

**UNIVERSIDADE FEDERAL DE SANTA CATARINA  
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA  
COMPUTAÇÃO**

**Leandro Coser**

**Filtro de Difusão Anisotrópico Orientado por  
Evidência de Borda**

Dissertação submetida à Universidade Federal de Santa Catarina como parte dos requisitos para obtenção do grau de Mestre em Ciência da Computação

Prof. Dr. Jovelino Falqueto

Florianópolis, Fevereiro de 2009.

# **Livros Grátis**

<http://www.livrosgratis.com.br>

Milhares de livros grátis para download.

# **FILTRO DE DIFUSÃO ANISOTRÓPICO ORIENTADO POR EVIDÊNCIA DE BORDA**

Leandro Coser

Esta Dissertação foi julgada adequada para a obtenção do título de Mestre em Ciência da Computação Área de Concentração Sistemas de Computação, e aprovada em sua forma final pelo Programa de Pós-Graduação em Ciência da Computação.

---

Prof. Dr. Frank Augusto Siqueira  
Coordenador do Curso

Banca Examinadora

---

Prof. Dr. Jovelino Falqueto  
Orientador

---

Prof. Dr. rer.nat. Aldo von Wangenheim  
Co-orientador

---

Prof. Dr. rer.nat. Eros Comunello

---

Prof. Dra. Agma Juci Machado Traina

*Agradeço a minha família, especialmente aos meus pais Humberto e Sonia, pelo apoio incondicional e auxílio que me prestaram frente às dificuldades.*

*Ao meu orientador Jovelino e aos professores e Aldo e Eros pela oportunidade e condução na busca do conhecimento.*

*Aos meus amigos do laboratório Lapix, em especial Antonio, Adiel, Eros, Vilson e Sylvio, pela paciência em discutir vários aspectos relacionados ao desenvolvimento deste e outros trabalhos.*

*Em especial à Nozomi pelo auxílio, carinho e compreensão frente aos dias de trabalho árduo que passou em minha companhia.*

*Enfim agradeço a todos aqueles que de maneira direta ou indireta me auxiliaram na conclusão desta dissertação.*

# SUMÁRIO

Capítulo 1 .....	1
Introdução .....	1
1.1 Definição do Problema .....	1
1.2 Justificativa .....	2
1.3 Motivação .....	2
1.4 Hipótese .....	3
1.5 Contribuições .....	3
Capítulo 2 .....	4
Objetivo Geral .....	4
2.1 Objetivos específicos .....	4
Capítulo 3 .....	5
Fundamentação Teórica.....	5
3.1 Imagens Digitais .....	5
3.2 Imagens Coloridas .....	7
3.2.1 Modelo de Cor.....	7
3.2.1.1 Modelo de Cor RGB.....	8
3.3 Técnicas de Processamento Digital de Imagens .....	9
3.4 Filtragem de Imagens.....	11
3.4.1 Filtros para Realce e Suavização.....	12
3.4.1.1 Filtro de Difusão Anisotrópico.....	14
3.5 Segmentação de Imagens .....	17
3.5.1 Segmentação por limiarização .....	18
3.5.2 Segmentação por crescimento de regiões.....	20
3.5.2.1 Segmentação pela funcional de Mumford&Shah.....	22
3.5.3 Segmentação por Detecção de Borda.....	26
3.5.3.1 Detecção de Bordas pelas Máscaras de Sobel.....	27
3.5.3.2 Gradientes pelas Máscaras de Simetria Positiva .....	29
3.5.3.3 Detecção de Bordas pelo Método de Canny .....	29
3.6 Métricas de Qualidade em Imagens .....	32
3.6.1 Erro Máximo .....	32
3.6.2 Erro Médio Absoluto.....	33
3.6.3 Erro Médio Quadrático.....	33

3.6.4	Medidas de Dissimilaridade em Imagens.....	33
3.6.4.1	Índice Rand.....	33
3.6.4.2	Correspondência de Grafos Bipartidos (BGM).....	34
Capítulo 4	.....	36
Desenvolvimento	.....	36
4.1	Filtro de Difusão Anisotrópico com Lambda Adaptativo e Orientação por Mapas de Gradientes .....	36
4.1.1	Aspectos de Implementação.....	36
4.2	Contribuições do Trabalho.....	40
4.2.1	Contribuição de Mapa de Pesos por Gradientes.....	40
4.2.2	Lambda Adaptativo .....	42
4.2.3	Sistema de Equações Final .....	44
Capítulo 5	.....	46
Metodologia	.....	46
5.1	Conjunto de Imagens para Testes .....	46
5.2	Aplicação dos filtros, Versão Tradicional e Modificada .....	49
5.3	Segmentar as Imagens Filtradas .....	49
5.4	Aplicação de Métricas de Qualidade em Imagens com Ground Truth .	50
5.5	Visão Geral do Processo de Teste.....	50
Capítulo 6	.....	52
Resultados	.....	52
6.1	Resultados da Aplicação do Filtro .....	52
6.2	Resultados da Validação e Segmentação.....	56
Capítulo 7	.....	60
Discussões e Conclusões	.....	60
7.1	Trabalhos Futuros .....	61
Referências	.....	62
Anexos	.....	65

## LISTA DE ILUSTRAÇÕES

Figura 1 - Representação de uma imagem em um sinal luminoso em $f(x,y)$ . .....	6
Figura 2 – Luz Visível: faixa de comprimentos de ondas eletromagnéticas em nanômetros ( (WIKIPEDIA, 2008), (GONZALEZ & WOODS, 2002) ). .....	7
Figura 3- Exemplo de composição de cores com máximos valores de RGB. ....	8
Figura 4 - Cubo RGB ( (MATHWORKS, 2007). .....	9
Figura 5 - Passos em processamento digital de imagens (PACIORNIK & MAURICIO, 2004). ..	11
Figura 6 - Imagem de ressonância magnética submetida ao filtro passa-baixa. A) Imagem original, B) após filtragem. ....	13
Figura 7 - Fluxograma de funcionamento do algoritmo de limiarização. ....	19
Figura 8 - Imagem submetida à limiarização com diferentes valores de limiar ,A) imagem original; B),C),D),E),F) diferentes resultados de limiarização. ....	19
Figura 9 - Exemplo de valores de pixel em tons de cinza de uma imagem. ....	20
Figura 10 - Exemplo de valores de pixel em tons de cinza de uma imagem e sua representação em tons de cinza. ....	21
Figura 11 - Resultado de uma segmentação por crescimento de regiões, nos valores em tons de cinza. ....	21
Figura 12- Diferentes resultados de segmentação por crescimento de regiões definidos a partir de diferentes valores de $\lambda$ . ....	25
Figura 13 - Borda de um objeto, em A) imagem original, em B) zoom da imagem para observação de borda. ....	26
Figura 14 - Em A) imagem original, em B) imagem submetida à detecção de bordas do algoritmo de Canny. ....	32
Figura 15 – Pseudocódigo do filtro de difusão anisotrópico. ....	39
Figura 16 – Resultados da aplicação de filtro de difusão tradicional e orientado por mapa de gradientes. (a) imagem original; (b) imagem filtrada com o filtro tradicional; (c) imagem de gradiente; (d) aplicação do filtro com os pesos de (c). ....	42
Figura 17 – Gráfico demonstrando o comportamento do $\lambda$ adaptativo de acordo com a média e desvio padrão. ....	44
Figura 18 – Imagem 124084 de Berkeley, aplicação de ambas as versões do filtro com $\lambda = 20$ e 90 iterações. ....	49
Figura 19 – Imagem original e imagens filtradas após segmentação com Mumford&Shah com $\lambda = 62500$ . ....	49
Figura 20 - Comparação da imagem segmentada com seu respectivo GT e índices RAND e BGM. ....	50
Figura 21 – Visão geral do processo de validação da metodologia. ....	51
Figura 22 – Gráfico de índices RAND para o conjunto de validação. ....	56
Figura 23 - Gráfico de índices BGM para o conjunto de validação. ....	57

