizando-se a implementação deste projeto.

CAPITULO 2 – Processamento de Imagens

2.1 – Considerações Iniciais

Como já comentamos no capítulo 1, as imagens digitais podem ser representadas como uma matriz de pontos, chamados pixel. A cada pixel está associado um tom de cinza ou uma cor. Quando a imagem é colorida, são usados três canais para representá-la, um canal azul(Blue – B), um verde(Green – G) e um vermelho(Red –R), e com a variação da intensidade de cada canal conseguimos representar todas as cores.

Na figura 2.1 podemos ver a representação desses canais e suas combinações. Repare também que quando a intensidade dos três canais é igual, a imagem é representada somente em tonalidades de cinza, desde o branco ao preto.(diagonal pricipal do cubo de cores da figura 2.1)

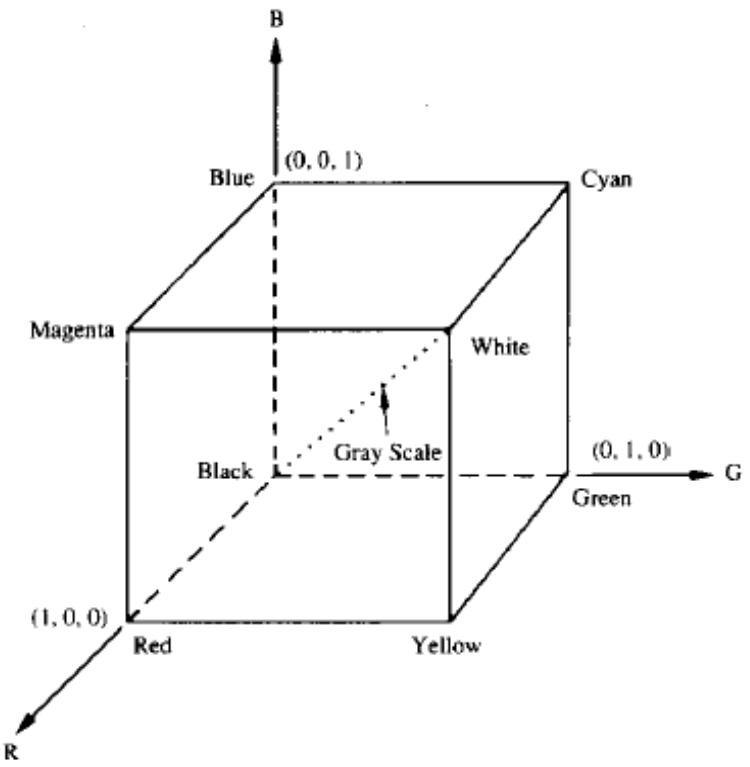
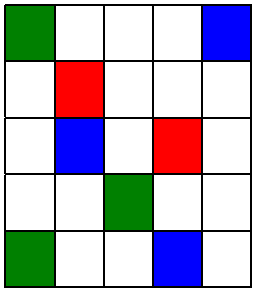
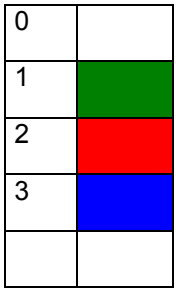


Figura 2.1 – Representação dos canais de cores em forma de cubo

Uma forma usual de representar imagens coloridas é associar a cada combinação diferente dos canais R,G,B um número, e usar esse número na matriz da imagem. Essa forma é chamada de representação por tabela de cores ou palette de cores, onde cada um dos números irá indexar uma entrada na tabela. A figura 2.2. mostra uma imagem, sua matriz relacionada e sua tabela de cores.



1	0	0	0	3
0	2	0	0	0
0	3	0	2	0
0	0	1	0	0
1	0	0	3	0

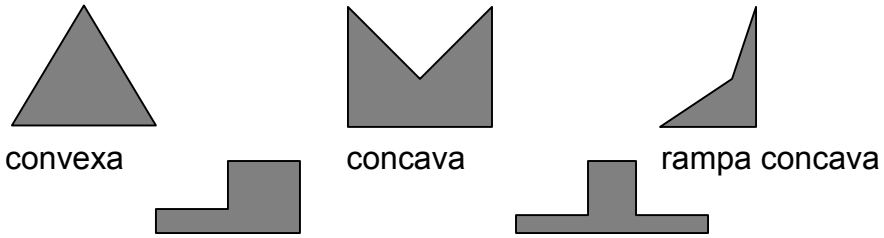


A melhor forma de se trabalhar com uma imagem é quando o valor de sua resolução é de 24bits/pixel. Nessa representação chamada de true-color, cada pixel é representado por 3 bytes, onde cada valor desses bytes está associado a uma cor azul verde ou vermelha, assim podemos trabalhar nos bytes de dados da imagem diretamente, sem precisar acessar a palette de cores. No caso de ser usado um byte por canal, o valor máximo será 255.

Ruídos podem ser causados por erros na transmissão de dados, ou na aquisição da imagem na hora de sua digitalização. Eles também podem ser adicionados propositalmente à imagem para produzir algum efeito especial, ou para o estudo de técnicas de eliminação dos mesmos. Os pixels corrompidos pelo ruído, ou são alterados para o valor máximo, ou tem alguns bits alterados, causando uma diferença brusca de tons entre este pixels e seus vizinhos. Quando os pixels são alternadamente modificados para 0 ou o máximo, este ruído e chamado de ruído salt and pepper, devido a sua aparência.(já que essa adição produz pontos brancos e pretos na imagem)

Filtros, como o nome já diz, são métodos de eliminar ou ajustar alguma informação na imagem, por exemplo aumento de nitidez e aguçamento da imagem. Existem vários tipos de filtros, entre eles os de passa-baixa que são responsáveis por filtrar(retirar) elementos de alta variação espacial, ou seja coisa que não se repetem de vizinhança em vizinhança de pixels, como ruídos.Os filtros passa-alta fazem exatamente o oposto, ou seja aguçam e enfatizam os limites e bordas da imagem.
Além desses métodos existem os métodos que usam derivadas. Esses métodos são baseados nas derivadas de primeira ou Segunda ordem. Os métodos baseados em

gradiente ou de primeira ordem são por exemplo os de roberts, sobel e prewitt. O método de laplaceé um exemplo de utilização da segunda derivada. Mas ambos podem ser usados para detectar descontinuidades, bordas, linhas direcionais e pontos. Essas descontinuidades podem ser classificadas da forma mostrada na figura 2.3:



Mais informações sobre os métodos usados para fazer a derivada das imagens podem ser obtidos em [1][2][3].

Nas seções que seguem descreveremos em detalhes diversas técnicas que são implementadas neste trabalho para o tratamento de imagens. Essas são:

- Separação/Combinação de canais de cores;
- Inversão da Imagem;
- Transformação para tons de cinza;
- Ajuste de contraste;
- Correção gamma;
- Conversão para true-color;
- Adição de ruído;
- Filtragem por convolução;

2.2 – Separação/Combinação de canais de cores (Channel Splitting/Combining)

Essa técnica que consiste na separação de uma imagem colorida, em três imagens, cada uma representando um dos três canais de cores primárias, é muito utilizada quando desejamos tratar ou melhorar um canal de cor por vez. Para isso fazemos a divisão de uma imagem que em seu formato normal era (Red,Green,Blue) em 3 imagens, em tons de cinza, nos formatos (Red,Red,Red) , (Green,Green,Green) e (Blue,Blue,Blue), como é mostrado nas figuras 2.5 a 2.7, que representam os canais da figura 2.4:

Figura 2.4 – Imagem Original 1

Figura 2.5 – Canal Vermelho

Figura 2.7 – Canal Azul

2.3 – Inversão da Imagem (Negative)

Essa técnica também chamda de inversão de tons consiste em inverter os tons de cada canal da imagem. Nela, para cada pixel que inicialmente teria valores (Red,Green,Blue) passa a ser descrita por novos valores dados por (Max – canal), onde Max é o valor máximo possível do canal. Assim se for usado 1 byte por canal, mmax será 255 e a técnica pode ser descrita como (255 – Red,255 – Green,255 – Blue), por exemplo se a cor for (255,0 ,0) um vermelho puro, ao invertermos teremos (0,255,255) que representa a cor ciano. O resultado é o mesmo obtido ao vermos uma foto e seu

negativo. Um exemplo da aplicação dessa técnica implementada em nosso trabalho pode ser visto na figura 2.9, que representa a imagem inversa da figura 2.8.

Figura 2.8 – Imagem Original 2

Figura 2.9 – Imagem Negativa

2.4 – Transformação para tons de cinza (Gray Scale)

Transformar uma imagem em cores para tons de cinza (ou grayscale), consiste em pegar os valores de cada canal, somá-los e dividir por 3, isto é, (R + G +B)/3 e colocar o resultado como o valor tanto d canal R, quando dos canais G e B, assim ao final teremos uma imagem cujo pixel terá o seguinte formato [(R+G+B)/3,(R+G+B)/3,(R+G+B)/(3)], que representa um tom de cinza. Como conseqüência a imagem se apresenta em níveis de cinza (grayscale). A figura 2.11 ilustra esse efeito, representando em escala de cinzas a figura 2.10.

Figura 2.10 – Imagem Original 3

Figura 2.11 – Imagem em Gray Scale

2.5 – Ajuste de contraste (Contrast)

Quando uma imagem é digitalizada os limites de intensidade de luz, conhecidos como contraste pode ser reduzido ou ampliado, por isso, esse filtro vem por intermédio de uma formula simples ajustar o valor do contraste de acordo com a porcentagem dada pelo usuário. Existem muitas fórmulas possíveis para o ajuste do contraste, mas a fórmula escolhida em nossa implementação é:

$$X = \text{Min}\left(128 + \frac{(\text{Canal} - 128)*\text{PERC}}{100}, 255\right)$$, onde PERC é o percentual dado pelo usuário.

Y = Max (X,0) e em seguida Canal = Y.

Como pode ser obaservado na comparação entre as figuras 2.12 e 2.13, a aplicação do contraste melhora muito a nitidez da imagem.

Figura 2.12 – Imagem Original 4

Figura 2.13 – Imagem com Contrast (PERC = 250)

2.6 – Correção Gamma (Gamma Correction)

A correção gamma é uma correção de intensidade de canais baseada num fator gamma (x) dado pelo usuário e atua sobre os canais pela formulada dada abaixo:

$$MaxRange = \frac{255^{x}}{255}, \quad Z = \frac{canal^{x}}{MaxRange}, \quad Y = \min(Z,255)$$

$$canal = \max(Y,0)$$

Como pode ser observado pela comparação entre as figuras 2.14 e 2.15, essa expressão também resulta no aumento do contraste.