## LISTA DE TABELAS

Tabela 1 – Imagens selecionadas randomicamente da base de dados de Berkeley para validação da metodologia proposta. ....	48
Tabela 2 – Resultados da aplicação do filtro anisotrópico, casos de sucesso. ....	54
Tabela 3 – Resultados da aplicação do filtro anisotrópico: casos de falha. ....	55
Tabela 4 – Imagens com resultados menos relevantes de segmentação. ....	58
Tabela 5 – Média dos índices RAND e BGM. ....	59
Tabela 6 – Dados gerados durante a validação. ....	90
Tabela 7 – Imagens originais seguidas da filtragem pelo método tradicional e em seguida pelo método adaptado. ....	97



## **ABREVIACES**

API	Application Programming Interface
CMY	Cyan, Magenta and Yellow
GT	Ground Truth
nm	Nanmetro
PDI	Processamento Digital de Imagens
RGB	Red, Green and Blue
RAND	Rand Index
BGM	Correspondncia de Grafos Bipartidos
ADF	Anisotropic Diffusion Filter - Filtro de difuso anisotrpica
MS	Equao de Energia Funcional de Mumford&Shah

## LISTA DE SÍMBOLOS

$\theta$	Ângulo
$DX$	Derivada no eixo X
$DY$	Derivada no eixo Y
$G$	Gradiente aproximado de uma imagem
$G_x$	Gradiente em X de uma imagem
$G_y$	Gradiente em Y de uma imagem
$I$	Imagem (Matriz de intensidades luminosas ou cores)
$M$	Máscara de convolução
$\in$	Indica a pertinência de elementos a conjuntos
$\mathbb{R}$	Conjunto dos Números Reais

## **RESUMO**

Filtro de Difusão Anisotrópico é uma técnica bem estabelecida para melhoria de imagens que pode ser empregada para suavização de imagens ainda preservando alguma informação de borda. Entretanto, quando é necessário aplicar muitas iterações do filtro, gradualmente as bordas desaparecerão e serão esmaecidas pelo processo. Este trabalho propôs a adoção de mapa de gradientes coloridos para guiar o processo de suavização que claramente define quais bordas serão preservadas depois de muitas iterações. Como característica adicional, o método proposto emprega informações estatísticas para variar o parâmetro de lambda dinamicamente durante a filtragem, permitindo uma suavização adaptativa mais sensível. Os resultados alcançados demonstram-se superiores quando comparados com o filtro tradicional.

Palavras-Chave: Filtro de Difusão Anisotrópica, Lambda Adaptativo, Mapa de Gradientes Coloridos.

## **ABSTRACT**

Anisotropic diffusion filter is a well-established technique for image enhancement that can be employed to smooth images while preserving some degree of edge information. However, when it becomes necessary to apply a large number of filtering iterations, edges gradually fade away and are ultimately smoothed by the process. We propose the adoption of a color gradient map to guide the smoothing so that clearly-defined edges are preserved even after many iterations. As an additional feature, our method employs the statistical information to vary the lambda parameter dynamically during filtering, allowing for a more sensitive adaptive smoothing. The results achieved show good results when compared with traditional filter.

Key-words: Anisotropic Diffusion Filter, Adaptive Lambda, Color Gradient Map

# ***Capítulo 1***

## ***Introdução***

Uma imagem digital é composta por sinais de luminosidade e coordenadas fiéis à cena original capturada. De certa forma, sua maior funcionalidade é manter a representação do mundo real. Porém, quando algum processo de visão computacional deve ser aplicado sobre a imagem, por exemplo, a detecção de determinados objetos, estas atividades podem ser complexas e com respostas não satisfatórias em determinados casos. São necessários métodos e rotinas para modificar a imagem original de maneira que se alcance um resultado desejável. Este tratamento é conhecido como processamento digital de imagens.

A área de processamento digital de imagens tem evoluído muito nos últimos anos, principalmente porque a capacidade de processamento atual é muito maior que em anos anteriores. Logo, sistemas de visão computacional têm ganhado cada vez mais espaço em atividade cada vez mais complexas, permitindo uma grande quantidade de aplicações serem desenvolvidas.

Os filtros são técnicas de grande visibilidade dentro do processamento de imagens atuando durante a etapa de pré-processamento de imagens, para que estas possam oferecer melhores resultados para etapas subseqüentes. Este trabalho apresenta as atividades realizadas para que se houvessem melhorias de uma técnica de filtragem de imagens, conhecida como “filtro de difusão anisotrópico”, muito utilizada na etapa de pré-processamento de imagens digitais.

### ***1.1 Definição do Problema***

O filtro de difusão anisotrópico é uma técnica de esmaecimento de imagens. Trata-se de um filtro muito superior às técnicas de filtragem de imagens comumente utilizadas como filtros de médias e filtros gaussianos. Infelizmente ainda podem ocorrer falhar em dois aspectos: Em casos onde são necessárias muitas iterações do algoritmo, acontece o esmaecimento sob regiões de bordas, no processo de difusão. Conseqüentemente ao longo do tempo essas bordas acabam sendo difundidas e não

serão mais evidentes ou claras. Um segundo aspecto é quando o usuário pretende usar o filtro de difusão anisotrópico (dada suas características) como pré-segmentação. Por causa das características de construção ele pode esmaecer regiões nitidamente desnecessárias. Assim o que se pretende é ter forte esmaecimento em regiões não fronteiriças ou de baixa variação e baixo esmaecimento em regiões de borda.

## ***1.2 Justificativa***

O processamento digital de imagens é um sub-conjunto de tarefas ligadas à visão computacional. É sabido que sistemas de visão computacional ainda não possuem um conjunto de técnicas definitivas para solução global de problemas gerais. Por outro lado sabe-se que a abordagem de solução de pequenas etapas de um problema muito complexo com o uso de processos específicos, pode levar a resultados muito significativos na solução de alguma necessidade de visão computacional.

Neste contexto, durante o pipeline de processamento digital de imagens, é necessário, na maioria das vezes, realizarmos alguma tarefa de filtragem para melhorar a resposta das etapas subseqüentes. Neste ponto é que estudos sobre o filtro de difusão anisotrópico são de grande importância, porque este é um filtro gaussiano muito poderoso por preservar as bordas entre regiões. Estima-se que melhoria na qualidade dos resultados deste filtro no pré-processamento de imagens pode auxiliar muitos sistemas de visão computacional.

Este trabalho tem o objetivo principal de realizar a etapa de filtragem de imagens com filtro de difusão anisotrópico tendo em vista a necessidade de melhoria da informação visual para a interpretação humana e segmentação de cenas para percepção automática através de máquinas.

## ***1.3 Motivação***

De maneira geral a filtragem de imagens pode explicitar a melhoria do aspecto visual de certas feições estruturais das imagens para o analista humano ou mesmo para o computador, e fornecer outros subsídios para a sua interpretação. Então toda tentativa

bem sucedida de melhorar as técnicas atuais de processamento de imagem tem impacto direto na usabilidade das ferramentas e sistemas de processamento digital de imagens.

### ***1.4 Hipótese***

Espera-se que com o uso de mapas de gradientes e adaptação de parâmetros internos ao algoritmo de filtragem conhecido como filtro de difusão anisotrópico fossem superem dificuldades de esmaecimento de regiões de bordas e melhorem os resultados do filtro de difusão anisotrópico na pré-segmentação de imagens.

### ***1.5 Contribuições***

O emprego do filtro de difusão anisotrópico mantido pelo mapa de gradientes e o uso de adaptação dos parâmetros de lambda – parâmetro que determina a velocidade de difusão na região – embutidos em etapas de segmentação mostraram ser melhorias importantes no processo de visão computacional.

## ***Capítulo 2***

### ***Objetivo Geral***

O objeto de estudo do presente trabalho envolve a modificação da técnica conhecida como filtro de difusão anisotrópica com a adição de características que permitam que este filtro realize suavização de imagens, sem entretanto interferir sobre regiões que delimitam objetos distintos dentro destas imagens. Tornando a etapa de pré-segmentação mais confiável.

#### ***2.1 Objetivos específicos***

1. Definir heurísticas para o melhor funcionamento do filtro de difusão anisotrópico.
2. Aplicação do filtro de difusão anisotrópico como filtro de suavização na técnica de pré-segmentação.
3. Observar a precisão de métodos de segmentação após filtragem com as modificações inseridas.



## ***Capítulo 3***

### ***Fundamentação Teórica***

Neste capítulo são abordados vários conceitos inerentes à área de processamento digital de imagem e técnicas necessárias para a validação da metodologia de filtragem proposta no presente trabalho.

#### ***3.1 Imagens Digitais***

Imagens são importantes para os seres humanos como fonte de informações, isto porque dentre os cinco sentidos de que os humanos são dotados, a visão, juntamente com os sistemas que a compõem, é considerada a mais evoluída (ou mais complexa), possibilitando a compreensão quase que instantânea do ambiente relacionado. O sistema visual tem a capacidade de análise de imagens e objetos para extração de informações tais como formato, cor, tamanho, profundidade, dentre outras características, as quais são necessárias para racionalizar ou compreender o objeto dentro do ambiente e assim tomar as decisões sobre ações necessárias.

As imagens podem ser vistas como o principal mecanismo de comunicação não-formal existente, permitindo que se tenha um entendimento imediato de determinadas situações, sem a necessidade de interpretações muito precisas ou formalização de regras. Por esta razão a imagem tem importante papel na aquisição de informações para análise e tomada de decisão em diversas áreas do conhecimento humano.

Para (GONZALEZ & WOODS, 2002), uma imagem é definida como uma função  $f(x,y)$  onde  $x$  e  $y$  são coordenadas em um plano instanciando um ponto luminoso, como pode ser observado na Figura 1. Esta é a definição matemática mais plausível de uma imagem do mundo real, ou seja, os pontos de luminosidade analógicos são contínuos, se  $(x,y) \in \mathbb{R}$ , conjunto dos reais.

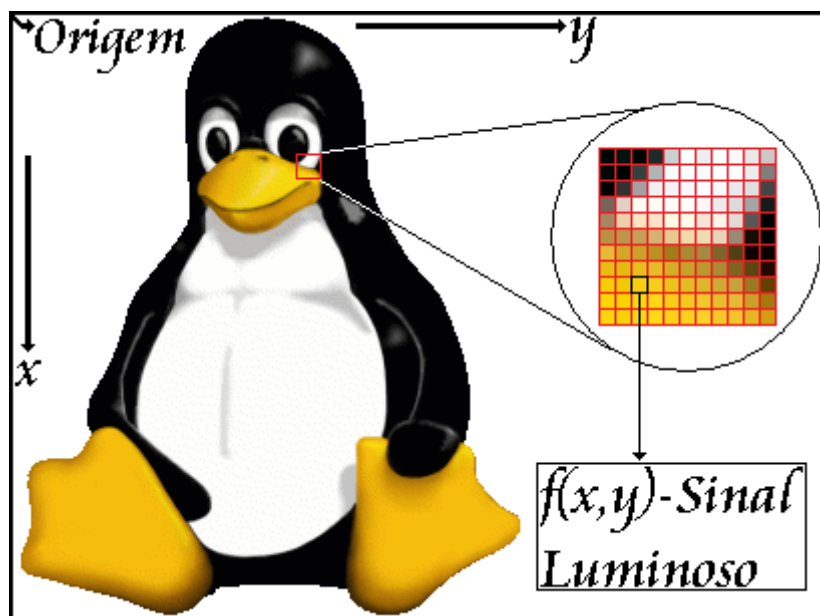


Figura 1 - Representação de uma imagem em um sinal luminoso em  $f(x,y)$ .

Toda imagem tratada computacionalmente e numericamente é denominada imagem digital. Deste modo, as imagens digitais são imagens com uma função bidimensional  $f(x,y)$  os quais são atribuídos valores numéricos digitais, pois como computadores não manipulam valores analógicos, é necessário realizar a conversão da forma contínua para a forma digital. Isto é feito através da divisão da área da imagem em uma matriz bidimensional de tamanho  $(x, y)$  (PARKER, 1997). Para que a captação destas imagens digitais em matrizes bidimensionais seja possível são utilizadas diversas técnicas que permitem a captação das imagens a partir do mundo exterior. O conjunto destas técnicas é denominado de imageamento.

Na representação por matrizes bidimensionais, cada elemento do plano cartesiano é conhecido como pixel (*Picture element* ou elementos da figura), e a cada pixel ou elemento da matriz é atribuído um valor identificando uma intensidade/tonalidade de cor. Desta forma o acesso à informação é direto em qualquer ponto da imagem.

O pixel é também um elemento de dimensões finitas na representação de uma imagem digital. A quantidade de pixels em uma imagem reflete a sua resolução. O número de níveis ou cores é chamado quantificação. Estes dois atributos definem a qualidade da imagem.

## 3.2 Imagens Coloridas

Essencialmente as cores que humanos e outros animais vêem nos objetos são determinadas pela natureza da luz refletida pelo objeto. Como ilustrado na Figura 2 – Luz Visível: faixa de comprimentos de ondas eletromagnéticas em nanômetros ( , ), a luz visível é composta por uma pequena banda de frequências no espectro eletromagnético (GONZALEZ & WOODS, 2002).

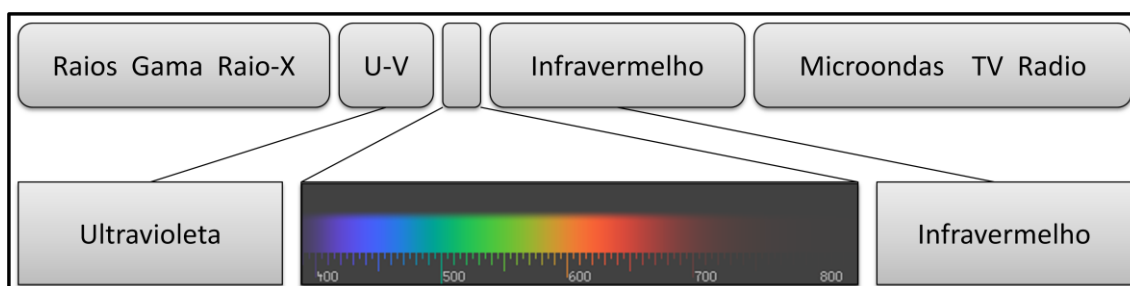


Figura 2 – Luz Visível: faixa de comprimentos de ondas eletromagnéticas em nanômetros ( (WIKIPEDIA, 2008), (GONZALEZ & WOODS, 2002) ).

É de fundamental importância entender o funcionamento do sistema visual humano, porque basicamente a percepção é limitada à visualização de imagens do espectro visível de cores, ou seja, o olho humano vê um número limitado de cores (BOTELHO, 2005). Isto significa que não é possível visualizar muitas cores que são geradas e estão presentes no mundo real. Também para o sistema ocular humano não é possível captar imagens que são geradas por outro meio físico que não a luz.

Existem outras formas possíveis de captar imagens além da utilização da luz emitida pelos objetos, podendo estas ser geradas por outros processos físicos como som, pressão, absorção e radiação. Estas formas exigem uma conversão prévia, de maneira que a sua visualização ou digitalização para análise seja possível.

### 3.2.1 Modelo de Cor

A imagem digital é armazenada através de pixels, onde cada região ou coordenada corresponde a uma única cor, mas a especificação de que informações são as cores propriamente ditas pode variar de acordo como o sistema de aquisição ou propósito de dada imagem digital. Todos os formatos de arquivos de imagens digitais podem

armazenar o pixel como uma estrutura de dados contendo informações diversas, as quais descrevem uma cor e a posição do pixel.

Por esta razão foram especificados os modelos de cores, criados para facilitar a especificação de cores em muitos padrões. Assim indiferente do sistema de aquisição e/ou finalidade da imagem a cor é armazenada seguindo um determinado padrão, facilitando o intercâmbio e uso destas informações em diversos sistemas de software e hardware.

Os modelos de cor mais conhecidos são o RGB, o CMY, CMYK e o HSI (GONZALEZ & WOODS, 2002). Todos estes modelos têm finalidades um pouco diferentes, mas seguem os padrões de modelo de cor, permitindo facilmente a conversão de um pixel colorido em um modelo para outro modelo. O modelo RGB é um padrão de indústria e será usado aqui. Uma descrição mais detalhada é fornecida na referência.

### 3.2.1.1 Modelo de Cor RGB

O modelo de cor RGB é um padrão de indústria para aquisição, armazenamento e visualização de imagens coloridas. É formado por três cores básicas, o vermelho, o verde, o azul e sua combinação gera as demais cores. RGB é abreviação do nome das três cores em inglês: Red ou vermelho, Green ou verde e Blue ou azul.