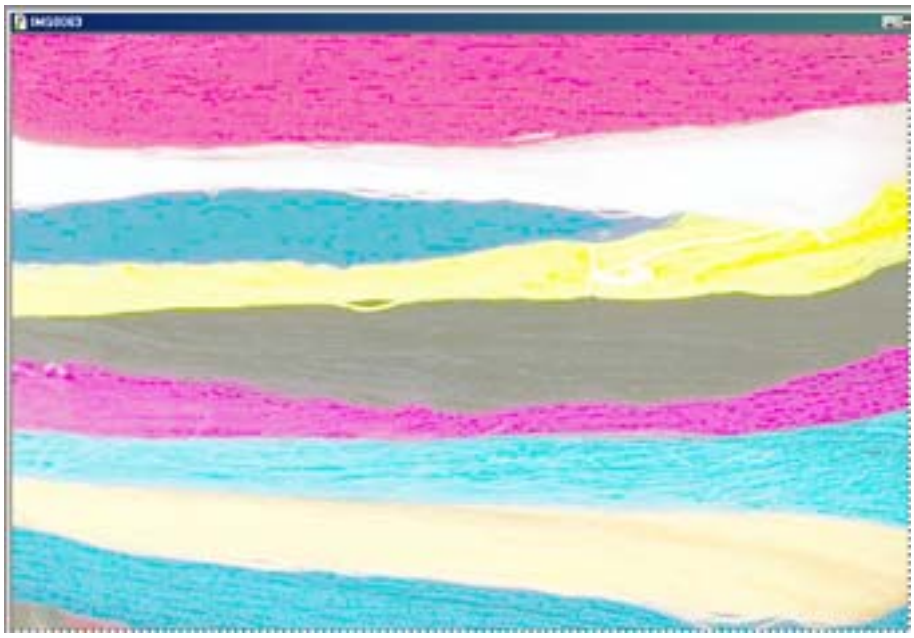


Figura 2.14 – Imagem Original 5

Figura 2.15 – Imagem com Gamma Correction (X = 4)

2.7 – Conversão para True-Color (Increase Color Depth)

Esse processo apenas faz com que o mapeamento das cores que existia na tabela de cores (ou palette) seja passado para o próprio pixel, desta forma ele não mais referencia a tabela para descobrir qual a cor representa. E passa a ser associado a valores (R,G,B) do próprio ponto. Na imagem nada muda visualmente, só a sua representação será alterada.

2.8 – Adição de ruídos (Add Noise)

Esse processo serve pra adicionar um ruído randômico à imagem. Para a adição desse ruído o usuário passa uma porcentagem, que representa a porcentagem da imagem que será corrompida pelo ruído. Para isso usamos a seguinte formula :

Test = random( ) (onde random é uma função que gera números randômicos)
Caso Test seja maior que RAND_MAX*PERC , (onde RAND_MAX é uma constante
100 cujo valor é o maior valor possível
para um número randômico e
PERC é a porcentagem passada
pelo usuário)
o valor do canal será dado pela seguinte fórmula:

$$X = \frac{random()*256}{RAND\_MAX} - 128$$

Y = Min (Canal + X, 255)

Canal = Max (Y, 0)

Caso contrário o valor continuará o mesmo.

O resultado desse efeito pode ser visto nas figuras 2.17 e 2.18 que representam a figura 2.16 com 15% e 25% de ruído consecutivamente.

Figura 2.16

figura 2.17

figura 2.18

2.9 – Filtros de Média 3x3/5x5/7x7 (Mean)

Os filtros de média são filtros simples utilizados para redução de ruídos ou pré-processamentos para processos futuros tal como segmentação. O Filtro de média consiste em pegar todos os valores dos pixeis vizinhos e fazer uma média aritmética sobre estes valores, no qual o resultado será o valor associado ao pixel corrente[IAN_98]. Esses filtros são do tipo passa-baixa, ou seja, suavizam (diminuem o contraste) da imagem.Assim uma característica nos filtros de média é que eles eliminam pequenos detalhes da imagem, efeito conhecido como bluring, que literalmente borra a imagem e quanto maior a máscara maior o efeito do bluring. Um exemplo de máscara 3x3 é a seguinte :

$$K = \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Veja nas figuras 2.20 a 2.22 alguns exemplos onde utilizamos um filtro de média com janleas de 3x3, 5x5, 7x7, todas aplicadas sobre a figura 2,19 onde utilizamos um filtro de média.

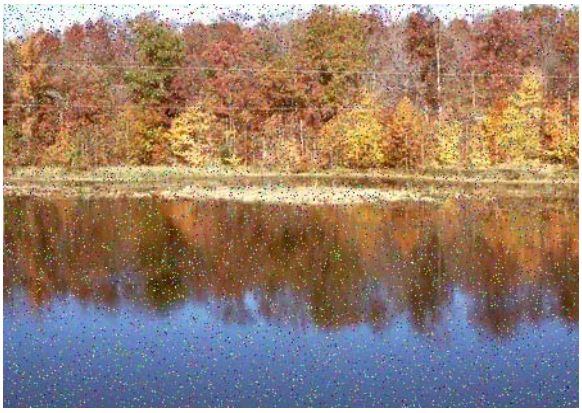


figura 2.19


figura 2.20

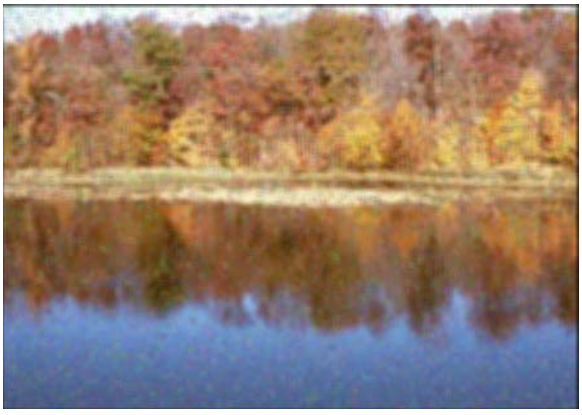


figura 2.21

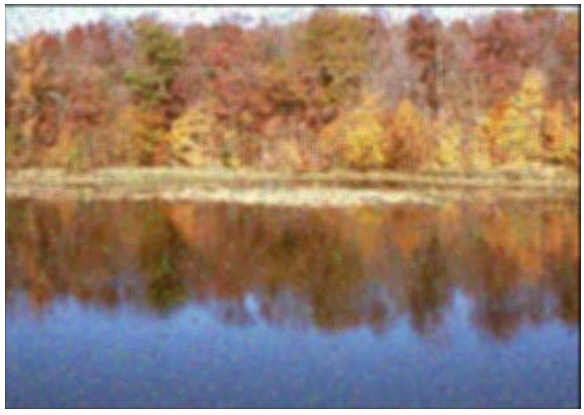


figura 2.22

2.10 – Filtros de Mediana 3x3/5x5/7x7 (Median)

Os filtros de mediana também são filtros passa-baixa, utilizados para suavizar, ou seja, reduzir imperfeições das imagens , como por exemplo ruídos. Para esse filtro os valores de pixels de uma dada vizinhaça, em torno do pixel em análise, são ordenados em ordem crescentes ou decrescente. O valor central (mediana) desta ordenação é então utilizado para substituir o valor do pixel corrente[IAN_98]. A vantagem dessa técnica sobre a de média é que ela não elimina os detalhes, desta forma a imagem mantém o mesmo contraste.[4]

Compare nas figuras 2.24 a 2.26 os efeitos do uso do filtro de mediana 3x3, 5x5 e 7x7 sobre a figura 2.23

figura 2.23

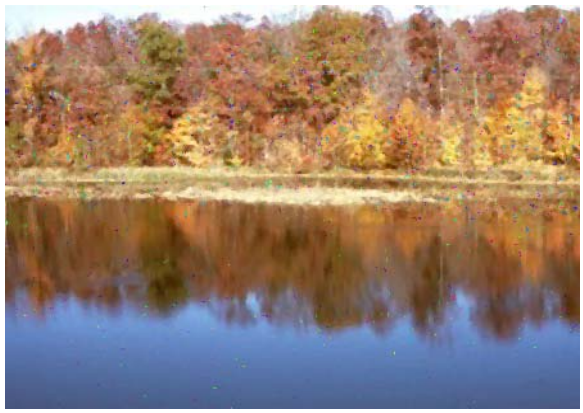


figura 2.24

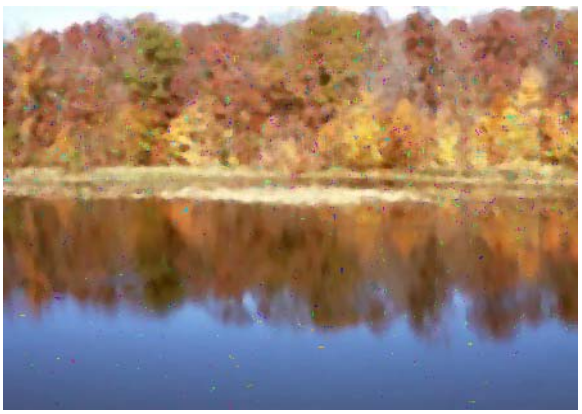


figura 2.25

figura 2.26

2.11 – Filtros Passa-Alta (Sharpening Balanced 1 e 2 e Unbalanced)

Os filtros de ampliação de contraste, passa-alta ou de sharpening são filtros de aguçamento de detalhes em imagens, podem reduzir e até remover o efeito de bluring. No caso deste projeto implementamos 3 filtros desse tipo. 2 filtros são balanceados, ou seja, tem somatório dos pesos é igual a 1, mantendo assim a intensidade média.Um filtro é desbalanceado, ou seja, apresenta somatório dos pesos maior que um. Veja como são as máscaras usadas em cada um :

$$k = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 6 & -1 \\ 0 & -1 & 0 \end{bmatrix} \qquad k = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{bmatrix} \qquad k = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

Unbalaced Balanced 0 Balanced 1

(nomenclarura usada no menu de implementação)