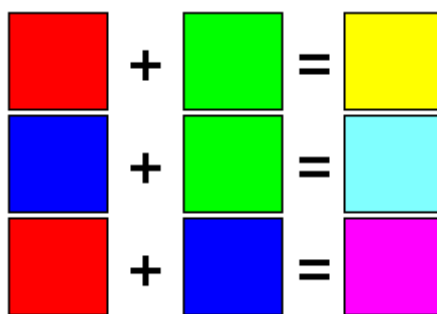


Figura 3- Exemplo de composição de cores com máximos valores de RGB.

Na Figura 3 nota-se que para compor a cor amarela é necessário a cor vermelha e a verde. A intensidade máxima de vermelho e a intensidade máxima de verde fornecem o amarelo, também em intensidade máxima. As tonalidades e cores são formadas com a variação de intensidades de valores para RGB, assim o valor mínimo para um dado canal de cor é 0 e o valor máximo é 255. Conseqüentemente são  $2^8$  possibilidades de cor

por canal, isto é, 256 variações de cada uma das componentes de cor, quando se utiliza 8 bits para indicação de cor.

O modelo RGB foi arquitetado tendo em mente a fisiologia do funcionamento dos olhos humanos, ou seja, a maneira como os cones (células fotorreceptoras) recebem e processam as cores (HUNT, 2004). O objetivo foi criar um modelo de cor para ser usado em hardware que possuísse a melhor resposta de acordo com o funcionamento do olho humano.

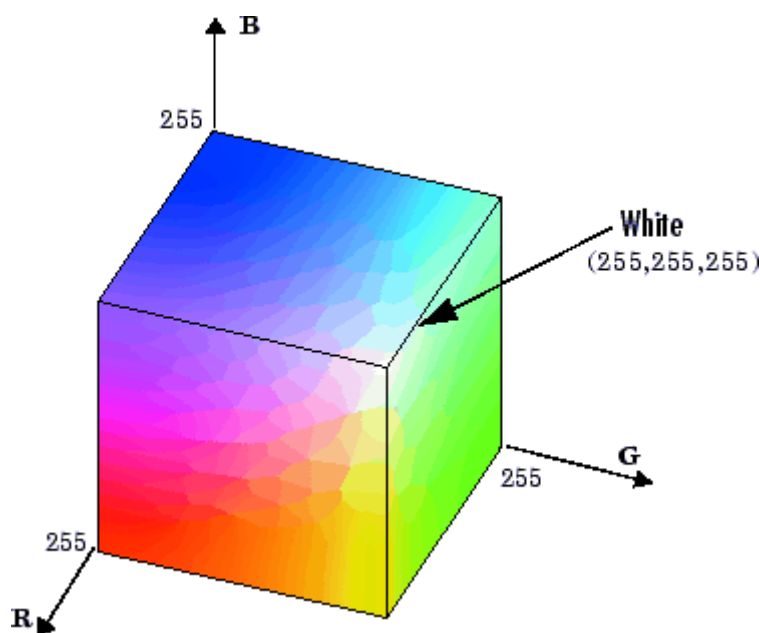


Figura 4 - Cubo RGB ( (MATHWORKS, 2007).

O modelo RGB pode ser visualizado como um cubo, ou seja, a maneira de representação da disposição das três cores é baseada no sistema de coordenadas cartesiano, na qual cada aresta do cubo corresponde a uma das cores do modelo RGB, conforme pode ser visualizado na Figura 4.

### ***3.3 Técnicas de Processamento Digital de Imagens***

Um longo caminho foi percorrido para se alcançar o que é possível realizar hoje em termos de processamento de imagens. Historicamente falando, uma das primeiras aplicações que utilizou técnicas de processamento digital de imagens foi a melhoria de ilustrações de jornais enviados por cabo submarino entre Londres e New York por volta de 1920 (GONZALEZ & WOODS, 2002).

Mas as técnicas de processamento digital de imagens evoluíram em meados dos anos 60 com o advento de computadores digitais e com o programa espacial norte-americano, onde, em 1964, as imagens da lua transmitidas pela sonda Ranger 7, foram processadas por um computador para corrigir vários tipos de distorções inerentes à câmara de televisão à bordo. As técnicas de processamento usadas nesta época, serviram de base para o realce e restauração de imagens de outros programas espaciais posteriores, como as expedições tripuladas da série Apollo para a lua, por exemplo.

O processamento digital de imagens é um conjunto de métodos e rotinas computacionais que realizam tarefas distintas sobre imagens, atividades estas que podem ser desde simples como filtro passa-baixa, até poderosas funções de segmentação.

Para realização do processamento normalmente não existem métodos completamente genéricos que sejam capazes de realizar uma dada tarefa em qualquer tipo de imagem, ou domínio de problema. Isso significa que de fato não existem métodos realmente definitivos para uma dada tarefa. Em geral existem inúmeras rotinas e métodos, cada uma com uma função e qualificação específica idealizada para problemas muito específicos. Estas atividades são selecionadas de acordo com a “aptidão” da função, em relação à necessidade; realizam atividade complexas como um todo, sob a forma de um processo.

Existem outras tarefas com imagens digitais tais como reconhecimento de padrões, onde a entrada do processo é uma imagem e a saída constitui-se em uma classificação, ou geralmente, a descrição da mesma. A área de gráficos por computador, que envolve a geração de imagens digitais a partir de descrições destas imagens (PACIORNIK & MAURICIO, 2004) é outro exemplo de processamento de imagens digitais.

Desde toda a evolução explicitada até os dias atuais, tem-se uma gama bastante variada de métodos e funções que realizam determinadas tarefas dentro do contexto do processamento digital de imagens. Dentre as diversas técnicas, destaca-se a correção de erros, realce, segmentação, filtros diversos, entre outras. Em algumas situações, muitos

passos de processamento digital de imagens podem ser requeridos para correção de problemas em imagens, tais como: ruído eletrônico, iluminação irregular, etc.

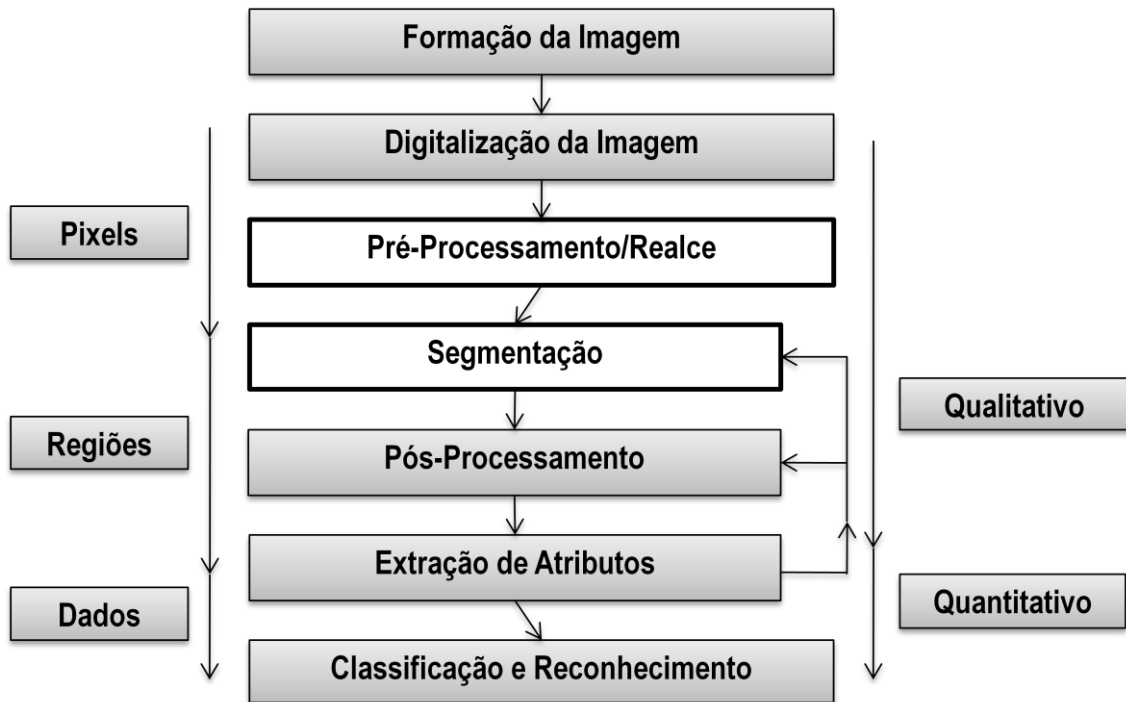


Figura 5 - Passos em processamento digital de imagens (PACIORNIK & MAURICIO, 2004).

A Figura 5 mostra os passos de processamento de imagens e os resultados de processamento de cada etapa desde a criação da imagem até a obtenção de dados qualitativos.

### ***3.4 Filtragem de Imagens***

Imagens digitais frequentemente sofrem degradações durante o processo de aquisição, transformação e processamento. Essa degradação é normalmente chamada de ruído (PEDRINI & SCHWARTZ, 2008). A filtragem de imagens normalmente está associada ao passo de pré-processamento de imagens, mas também pode ocorrer em outras etapas de um sistema de processamento digital de imagens. Filtragens consistem na tentativa de minimizar ruído em imagens, mas também podem ser usadas tanto para suavizar como para realçar feições ou mesmo detectar bordas.

Em geral, filtros são aplicados na forma de operações de convolução. Dada uma imagem  $I$ , a convolução é um cálculo aplicado a uma matriz, que consiste em calcular o

valor do pixel da imagem de saída fazendo uma soma ponderada, conforme equação (1) para  $p_i = f(x,y)$ ,  $1 \leq i \leq 9$ , combinando um pixel  $I(x,y)$  e seus vizinhos.

$$d(x,y) = m_1 * p_1 + m_2 * p_2 + \dots + m_9 * p_9 = \sum_{i=1}^9 m_i * p_i \quad (1)$$

A soma é efetuada utilizando pesos contidos em uma matriz quadrada, conforme a equação (2), de ordem  $n+1$  denominada máscara de convolução  $M$ .

$$M = \begin{bmatrix} M_{1,1} & M_{1,2} & M_{1,3} \\ M_{2,1} & M_{2,2} & M_{2,3} \\ M_{3,1} & M_{3,2} & M_{3,3} \end{bmatrix} \quad (2)$$

(PEDRINI & SCHWARTZ, 2008).

A equação (3) demonstra o modelo geral para operação de convolução (SCHALKOFF, 1989), aplicado ao pixel  $d(x,y)$ . Observa-se a soma nos sentidos cartesianos  $i,j$  aplicando-se a matriz de convolução  $M(i,j)$  sobre a vizinhança de  $I(x,y)$ .

$$d(x,y) = \left[ \sum_{i=0}^{\frac{n}{2}} \sum_{j=0}^{\frac{n}{2}} M[i,j] * I(x-i, y-j) \right] \quad (3)$$

(SCHALKOFF, 1989)

O resultado de filtros dependerá então da máscara de convolução utilizada. Independente do filtro utilizado, o processo de filtragem consiste em somar os produtos entre os coeficientes da matriz de observação e os valores de intensidade dos pixels na janela em uma posição específica da imagem.

### 3.4.1 Filtros para Realce e Suavização

Filtros para suavização de imagens são amplamente usados em processos de visão computacional. É comum utilizar técnicas para transformar, suavizar ou realçar efeitos, detectar bordas entre outras atividades. Existem diversos tipos de filtros, os mais conhecidos são os filtros passa-baixa, passa-alta, mediana, estatístico e gaussianos.

O filtro conhecido como passa-baixa tem o resultado de suavizar (PEDRINI & SCHWARTZ, 2008) a imagem atenuando ou eliminando os componentes de altas-



freqüências, isto é, o filtro deixa passar as baixas-freqüências, ou seja, as baixas-freqüências permanecem inalteradas. O mesmo tende a minimizar ruídos e apresenta o efeito de borramento/esmaecimento da imagem (RABELO, 2002), como pode ser visualizado na Figura 6, devido à atenuação de altas freqüências que correspondem às transições abruptas.

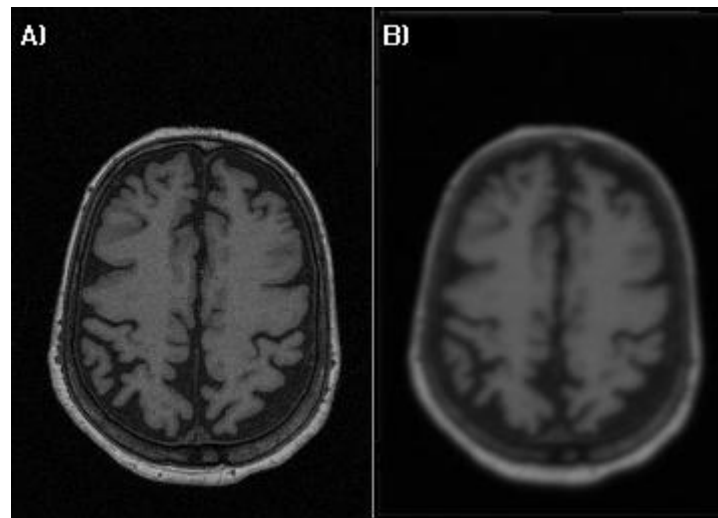


Figura 6 - Imagem de ressonância magnética submetida ao filtro passa-baixa. A) Imagem original, B) após filtragem.

Os filtros passa-alta realçam detalhes, atenuam ou eliminam as componentes de baixa-freqüência. Esses componentes são responsáveis por características que variam lentamente, tais como contraste total e intensidade média, o que produz um realce das bordas da imagem (RABELO, 2002). Ou seja, as transições entre regiões diferentes tornam-se mais nítidas, entretanto acabam por ampliar o sinal de ruído que possa existir na imagem (PEDRINI & SCHWARTZ, 2008).

Filtros gaussianos possuem coeficientes para máscara gerados a partir de uma função gaussiana. A equação (4) define uma função gaussiana discreta com média zero e desvio padrão  $\sigma$  (PEDRINI & SCHWARTZ, 2008).

$$G(x, y) = \frac{1}{2\pi^2} e^{\left(\frac{-(x^2+y^2)}{2\sigma^2}\right)} \quad (4)$$

(PEDRINI & SCHWARTZ, 2008)

Filtros gaussianos são filtros utilizados para suavização de imagens, e (PEDRINI & SCHWARTZ, 2008) define uma série de características que tornam o uso de filtros gaussianos interessantes:

- A resposta do filtro é isotrópica, ou seja, igual em todas as direções.
- A resposta do filtro é uma média ponderada dos vizinhos do pixel.
- A largura do filtro, ou o grau de suavização está relacionado ao parâmetro  $\sigma$ .
- Funções gaussianas são separáveis, o que significa que a convolução pode ser aplicada em uma direção por uma máscara e outra máscara pode computar o resultado em outra direção, ao mesmo tempo.

Filtros gaussianos são importantes também por formarem a base da filtragem também em processos de difusão anisotrópica, discutidos a seguir.

### 3.4.1.1 Filtro de Difusão Anisotrópico

As bases sobre a teoria envolvida no desenvolvimento de filtro de difusão não linear anisotrópico estão intimamente envolvidas com o conceito matemático de equações que descrevem fenômenos físicos, mais especificamente termos de sistemas de equações diferenciais parciais (PDE - *Partial Differential Equations*). O avanço tecnológico atual permite a utilização em áreas antes proibitivas, como o processamento digital de imagens, e conseqüentemente vários sistemas são desenvolvidos e fundamentados nas bases teóricas de PDEs para processamento de imagens.

Os principais estudos nesta área são para o uso em técnicas de smoothing<sup>1</sup>, restauração de imagens e detecção de bordas (B.M. ter Haar Romeny (Ed), 1994). Filtro de difusão encontra-se em uma classe de sistemas PDE conhecidos como parabólicas, e correspondem ao axioma natural do processo de difusão, visando a representação de fenômenos físicos de propagação e transporte de massa, como por exemplo a dinâmica dos gases e fluídos (WEICKERT, 1998).