As figuras 2.28 a 2.30 permitem comparar as peculiaridades da aplicação desse filtro em uma mesma imagem, a figura 2.27

figura 2.27

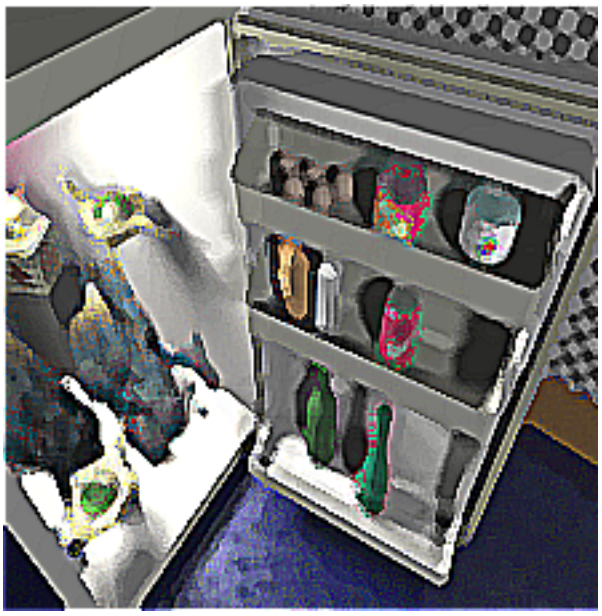


figura 2.28

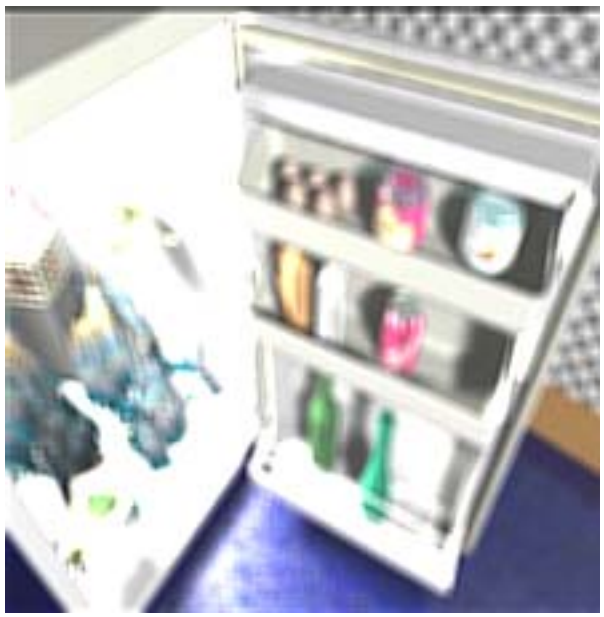


figura 2.30

2.12 – Filtro Sobel Direcional Horizontal/Vertical (Sobel)

O filtro de sobel representa as derivadas primeira na vertical ou na horizontal de uma imagem, elas são um bom exemplo de que é possível combinar mais de um operador em uma única mascara.O método de Sobel para detecção de bordas, que combina as operações de diferenciação vertical e horizontal com um pouco de *smoothing* para salientar os contornos de uma imagem.[5][6]

A mascará é a seguinte :

$$K=\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \qquad K=\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

Vertical Horizontal

Os resultados obtidos com a aplicação destes um uma imagem podem ser vistos nas figuras 2.32 e 2.33 que representam a filtragem da figura 2.31 na horizontal e na vertical consecutivamente.

figura 2.31

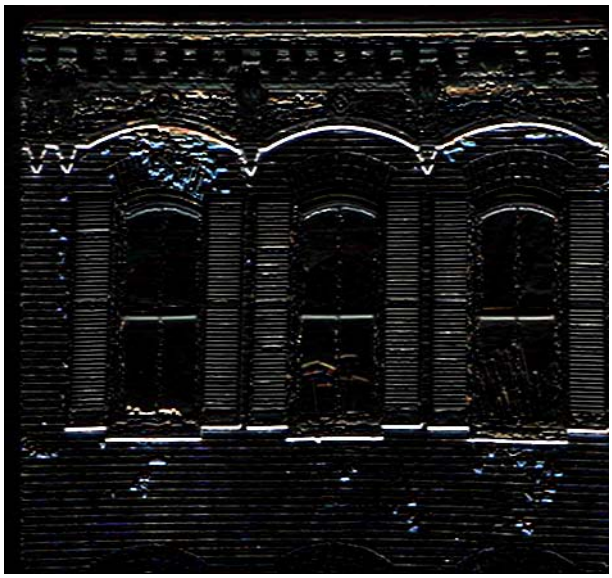


figura 2.32

figura 2.33

2.13 – Filtro Direcional de Roberts (Roberts)

O filtro passa-alta de Roberts é um filtro de gradiente direcional que detecta regiões de alta freqüência espacial, que por exemplo correspondem a bordas. Ele enfatiza bordas com orientação em 45º e 135º. Uma das vantagens desse método é que ele é bem rápido pelo fato de usar apenas 4 pixel para seus cálculos. Uma das suas desvantagens é que por ser muito pequeno ele é muito sensível a ruídos.Seus resultados são bordas bem finas. Esse filtro tem a seguinte máscara de convolução.[7] Suas máscaras 2x2 são representadas a seguir :

$$K=\begin{bmatrix}1 & 0\\ 0 & -1\end{bmatrix} \qquad K=\begin{bmatrix}0 & 1\\ -1 & 0\end{bmatrix}$$

135º 45º

As figuras 2.35 e 2.36 ajudam a entender os efeitos do filtro direcional de Roberts aplicado sobre a figura 2.34

figura 2.34

figura 2.35

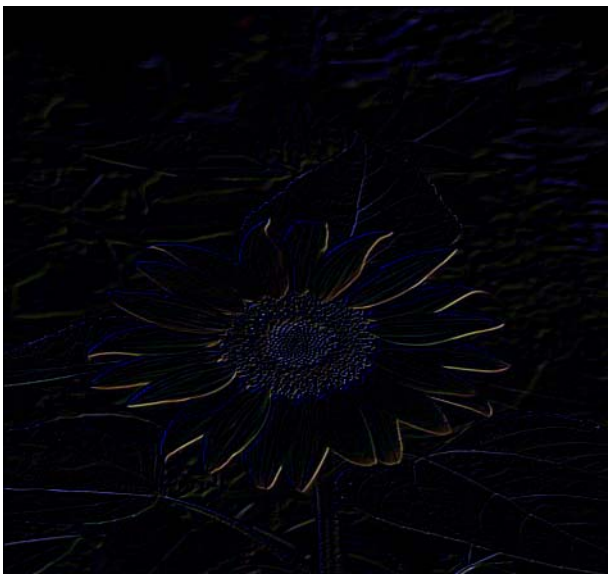


figura 2.36

2.14 – Filtros Passa-Alta de Prewitt (Prewitt)

Os filtros de Prewitt são geralmente formados por 8 máscaras de convolução, uma para cada direção de 45º em 45º graus, isto é, 0º , 45º, 90º,135º,180º,225º,270º, 315º. Assim como os outros filtros de gradiente, este filtro é utilizado para detectar bordas. Veja a seguir como são suas máscaras de convolução: [8]

$$\begin{bmatrix} -1 & 1 & 1 \\ -1 & -2 & 1 \\ -1 & 1 & 1 \end{bmatrix} \quad \begin{bmatrix} -1 & -1 & 1 \\ -1 & -2 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad \begin{bmatrix} -1 & -1 & -1 \\ 1 & -2 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad \begin{bmatrix} 1 & -1 & -1 \\ 1 & -2 & -1 \\ 1 & 1 & 1 \end{bmatrix} \quad \begin{bmatrix} 1 & 1 & -1 \\ 1 & -2 & -1 \\ 1 & 1 & -1 \end{bmatrix} \quad \begin{bmatrix} 1 & 1 & 1 \\ 1 & -2 & 1 \\ 1 & 1 & -1 \end{bmatrix} \quad \begin{bmatrix} 1 & 1 & 1 \\ 1 & -2 & 1 \\ -1 & -1 & -1 \end{bmatrix} \quad \begin{bmatrix} 1 & 1 & 1 \\ 1 & -2 & 1 \\ -1 & -1 & 1 \end{bmatrix}$$

0º 45º 90º 135º 180º 225º 270º 315º

As figuras 2.37 a 2.45 mostram os resultados obtidos pela aplicação dessas máscaras em uma imagem.

figura 2.37

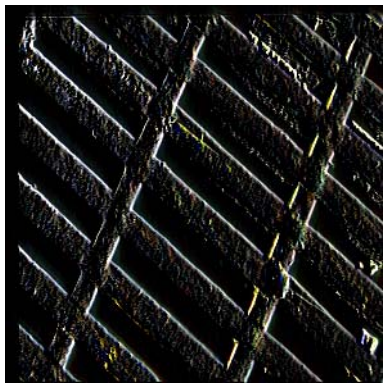


figura 2.38

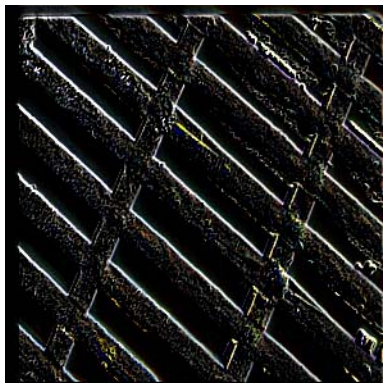


figura 2.39

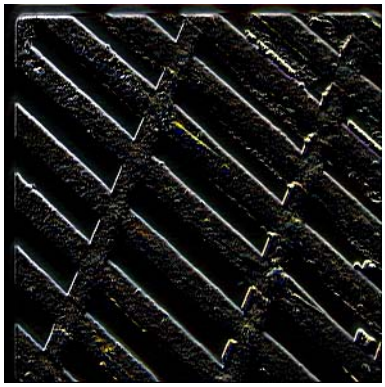


figura 2.40

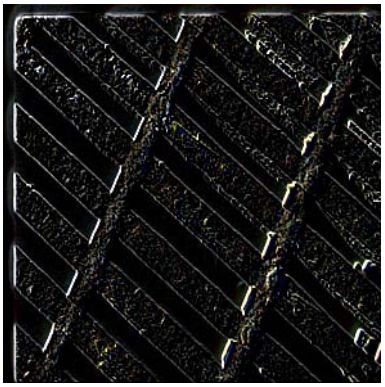


figura 2.41

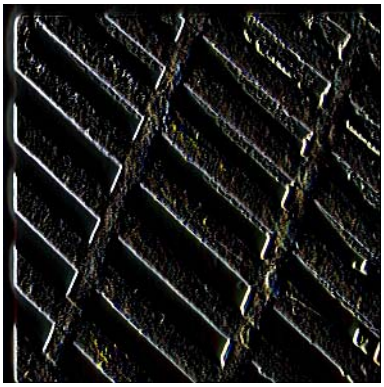


figura 2.42

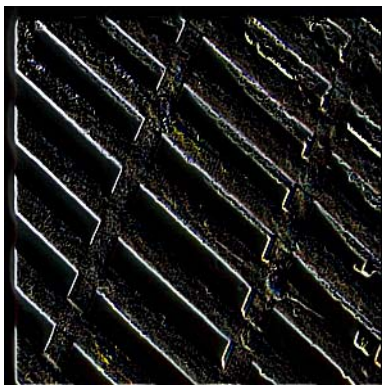


figura 2.43

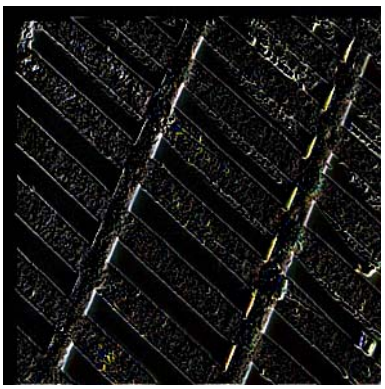


figura 2.44

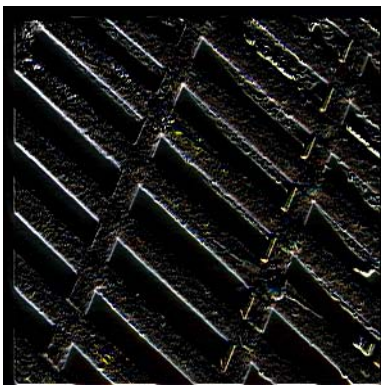


figura 2.45

2.15 – Filtrogem definida pelo usuário (User Defined Filter)

O usuário pode definir seu próprio filtro, no formato máximo de mascará de 7x7. Nesta opção o usuário define os pesos a serem aplicados na operação de convolução da imagem. Obviamente o resultado obtido depende dos pesos usados.

Nas figuras 2.46 e 2.47 temos exemplos de possíveis máscaras a serem aplicadas.

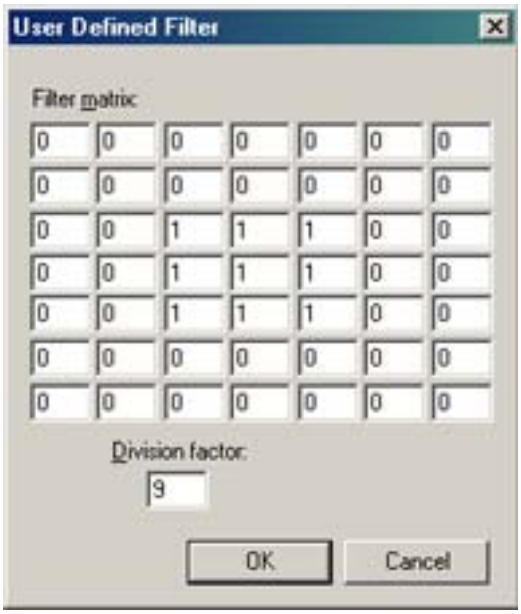


figura 2.46

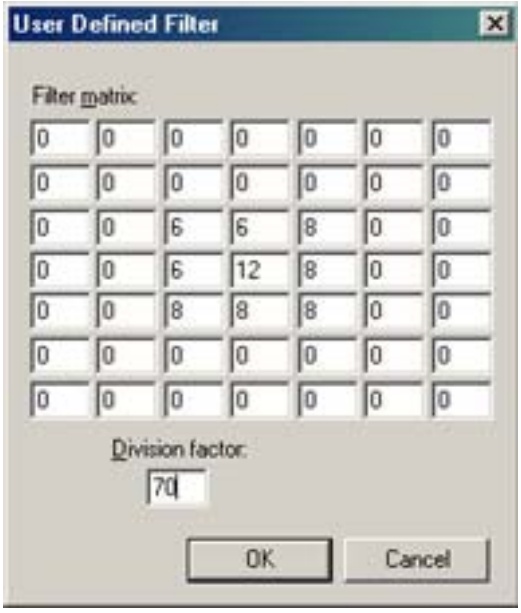


figura 2.47

Agora que já vimos exemplos de aplicações dos diversos processos implementados no nosso projeto, vamos entrar um pouco no detalhamento da implementação visual do programa. No próximo capítulo vamos ver alguns menus e componentes utilizados no programa.

CAPITULO 3 – O programa implementado

3.1 – Introdução

Os exemplos do capítulo anterior foram todos executados no sistema implementado nesse trabalho. Nesse capítulo apresentaremos mais detalhes da internface desse sistema. O programa foi implementado na liguacem c (Visual c++ 6.0) e utiliza imagens do tipo bitmap.

Ao iniciar o aplicativo o usuário se depara com a interface observada na figura 3.1.

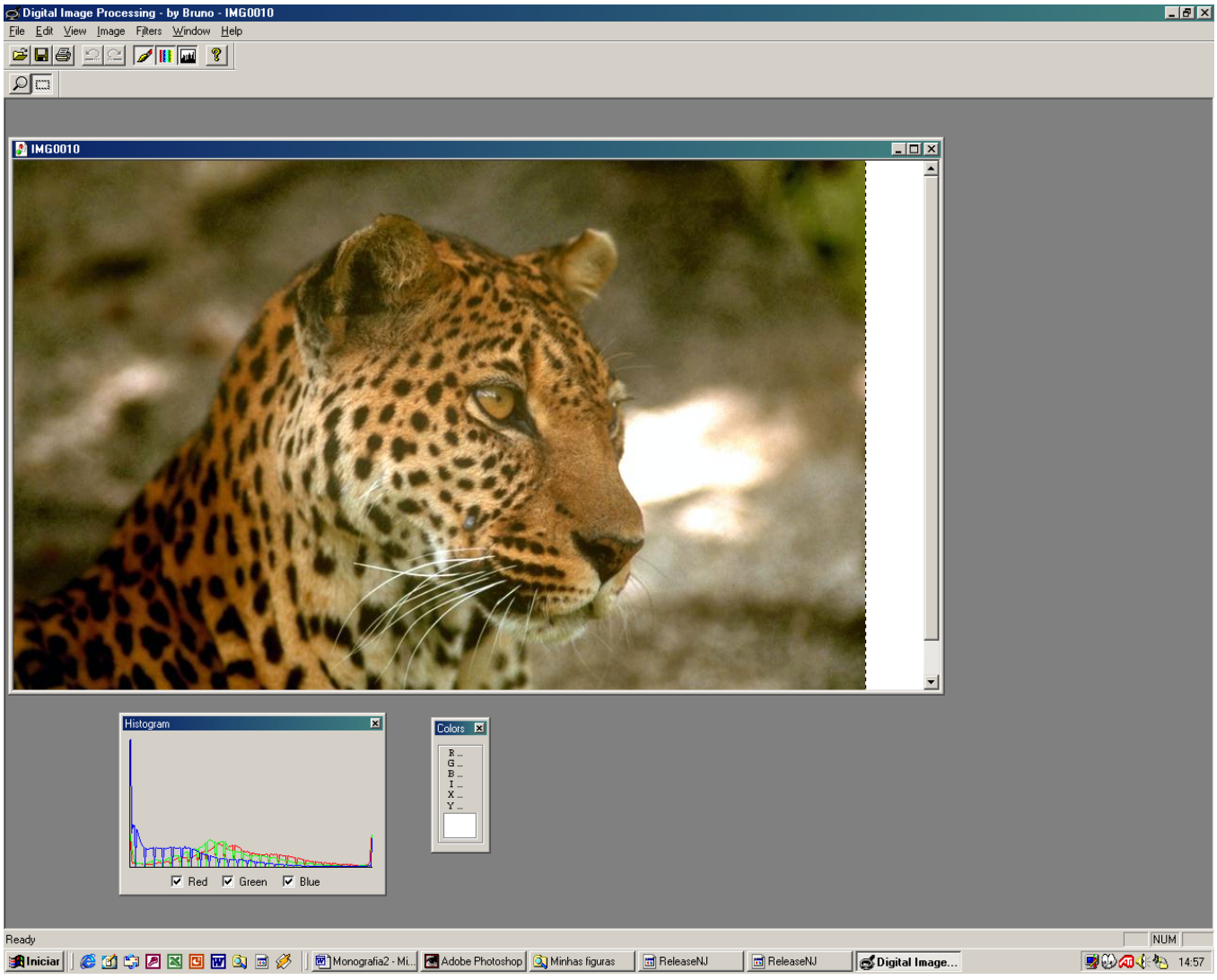


figura 3.1

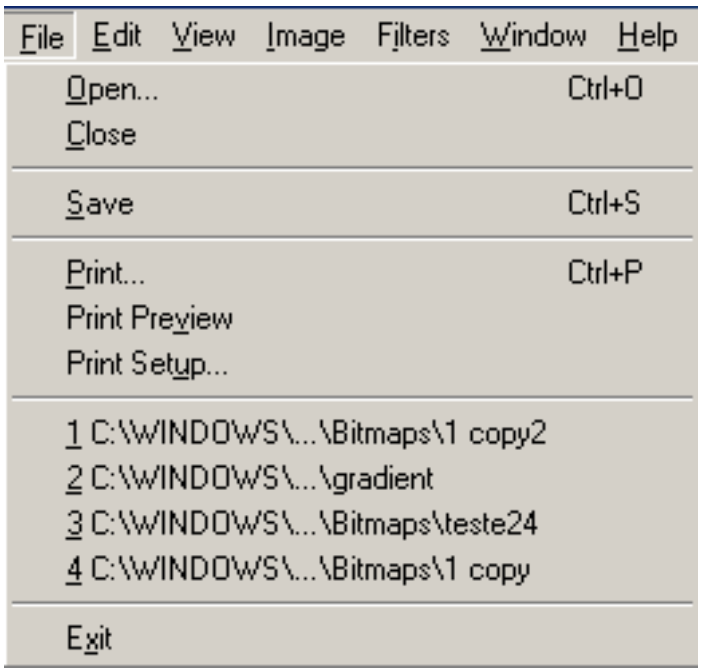


Figura 2.1 – Menu File

3.2 - Menu File

Neste menu (figura 3.2) você encontra as ações principais relacionadas aos arquivos, tais como:

Open – Abre um arquivo do tipo .BMP (Bitmap), que será discutido no apêndice I.
Close – Fecha o arquivo em uso.
Save – Salva o arquivo em uso, atualizando as alterações feitas.
Print – imprime o arquivo em uso.
Print Preview – Exibe uma amostra do arquivo para que se tenha uma noção da sua posição na folha.
Print Setup – Configura as opções da impressora para melhores resultados na hora da impressão.
1, 2, 3, ... – exibe os 10 (ou menos) últimos arquivos utilizados.
Exit – Sai do programa .