O problema é que estes filtros operam no espaço escalar utilizado para a análise de imagens e sinais em multi-escala. Técnicas de espaço escalares trabalham de modo linear resultando em imagens com resolução grosseiras fazendo uma convolução gaussiana da imagem inicial (WEICKERT, 1998).

---

<sup>1</sup> Smoothing, trata-se da operação de suavização ou esmaecimento.

Para superar esta dificuldade é que modificações destes sistemas PDE são sugeridas por Perona e Malik em 1987 (WEICKERT, 2001). Estas apresentam o conceito de filtro de difusão não-linear, das quais tornam uma suavização, por exemplo, em filtro adaptativo dependente de propriedades locais da imagem. Perona e Malik aplicaram um processo não-linear (por definição anisotropia) que reduz o coeficiente de difusão até que esses locais estejam “encapsulados” por uma borda (WEICKERT, 1998). O filtro é dado pela equação (5) e assegura que as bordas “grossas” fiquem menos borradas pelo filtro de difusão.

$$g(s^2) = \frac{1}{1 + s^2/\lambda^2} \quad (5) \text{ (WEICKERT, 2001)}$$

para  $\lambda > 0$  (lambda)

Eles discretizaram (espaço-temporalmente) a sua equação de difusão anisotrópica como na equação (6), onde  $I(s,t)$  é a imagem discretizada espacial e temporalmente,  $s$  denota a posição do pixel numa grade discreta bi-dimensional;  $t$  é o passo de tempo discreto ( $t \geq 0$ );  $\lambda$  é a constante que determina a velocidade de difusão;  $|\eta_s|$  representa o conjunto de vizinhos espaciais de um pixel  $s$  (normalmente vizinhos de 4),  $\nabla I_{s,p}(t)$  indica a magnitude do gradiente da imagem  $I$  no ponto  $s$  na direção de  $(s,p)$  na iteração  $t$  onde  $\nabla I_{s,p}(t) = I(p,t) - I(s,t)$ , para todo ponto em  $\eta_s$  (WEICKERT, 2001).

$$I(s, t + 1) = I(s, t) + \frac{\lambda}{|\eta_s|} \sum_{p \in \eta_s} g(|\nabla I_{s,p}(t)|) \nabla I_{s,p}(t) \quad (6)$$

Na formulação da difusão anisotrópica de Perona-Malik, existe uma função chamada parada-na-aresta (*edge stopping function*)  $g$ , que controla a intensidade da difusão de acordo com o gradiente do ponto que deve sofrer difusão. A função parada-na-aresta possui um parâmetro de escala  $\lambda$  que, em conjunto com o gradiente, indica se a difusão deve ser forte ou fraca. Duas funções são sugeridas (WEICKERT, 1998) conforme equações (7) e (8) (WEICKERT, 1998)

$$g(x) = \frac{1}{1 + \frac{x^2}{2\sigma^2}} \quad (7) \text{ (WEICKERT, 1998)}$$

$$g(x) = e^{-\left[\frac{x^2}{2\sigma^2}\right]} \quad (8) \text{ (WEICKERT, 1998)}$$

Segundo Weickert (WEICKERT, 1998) esta técnica pode ser estendida para um verdadeiro processo de anisotropia com a utilização de tensores de difusão adaptativos. A equação (9) (BROX, 2006) descreve o processo de difusão anisotrópica, segundo Weickert (WEICKERT, 1998). É uma operação de convolução aplicada com uma máscara adaptativa  $D$ , onde basicamente para uma imagem  $I(x,y)$  com  $M$  canais, o sinal é inicializado com  $u(x,y,0) = I(x,y)$ , tendo  $i = 1 \dots M$  para cada canal independente (BROX, 2006).

$$\partial_t u_i = \text{div} \left( D \left( \sum_{k=1}^M \nabla u_k \nabla u_k^T \right) \nabla u_i \right) \quad (9) \text{ (BROX, 2006)}$$

Sendo assim, quando  $D$  é uma matriz identidade, temos uma difusão homogênea. Sendo  $D$  uma matriz identidade com um fator de escala, temos difusão isotrópica, e finalmente sendo  $D$  uma matriz arbitrária simétrica positiva, definida para cada local da imagem, temos uma difusão anisotrópica (BROX, 2006).

A formulação de difusão anisotrópica proposta por Weickert é semelhante aos modelos difusos de Perona e Malik, onde uma solução discretizada é encontrada por um procedimento iterativo. A forma discreta de um verdadeiro filtro anisotrópico é dada pelo modelo discreto (WEICKERT, 1998) pela equação (10) (WEICKERT, 2001):

$$u_{i,j}^{k+1} = u_{i,j}^k + \tau \sum_{(m,n) \in N(i,j)} g_{m,n}^k \frac{u_{m,n}^k - u_{i,j}^k}{|(m,n) - (i,j)|^2} \quad (10) \text{ (WEICKERT, 2001)}$$

Na equação decomposta acima o índice superior  $k$  denota a solução no tempo  $k\tau$  com  $\tau$  posicionando o tempo de parada; o conjunto  $N(i,j)$  contém a vizinhança do pixel  $(i,j)$ ; e a expressão  $|(m,n) - (i,j)|$  indica a distância linear entre os pixels  $(i,j)$  e  $(m,n)$ . O termo  $g_{m,n}^k$  aproxima o coeficiente de difusão ao local  $((i+m)/2, (j+n)/2)$  no tempo  $k\tau$ . Isto representa a taxa de difusão pertencente à conexão entre o gradiente  $(i,j)$  e  $(m,n)$ , onde a magnitude do gradiente  $g$  pode ser estimado de um dado discreto usando uma máscara.

Weickert considera o modelo de Perona e Malik como isotrópico, desde que utilizado um valor escalar difuso e não um tensor de difusão. O experimento de Perona e Malik é visualmente imprevisível, as bordas ficam estáveis somente após um alto número de iterações. Entretanto a detecção de bordas utilizando este processo ultrapassa os resultados de um detector de bordas por Canny linear (para mais detalhes veja página 29, seção 3.5.3.3), mesmo sem a utilização de máximos coeficientes, isto porque a difusão e a detecção de bordas interativa estão embutidas como um único processo simples em vez de processos independentes. O interior de um segmento sempre é mantido, enquanto que as bordas da difusão são inibidas.

### ***3.5 Segmentação de Imagens***

A segmentação de imagens é um procedimento aplicável a muitos casos no processamento digital de imagens, isto porque a segmentação executa uma tarefa muito importante que é requerida na maioria dos processos de análise de imagens. Com a segmentação de imagens, é possível tratar cada entidade individualmente, tornando-as regiões com propriedades significativas, permitindo e facilitando a classificação, observação e até mesmo a comparação.

Segundo (GONZALEZ & WOODS, 2000), a segmentação é a etapa do processamento digital de imagens que consiste na subdivisão da imagem de entrada em regiões constituintes ou objetos. Um segmento então pode ser definido como um conjunto de pixels, com características comuns e de vizinhança agrupada, formados a partir de regiões homogêneas na imagem, e cercado por regiões não homogêneas. O processo de segmentação é muito complexo porque tenta traduzir para o computador um processo cognitivo, extremamente sofisticado que é realizado através de um mecanismo de visão. O sistema ocular provê, de maneira natural nos seres vivos, a seleção e diferenciação de objetos no ambiente, permitindo a interação com o meio. Já no meio computacional, o nível de detecção de objetos ou segmentação dependerá do problema que está sendo solucionado, ou seja, em determinados sistemas não é necessária uma grande quantidade de objetos para a análise, apenas as regiões de maior interesse.

Existem diversos métodos para realizar a segmentação de imagens; os principais métodos são baseados em segmentação por limiarização, segmentação por bordas e

segmentação por crescimento de regiões. Porém os métodos de segmentação podem ser complementares e um apresentará resultados diferenciados do outro, e dependendo da aplicação onde serão utilizados, vários destes podem ser combinados melhorando a qualidade final das imagens resultantes. A essas combinações dá-se o nome de técnicas híbridas (COMUNELLO, 1999).