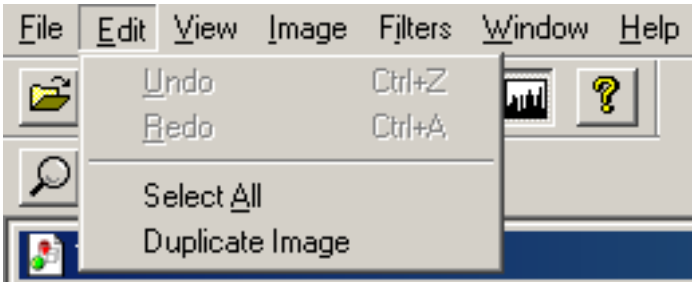


Figura 3.3 – Menu Edit

3.3 - Menu Edit

Neste menu (figura 3.3) você encontra algumas das opções relativas a ações feitas sobre a imagem, como undo e redo e as opções selecionar tudo e duplicar a imagem. As opções de Undo e Redo possuem 10 passos intermediários:

Undo – Desfaz a última ação realizada sobre a imagem.
Redo – Refaz a última ação realizada sobre a imagem.
Select All – Seleciona toda a área da imagem.
Duplicate Image – faz uma copia exata da imagem.

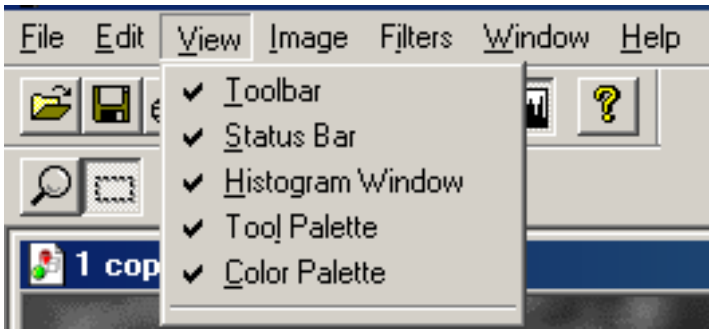


Figura 3.4 – Menu View

3.4 - Menu View

Nesse menu (figura3.4) você tem a opção de manter visível ou não um determinado componente do programa :

Toolbar – pode manter visível ou esconder a barra de ferramentas mostrada na figura 3.5.

Figura 3.5 – Toolbar

Status Bar – pode manter visível ou esconder a barra de status mostrada na figura 3.6.

Figura 3.6 – Status Bar

Histogram Window – pode manter visível ou esconder a janela do histograma mostrada na figura 3.7.O histograma foi implementado de forma a manipular os 3 canais de cores simultaneamente, porém o usuário pode escolher visualizar um canal por vez desmarcando os canais não desejáveis.

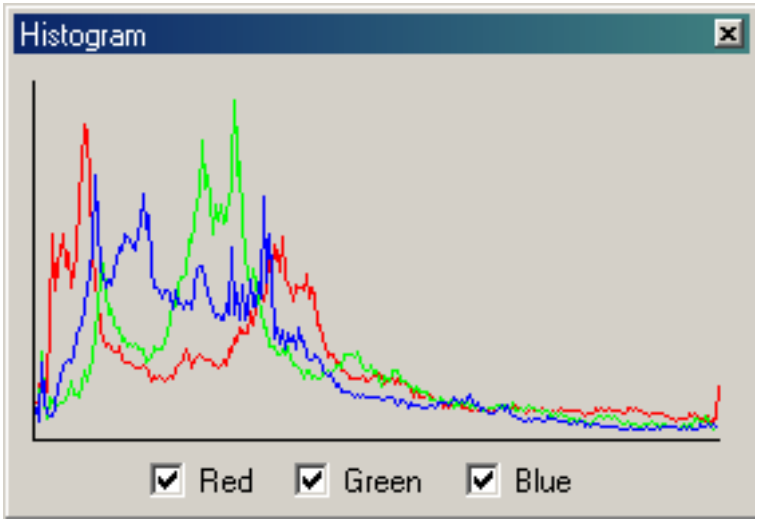


Figura 3.7 – Histogram Window

Tool Palette – pode manter visível ou esconder a palette de ferramentas mostrada na figura 3.8. Essa palette possui dois itens, o de zoom e o de seleção de área. Nas figuras 3.9 a 3.11 temos exemplos da utilização desses dois itens :


Figura 3.8 – Tool Palette

Color Palette – pode manter visível ou esconder a janela com a palette de cores mostrada na figura 3.11. Essa palette é muito útil quando desejamos saber alguma informação relativa a cor de um pixel numa dada posição (x,y) além mostrar o valor da intensidade, calculado por (R+G+B) / 3.

Figura 3.9 – Color Palette

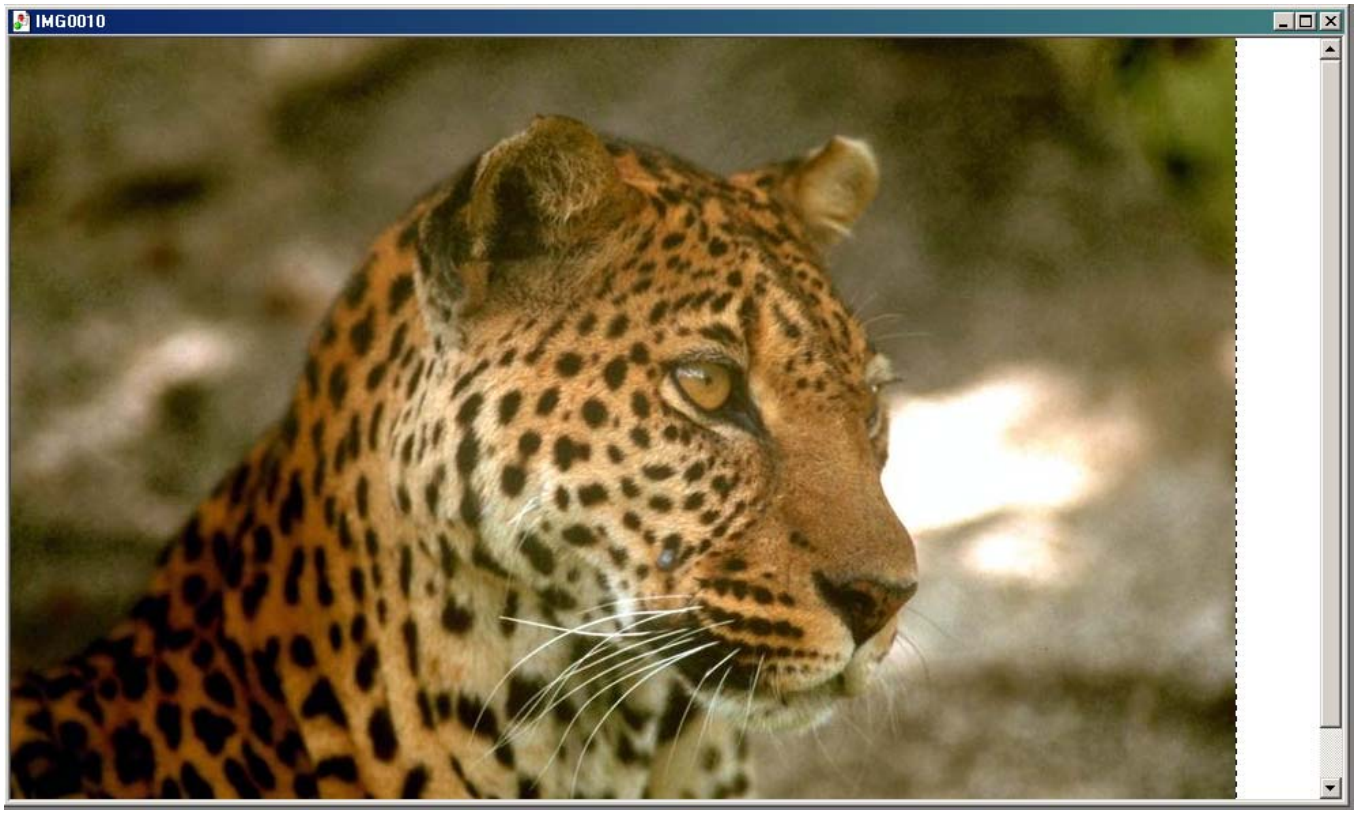


figura 3.9

figura 3.10

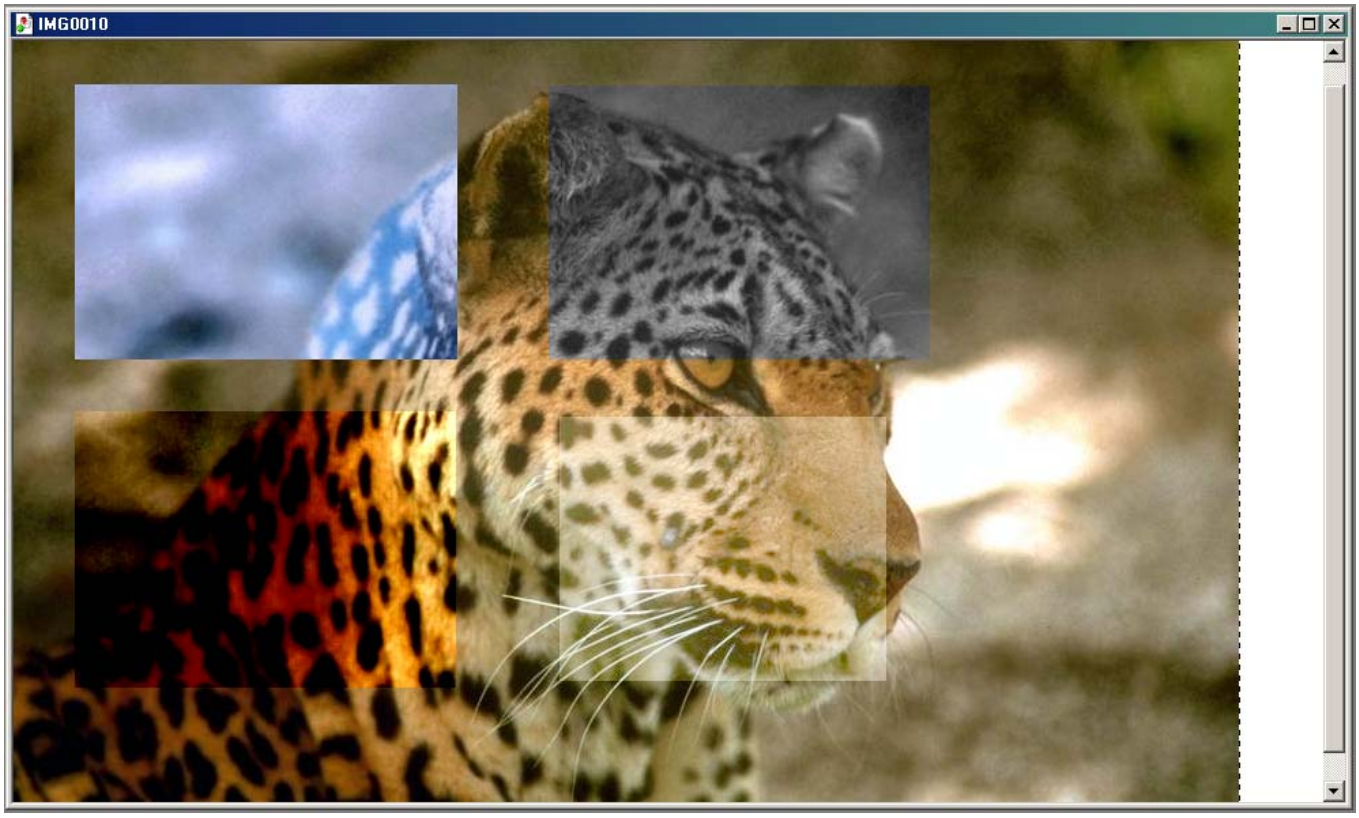


figura 3.11

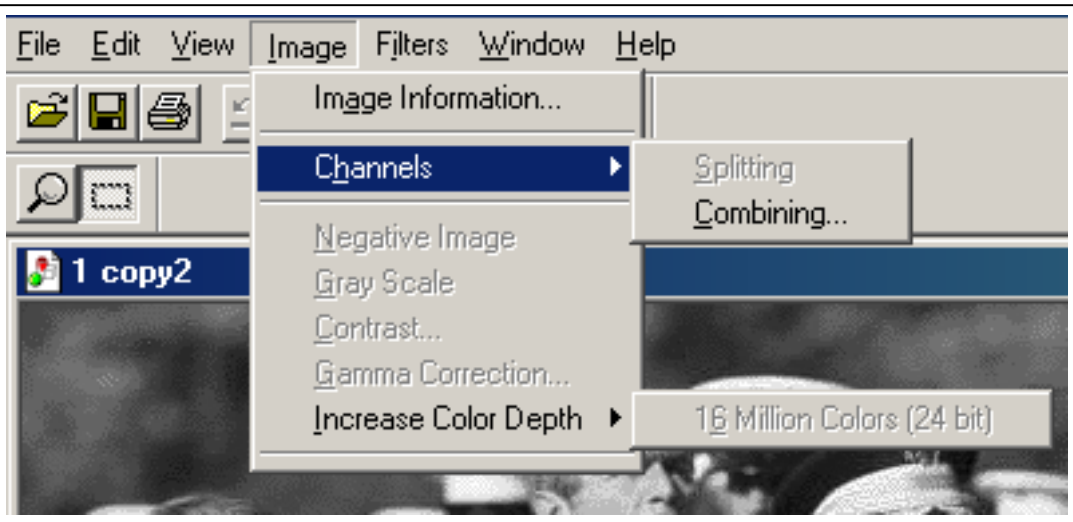


Figura 3.12 – Menu Image

3.5 - Menu Image

Neste menu (figura 3.12) você encontra algumas das diversas opções de funções aplicadas às imagens :

Image Information – exibe uma janela algumas informações relativas a imagem, tais como altura, largura e quantidade de bits por pixel. Essa janela é mostrada na figura 3.13.

Channels – aqui você tem a opção de dividir as imagens por canais (R,G,B) para trabalhar separadamente e depois juntá-los.

Negative Image – Inverte as cores da imagem, muito útil quanto tentamos evidenciar coisas nas imagens.

Gray Scale – transforma a imagens para que fique apenas na escala de cinzas (256 tons de cinza)

Contrast – aqui você pode ajustar o contraste para obter uma imagem mais nítida.

Gamma Corretion – aqui você pode ajustar o contraste na forma de uma função de correção gamma para que a imagem fique mais nítida.

Increase Color Depth – você pode, e deve, aumentar o numero de cores para que a imagem fique com 24 bits por pixel e a maioria dos filtros possa ser aplicada.

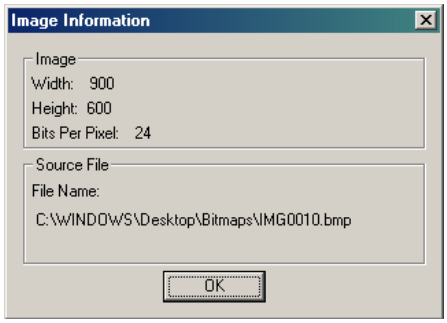


figura 3.13

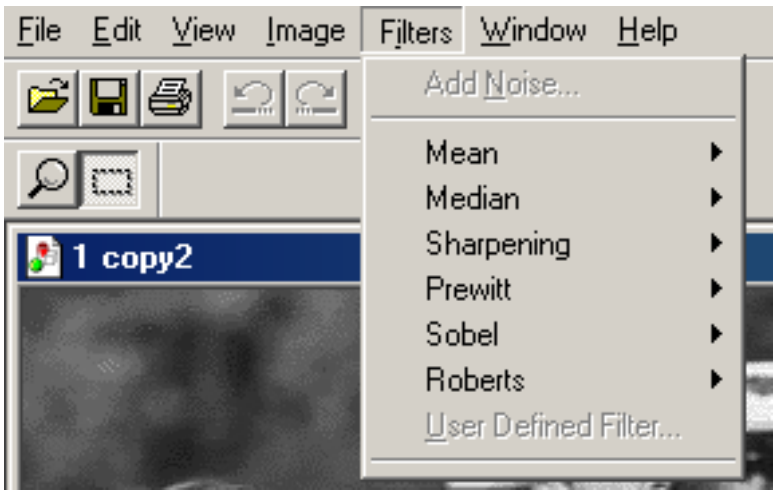


Figura 3.14 – Menu Filters

3.6 - Menu Filters

Neste menu (figura 3.14) você encontra todos os filtros, onde alguns deles podem ser usados para a melhoria da imagem:

Add Noise – adiciona ruído a imagem, muito útil no teste de filtros de suavização e remoção de ruídos.
Mean – Filtros de média 3x3 , 5x5 e 7x7.
Median - Filtros de mediana 3x3 , 5x5 e 7x7.
Sharpening – Filtros de aguçamento , balanceados e não balanceados.
Prewitt - Filtros de Prewitt N,NE,E,SE,S,SW,W,NW.
Sobel – Filtros de Sobel H, V.
Roberts – Filtros de Roberts H, V.
User Defined Filter – Filtro customizado pelo usuário, cuja mascara pode variar desde 3x3 até 7x7.

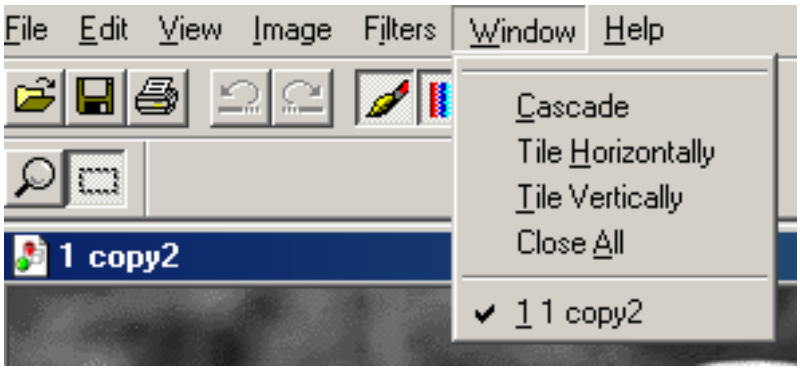


Figura 3.15 – Menu Filters

3.7 - Menu Windows

Neste menu (figura 3.15) você encontra opções de disposição das janelas abertas.

Cascade – exibe as janelas em cascata.
Tile Horizontally – exibe as janelas lado a lado horizontalmente.
Tile Vertically - exibe as janelas lado a lado verticalmente.
Close All – Fecha todas as janelas.

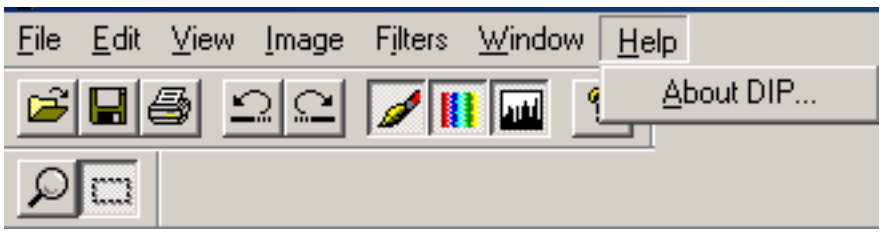


Figura 3.16 – Menu Help

3.8 - Menu Help

Neste menu (figura 3.16) você encontra algumas informações sobre o programa, A figura 3.17 mostra tais informações.

Figura 2.12 – About Dialog

Agora que já vimos o programa implementado vamos ver alguns exemplos práticos que são usados no-dia a-dia. No próximo capítulo iremos comparar exemplos de processamentos feitos por outros programas com a nossa implementação e assim mostrar a eficiência deste projeto.