Algoritmos de segmentação praticamente tomam por base duas propriedades do pixel: descontinuidade e similaridade. A descontinuidade é nada mais do que a própria borda na maioria dos casos, pois mudanças de tonalidade de maneira brusca sugerem o isolamento destes objetos. A similaridade busca o agrupamento de regiões que possuam, como a própria denominação sugere, características semelhantes às regiões vizinhas. Este método praticamente se resume a uma entrada ou limiar. Dependendo do critério utilizado para a junção destas áreas similares, a quantidade de segmentos será muito grande, prejudicando o isolamento de determinados objetos como um todo. Para isso, quando se utilizam métodos de segmentação de imagens, espera-se que os objetos possuam superfícies similares, principalmente no caráter da homogeneidade, onde as regiões correspondam a áreas de contraste ou pequenas variações de intensidade.

### 3.5.1 Segmentação por limiarização

A técnica de segmentação por limiarização trabalha com um limiar conhecido como threshold, o qual define o nível de seleção dos pontos da imagem a incorporar-se em uma dada classe, por meio de valores máximos e mínimos.

É uma técnica facilmente construída em software e hardware; para isso é necessário determinar os níveis de threshold ou limiares, o que pode ser feito pelo uso de histograma ou simplesmente por valor arbitrário. Logo após é realizada a verificação dos pontos, conforme fluxograma da Figura 7, onde os pixels são verificados e comparados com o limiar corrente. O limiar comporta-se como uma linha divisória, assim impõe uma única regra bem definida, ou seja, os pontos sempre farão parte de um lado ou outro da linha. Isso separa a imagem em duas ou mais classes distintas, dependendo estas do valor de limiares existentes. A Figura 8 demonstra uma imagem com segmentação por limiarização, onde foi considerado somente um limiar.

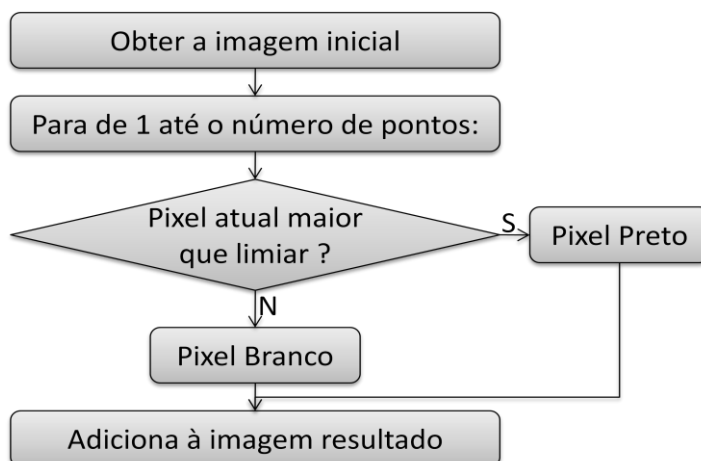


Figura 7 - Fluxograma de funcionamento do algoritmo de limiarização.

A segmentação por limiarização sempre apresenta resultados com alta fragmentação, mas também bons resultados são apresentados quando as imagens são homogêneas. O resultado deste tipo de processamento são imagens com duas cores e algumas vezes um pouco mais de tons de cinza. Este não é um fator limitante, pois isso dependerá da imagem a ser tratada e sua finalidade na solução de determinado problema.

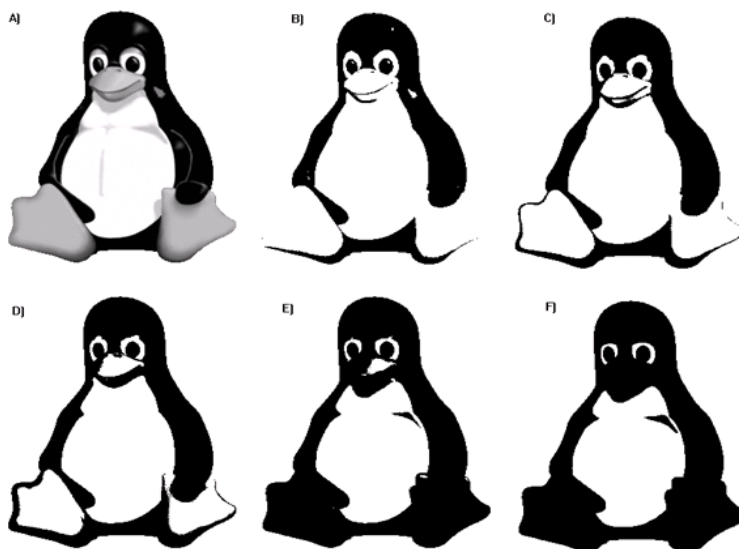


Figura 8 - Imagem submetida à limiarização com diferentes valores de limiar ,A) imagem original; B),C),D),E),F) diferentes resultados de limiarização.

Este método de segmentação é comumente utilizado para localizar falhas em imagens contínuas, como em circuitos eletrônicos ou indústrias de peças, onde os padrões da imagem estão muito bem definidos.

### 3.5.2 Segmentação por crescimento de regiões

A técnica de segmentação baseada em crescimento por regiões procura obter regiões com a mesma característica durante o processo de análise de imagens. Para que isso ocorra o usual nos algoritmos de segmentação por crescimento por regiões é a exploração simples, baseada no tratamento linha por linha da imagem de entrada. Nela são definidos segmentos, os quais são determinados pela aplicação de detecção do ponto inicial e assim para cada linha subsequente. Deste modo para cada segmento já existente são adicionados aqueles pixels adjacentes, os quais compartilham propriedades em comum (COMUNELLO, 1999). Da mesma forma, um novo segmento é iniciado com cada pixel que não é anexado ao segmento anterior.

85	61	61	85	85	61	85	85	61	85	85	0
61	85	61	85	61	85	85	61	85	85	85	0
85	85	85	85	85	85	85	85	61	85	85	0
85	85	73	85	85	85	61	85	85	73	73	0
158	158	158	158	158	81	69	57	57	57	57	0
158	146	158	158	146	57	57	81	57	81	57	0
158	158	146	158	158	57	69	81	57	81	57	0
158	146	146	158	158	57	57	57	69	81	57	0
158	158	158	158	158	57	57	81	57	57	57	0
0	0	0	0	0	0	0	0	0	0	0	0

Figura 9 - Exemplo de valores de pixel em tons de cinza de uma imagem.

Os segmentos detectados podem fundir-se, dividir-se ou desaparecer na linha seguinte. Por isso o método de crescimento por região normalmente resulta numa segmentação altamente fragmentada. Para obter regiões maiores pode-se, em paralelo ao processo de segmentação, agregar segmentos semelhantes devendo o processo ser guiado semanticamente pelo uso de propriedades da imagem. A relação de adjacência



entre regiões de propriedades variadas pode ser um critério (COMUNELLO, 2004). Pode-se observar que o conteúdo de um arquivo de imagem contém variados valores de tonalidades de cinza, como representado na Figura 9 e Figura 10. Estas figuras por sua vez, mostram como é realizado o processo de agregação de pixels semelhantes. Na Figura 11 visualiza-se o resultado de uma agregação de pixels semelhantes em regiões.

85	61	61	85	85	61	85	85	61	85	85	0
61	85	61	85	61	85	85	61	85	85	85	0
85	85	85	85	85	85	85	85	61	85	85	0
85	85	73	85	85	85	61	85	85	73	73	0
158	158	158	158	158	81	69	57	57	57	57	0
158	146	158	158	146	57	57	81	57	81	57	0
158	158	146	158	158	57	69	81	57	81	57	0
158	146	146	158	158	57	57	57	69	81	57	0
158	158	158	158	158	57	57	81	57	57	57	0
0	0	0	0	0	0	0	0	0	0	0	0

Figura 10 - Exemplo de valores de pixel em tons de cinza de uma imagem e sua representação em tons de cinza.



Figura 11 - Resultado de uma segmentação por crescimento de regiões, nos valores em tons de cinza.

Os algoritmos de segmentação baseados em crescimento de regiões mais utilizados e citados pela literatura são os algoritmos de WaterShed e Mumford&Shah (MUMFORD & SHAH, 1989). Estes se destacam por respectivamente serem ágeis e precisos no processo de segmentação.