CAPITULO 4 – Exemplos

Neste capítulo mostraremos áreas de aplicação das técnicas de processamento e melhoria de imagens digitais:

– Médica
  - Classificação cromossomial
  - Análise de células sanguínea
  - Análise da radiografia torácica
  - Tomografia computadorizada
  - Radiografia digital

– Sensoriamento Remoto
  - Estudo do uso e recursos da terra
  - Detecção de incêndios florestais
  - Movimento de icebergs
  - Previsão do tempo

– Geologia
  - Exploração de óleo e minério
  - Previsão de abalos sísmicos

– Oceanografia
  - Análise da superfície oceânica
  - Análise das placas tectônicas

– Astronomia
  - Estudo da atmosfera
  - Padrões de Brilho das estrelas

Agora vamos ver alguns exemplos práticos:

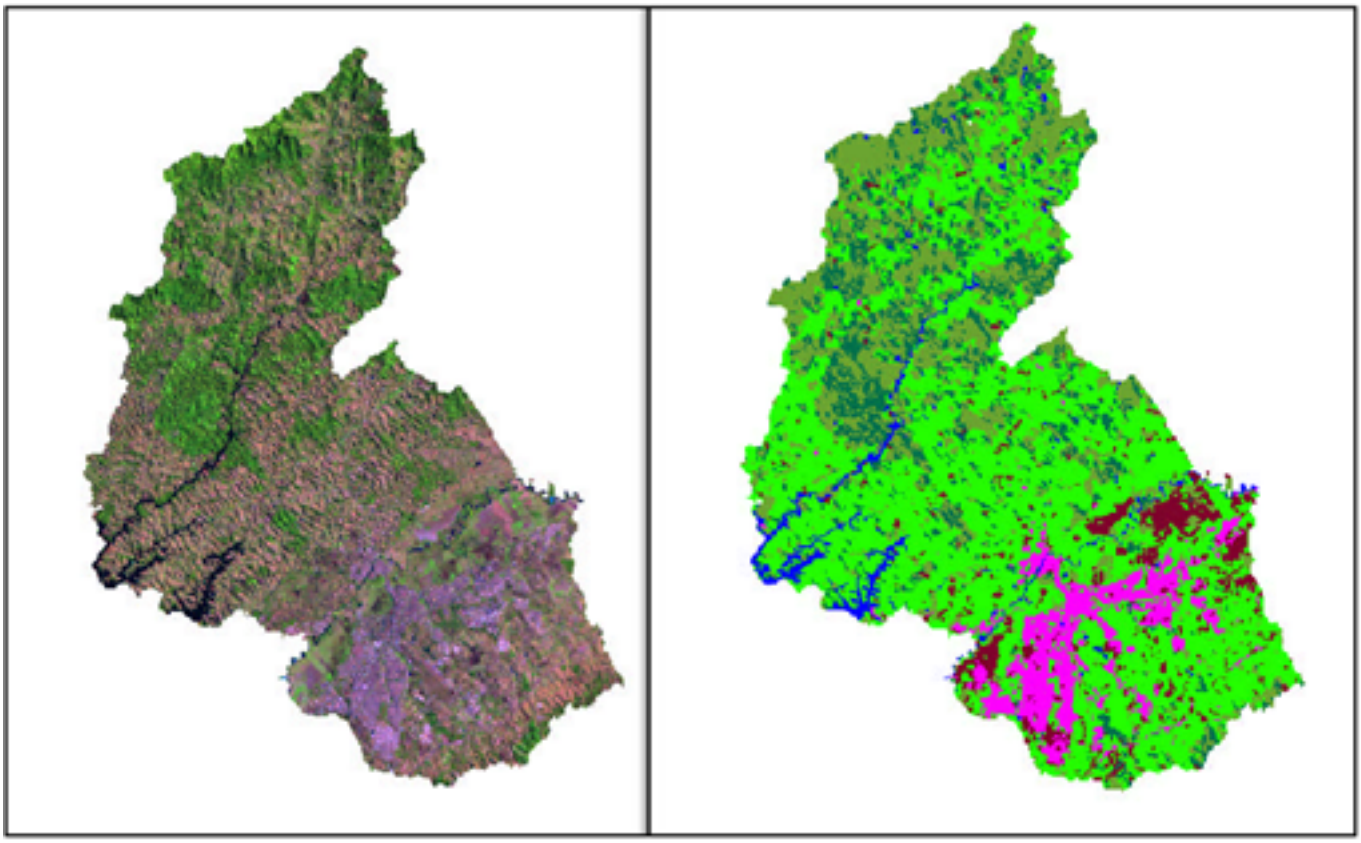


figura 4.1

Na figura 4.1, temos uma imagem de satélite mostrando uma região florestal. A técnica utilizada foi a da segmentação que possibilitou a extração de maiores informações, como a existência de um incêndio [9][10][11].

figura 4.3

figura 4.4

figura 4.5

Na figura 4.3 temos uma imagem obtida através de satélite, onde passamos um filtro de aguçamento para enfatizar os detalhes e remover o efeito de bluring, gerando o resultado mostrado na figura 4.4.Resultado semelhante foi obtido através do uso da nossa implementação, mostrado na figura 4.5[12].

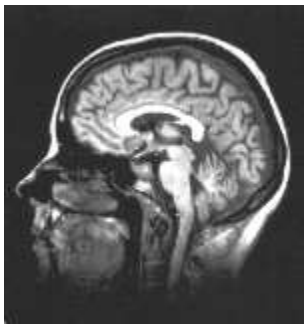


figura 4.6

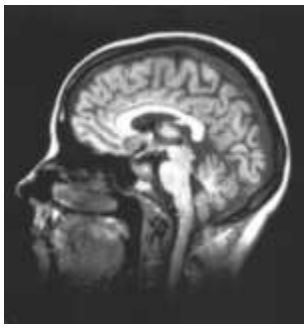


figura 4.7

figura 4.8

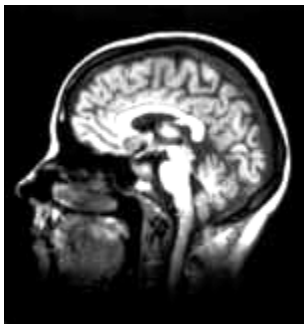


figura 4.9

figura 4.10

Nas figuras 4.6 a 4.8 temos a imagens de uma tomografia computadorizada, onde foi aplicado um alargamento de contraste e em seguida obteve-se o negativo dessa imagem. Nas figuras 4.9 e 4.10 temos resultado semelhante ao esperado utilizando-se nosso aplicativo[13].

No exemplo a seguir (figuras 4.11 a 4.12) temos outra imagem de tomografia, Porém o problema que tem de ser resolvido é o ruído, além de passar um filtro de mediana , neste exemplo temos um ajuste de contraste. Na figura 4.12 temos o resultado obtido no uso de nosso programa.[14]

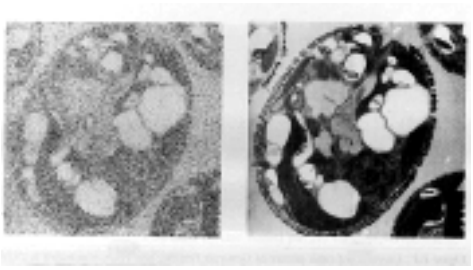


figura 4.11

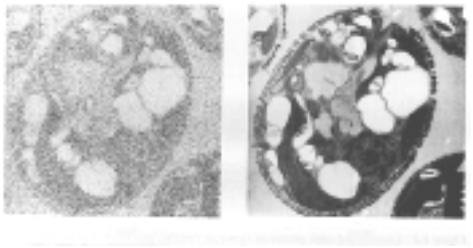


figura 4.12

Aqui temos um exemplo muito comum que é o ajuste de contraste e correção gamma sobre imagens de radiografias digitais. Na figura 4.13 temos o exemplo onde o programa utilizado foi o photoshop e na figura 4.14 temos o resultado obtido pela execução na nossa implementação, onde conseguimos pegar ainda mais detalhes[15].

figura 4.13

figura 4.14

No próximo exemplo temos a aplicação do filtro de aguçamento para remoção do efeito de bluring possibilitando reconhecer a pessoa. Muitas vezes essa técnica é utilizada para identificar suspeitos que foram gravados durante um assalto ou placas de carros. A figura 4.15 mostra o resultado desse processo, que pode ser comparado ao resultado obtido pelo processamento feito em nossa implementação, visto na figura 4.16.

figura 4.15

figura 4.16

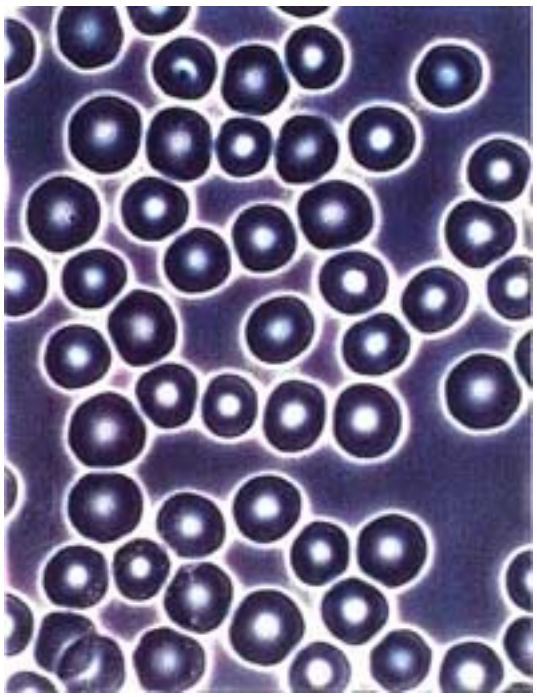


figura 4.17

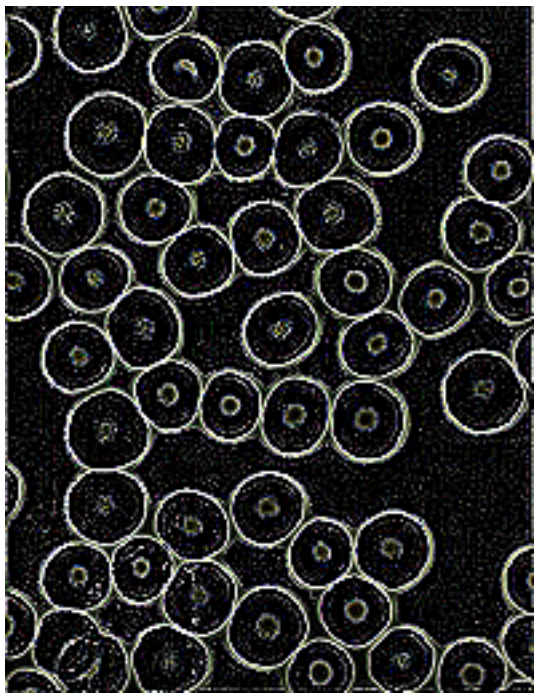


figura 4.18

No exemplo de análise de células sanguineas, mostrado na figura 4.17 e 4.18, as mudanças no formato da borda podem indicar um tumor ou infecção, por isso o filtro mais adequado aqui é o filtro de detecção de bordas. Resultado semelhante poderia ser obtido em nosso programa pela junção de imagens onde filtros direcionais tivessem sido passados, porém esta implementação ainda não disponiviliza adição de imagens.

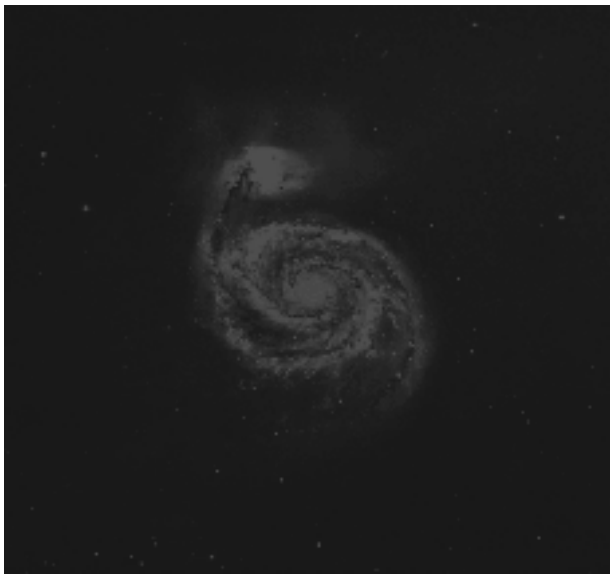


figura 4.19

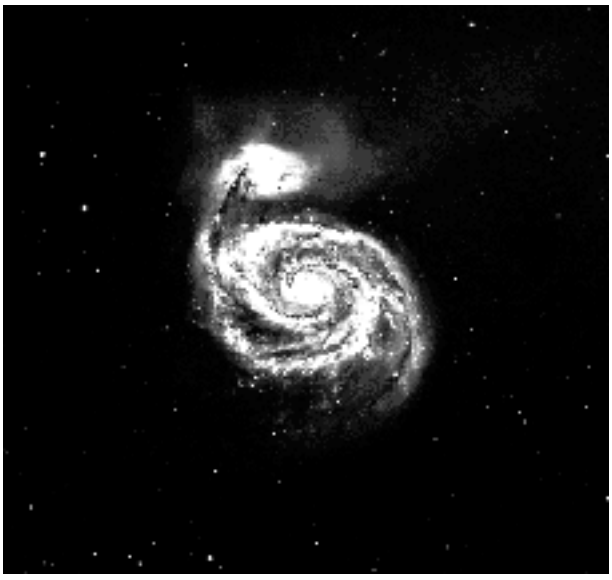


figura 4.20

figura 4.21

Aqui temos mais um exemplo do uso da correção de contraste para que uma análise de uma galaxia possa ser pesquisado e de acordo com as intensidades possa ser identificada sua idade. A figura 4.19 e 4.20 representam o antes de depois de uma filtragem feita por outro programa que não o nosso. E o resultado mostrado na figura 4.21 foi obtido pelo uso de nossa implementação.

No próximo capitulo estaremos dando uma avaliação final sobre o trabalho e últimos comentários.

CAPITULO 5 – Conclusão

Apêndice I – Formato de arquivo BITMAP

[16][17][18]

Introdução :

Aqui vou descrever um pouco sobre os formatos de imagem Bitmap utilizados pelo windows e IBM/OS2.

Arquivos BMP são armazenados no formato DIB (Device-Independent Bitmap) que permite exibir a imagem em qualquer dispositivo, ou seja, o bitmap especifica a cor do pixel numa forma independente do método usado pelo dispositivo para representá-la. A extensão padrão dos arquivos DIB do Windows é ".BMP". Referências a arquivos DIB são em última análise referências a arquivos BMP.

O BMP usa Formato Posicional, onde o significado do byte depende de sua posição no arquivo, o que dificulta mudanças em sua estrutura.

Os arquivos bitmap são guardados de baixo para cima, ou seja as primeiras linhas da imagem são as últimas da área de dados do bitmap.

Compressão :

É muitíssimo raro, mas arquivos formato BMP podem, nas versões de 4 e 8 bits/pixel, utilizar a compressão RLE (Run length encoded), de forma a reduzir o tamanho do arquivo que armazena o Bitmap. Como a compressão RLE é raramente implementada, mesmo para imagens de 4 e 8 bits/pixel, os arquivos BMP em geral tendem a ocupar mais espaço em disco do que outros formatos. Logo, se o tamanho do arquivo é importante, o formato BMP não é o mais indicado. Fato que inviabiliza o uso de BMP em aplicações para Internet, que requer, sempre que possível, arquivos pequenos, para maior eficência.

Vale resaltar que a técnica de compressão RLE é usada neste formato somente até 256 cores, por isto arquivos com Bit/Pixel > 8 não podem usá-la. Assim arquivos BMP de 24 (true color) e 32 bits podem ser excessivamente grandes. O formato BMP usado no OS/2 não usa nenhuma técnica de compressão.

Estrutura do Arquivo BITMAP :

a) *Cabeçalho de arquivo*
Contém a assinatura BM e informações sobre o tamanho e lay-out do arquivo BMP (disposição dos dados dentro do arquivo).

b) *Cabeçalho de mapa de bits*
Contém as informações da imagem contida no arquivo. Define as dimensões, tipo de compressão (se houver) e informações sobre as cores da imagem.

c) *Paleta ou mapa de cores (opcional)*
Somente estará presente em arquivos de imagens que usam 16 ou 256 cores (4 e 8bits/pixel). Nas demais, em seu lugar, vem diretamente a parte seguinte: área de dados da imagem.

d) *Área de dados da imagem contida no arquivo*
Dados que permitem a exibição da imagem propriamente dita, o dados dos pixels a serem exibidos. Podem ser com ou sem compressão.

Obs: Existem dentro da estrutura do BMP alguns campos ditos "Reservados", destinados a uso futuro, que sempre devem ser setados com ZERO. Outros dados são sempre idênticos ou fornecidos mais de uma vez no arquivo, para possível re-checagem e desvio de possíveis erros.

Segue abaixo a tabela identificando a posição no arquivo e o seu significado:

Offset	Campo	Tamanho	Conteúdo
0000h	Identifier	2 bytes	Os caracteres que identificam o Bitmap, com as possíveis entradas : 'BM' - Windows 3.1x, 95, NT, … 'BA' - OS/2 Bitmap Array 'CI' - OS/2 Color Icon

			‘CP’ - OS/2 Color Pointer ‘IC’ - OS/2 Icon ‘PT’ - OS/2 Pointer
0002h	File Size	1 dword	Tamanho Completo do arquivo em Bytes
0006h	Reserved	1 dword	Reservado para uso posterior.
000Ah	Bitmap Data Offset	1 dword	Offset do início do arquivo até o início dos dados do bitmap.
000Eh	Bitmap Header Size	1 dword	Tamanho do cabeçalho usado para descrever as cores do bitmap. Compressão, etc. Os seguinte tamanhos são possíveis : 28h - Windows 3.1x, 95, NT, … 0Ch - OS/2 1.x F0h – OS/2 2.x
0012h	Width	1 dword	Largura horizontal do bitmap em pixels.
0016h	Height	1 dword	Altura vertical do bitmap em pixels.
001Ah	Planes	1 word	Número de planos no bitmap.
001Ch	Bits Per Pixel	1 word	Bits por pixel usados para guardar a informação da palette de cores. Isso também identifica de forma indireta o número de cores possíveis. Possíveis valores são : 1 - Monochrome bitmap 4 - 16 color bitmap 8 - 256 color bitmap 16 - 16bit (high color) bitmap 24 - 24bit (true color) bitmap 32 - 32bit (true color) bitmap
001Eh	Compression	1 dword	Especificação da Compressão. Os seguintes valores são possíveis : 0 - none (também identificado por BI_RGB) 1 - RLE 8-bit / pixel (também identificado por BI_RLE4) 2 - RLE 4-bit / pixel (também identificado por BI_RLE8)

			3 - Bitfields (também identificado por BI_BITFIELDS)
0022h	Bitmap Data Size	1 dword	Tamanho dos dados do Bitmap em bytes. Esse número deve ser arredondado para o próximo limite de 4 bytes.
0026h	HResolution	1 dword	Resolução horizontal expressa em pixel por metro.
002Ah	VResolution	1 dword	Resolução vertical expressa em pixel por metro.
002Eh	Colors	1 dword	Número de cores usado pelo bitmap. Para um bitmap 8-bit / pixel esse valor será 100h ou 256.
0032h	Important Colors	1 dword	Número de cores importantes. Esse será igual ao número de cores usado quando todas as cores forem importantes.
0036h	Palette	N * 4 byte	Especificação da Palette. Para cada entrada da palette 4 bytes são usados para descrever o valor RGB da cor da seguinte forma : 1 byte para o canal azul 1 byte para o canal verde 1 byte para o canal vermelho 1 byte para alpha que é ajustado para 0 (zero)
0436h	Bitmap Data	x bytes	Depende das especificações da compressão. Esse campo contém todos os bytes de dados do bitmap que representam índices para a palette de cores.

Obs: Os seguintes valores foram usados na especificação acima::

Tamanho	Número de bytes
char	1
word	2
dword	4

Referências :

[CAS_79] Castleman, K R : Digital Image Processing, Prentice-Hall, Inc., Englewood Cliffs, N.J.,1979.

[JAI_89] Jain, A.K.: Fundamentals of Digital Image Processing, Prentice-Hall, Inc.,1989.

[GON_87] Gonzalez, R.C.; Wintz, P .: Digital Image Processing, Second Edition, Addison-Wesley Publishing Company, 1987.

[MAS_89] Mascarenhas, N.D.A.; Velasco, F.R.D.: Processamento Digital de Imagens, IV Escola Brasileiro-Argentina de Informática, Universidade Católica de Santiago Del Estero, Termas do Rio Hondo – Argentina, 1989.

[YOU_98] Young, I.T; Gerbrands, J.J; Van, V.L.J; Fundamentals of Image Processing, Delft University of Technology, Netherlands, 1998.

[WAT_98] Alan H. Watt, Fabio Policarpo - The Computer Image , Addison-Wesley Pub Co (Net); 1998

[1] http://www.cs.cf.ac.uk/Dave/Vision_lecture/

[2] http://www.ic.uff.br/~aconci/curso/cont.ps

[3] http://www.ic.uff.br/~aconci/CV.html

[4] http://www.ee.bgu.ac.il/~greg/graphics/special.html

[5] http://www.cogs.susx.ac.uk/users/davidy/teachvision/vision2.html#heading7

[6] http://www.dai.ed.ac.uk/HIPR2/sobel.htm

[7] http://www.dai.ed.ac.uk/HIPR2/roberts.htm

[8] http://www.dai.ed.ac.uk/HIPR2/prewitt.htm

[9] http://www.ufrgs.br/srm/usr/vitor/vitor.html

[10] http://www.iagusp.usp.br/~laerte/atividade1/sip_process2.html

[11] http://www.cprm.gov.br/geo/degeo00.html

[12] http://www.dpi.inpe.br/spring/usuario/c_restau.htm

[13] http://www.epub.org.br/informaticamedica/n0106/imagens.htm

[14] http://rad.usuhs.mil/rad/home

[15] http://www.pediatricdentistry.com/pedo/x-ray.html

[16] www.wotsit.org

[17] www.dcs.ed.ac.uk/home/mxr/gfx/index-hi.html

[18] www.cica.indiana.edu/graphics/image_specs