### 3.5.2.1 Segmentação pela funcional de Mumford&Shah

O algoritmo de Mumford&Shah (MUMFORD & SHAH, 1989) é um método que usa a idéia de que as regiões são grupos de pixels com uma “borracha”. As regiões crescem enquanto pudermos esticar a borracha. Quanto maiores as variações entre os pixels dentro de uma região, tanto menor a elasticidade da borracha. Do ponto de vista matemático, o algoritmo de Mumford&Shah é genial pela sua simplicidade. De todos os algoritmos de crescimento de regiões, é um dos que produzem os melhores resultados para uso geral. Infelizmente sua complexidade computacional pode tornar-se um problema do ponto de vista do tempo, que é bastante longo porque dependendo das imagens, na maioria dos casos se necessita de um alto custo computacional para os complexos cálculos efetuados até a obtenção do resultado. É um algoritmo amplamente utilizado e conhecido como o mais preciso método de segmentação (COMUNELLO, 1999).

Este método de segmentação é baseado no modelo matemático descrito por (MUMFORD & SHAH, 1989), ou seja, a equação de energia funcional Mumford&Shah e é dada por:

$$E(u, K) = \iint_{\Omega} \|u - g\|^2 dx dy + \iint_{\frac{\Omega}{k}} \|\nabla u\|^2 dx dy + \lambda * l(k) \quad (11)$$

(MUMFORD & SHAH, 1989)

A descrição dos componentes da fórmula dá-se da seguinte maneira:

$E(u, K)$	Energia funcional de Mumford-Shah em função de $u$ e da fronteira $K$ .
$G(x, y)$	Função intensidade de luz no ponto $x, y$ .
$w$	Domínio da imagem, geralmente um retângulo.
$k$	Fronteira entre regiões.
$\Omega_i$	Decomposição da imagem em micro-regiões.
$\Omega$	$\Omega_1 \cup \Omega_2 \cup \dots \cup \Omega_n \cup K$ .
$l(K)$	Comprimento dos arcos das bordas (fronteiras).
$\mu(x, y)$	Função diferenciável dentro de cada região $\Omega_i$ .
$\lambda$	Parâmetro de escala.

A principal função da equação (11) é medir se  $\mu(x,y)$  é uma boa aproximação de  $g(x,y)$ . Quanto melhor a função  $\mu(x,y)$  se aproximar da função  $g(x,y)$ , menor será a contribuição desse termo para o valor da energia, e neste caso, significa que as regiões são semelhantes e interessantes para uma operação de união.

Segundo (PAULO, 2004), a equação (11) pode ser decomposta em 3 unidades funcionais, sendo que cada termo tem uma determinada responsabilidade durante o processo de segmentação. A equação (12) demonstra o termo que mede se  $\mu(x,y)$  é uma boa aproximação de  $g(x,y)$ .

$$\iint_{\Omega} \|u - g\|^2 dx dy \quad (12)$$

O próximo termo é definido pela função descrita na equação (13) (PAULO, 2004) e este calcula a variação mínima de  $\mu(x,y)$  dentro de cada região sem a borda. No nosso caso como a função  $\mu(x,y)$  será o valor médio dos pontos de  $g(x,y)$  dentro da região e portanto um valor constante, não existirá variação, assim este termo será sempre zero (PAULO, 2004).

$$\iint_{\frac{\Omega}{k}} \|\nabla u\|^2 dx dy \quad (13)$$

O terceiro termo descrito na equação (14) calcula o comprimento das fronteiras, multiplicado por um parâmetro de escala  $l$ . Quanto maior o comprimento das fronteiras, maior será a contribuição desse termo para o valor da energia. Ainda esse valor será multiplicado por um parâmetro que funciona como um peso (PAULO, 2004).

$$\lambda * l(k) \quad (14)$$

A equação funcional da energia de Mumford&Shah utiliza como base para seus cálculos a variância dos tons de cinza da imagem. Sendo assim, quanto maior a variância do tom de cinza maior e mais difícil é a união de regiões. Porém esta energia

dirá o quanto uma região poderá agrupar-se com outras regiões, delimitando as extremidades das regiões (PAULO, 2004).

Existe uma variação do modelo convencional da fórmula de Mumford&Shah, a qual é conhecida como energia simplificada de Mumford&Shah, descrita na equação:

$$E(u, K) = \iint_{\Omega} \|u - g\|^2 dx dy + \lambda * l(k) \quad (15)$$

Para a equação (15), assume-se  $\mu(x,y)$  como função constante dentro de cada região, a qual é a média dos valores de  $g(x,y)$  nesta região. A função da energia simplificada de Mumford&Shah fica em função apenas de  $K$  (fronteiras das regiões).

Quando não for mais possível juntar nenhuma região, deve-se aumentar o valor de lambda, e essa variação de lambda pode ser linear ou exponencial.

Resumidamente, o algoritmo de Mumford&Shah atua na imagem fazendo com que as regiões realizem uma competição entre si em busca de outras regiões para realizarem uma fusão. Realiza isto através de seus cálculos de equação funcional, verificando a variância dos tons de cinza na imagem, ou seja, quanto maior a variância entre essas regiões, menor é a probabilidade de unirem-se, em contrapartida, o contrário ocorre quando estas possuem alguma similaridade de luminosidade (MUMFORD & SHAH, 1989).

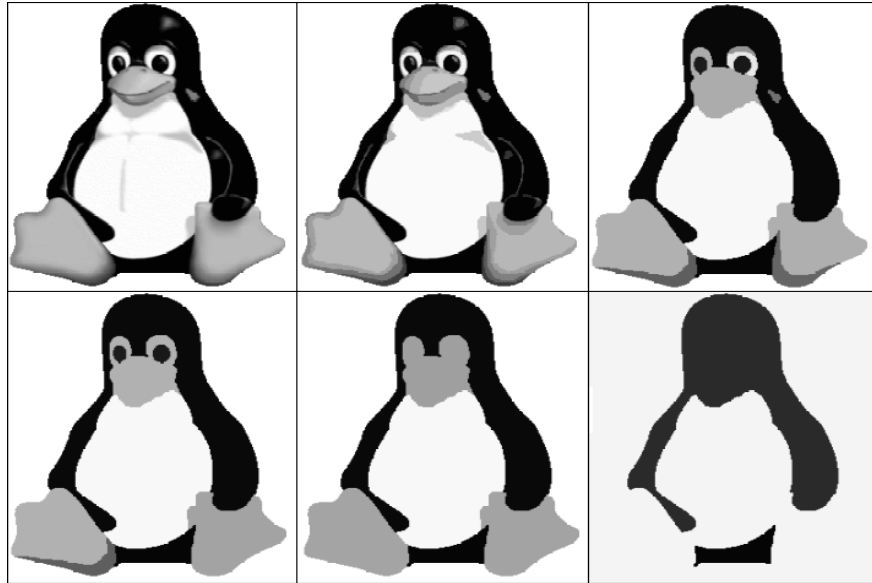


Figura 12- Diferentes resultados de segmentação por crescimento de regiões definidos a partir de diferentes valores de  $\lambda$ .

A Figura 12 mostra alguns resultados de segmentação por crescimento de regiões definidas a partir de diferentes valores de  $\lambda$ . O  $\lambda$  na funcional de Mumford & Shah determina quanta energia pode ser usada para agregar regiões em uma imagem. Pode-se observar claramente como o coeficiente de agregação  $\lambda$  (lambda) tem grande influência na separação ou junção de regiões, devendo este ser alterado de acordo com cada imagem de trabalho ou necessidade de segmentação.

### Modelo Discreto

Dadas duas regiões adjacentes  $O_i$  e  $O_j$  separadas por uma fronteira comum  $\partial(O_i, O_j)$  e o valor de energia  $E(u, K)$ . Uma nova segmentação  $(u', K')$  chamada de sub-segmentação, será obtida, removendo-se a fronteira comum  $\partial(O_i, O_j)$  dessas duas regiões adjacentes. Nesta nova segmentação  $(u', K') = ((u, K) \text{ sem } \partial(O_i, O_j))$  se a energia da nova sub-segmentação for menor que a energia anterior, ou seja,  $E(u', K') < E(u, K)$  então, é interessante a junção dessas regiões. Porém antes de juntarmos estas duas regiões devemos fazer o mesmo para todas as regiões vizinhas a estas duas regiões  $O_i$  e  $O_j$  e verificarmos qual é a junção que ocasionará o maior decréscimo de energia. Quando não for mais possível juntar nenhuma região, deve-se aumentar o valor de  $\lambda$ , o qual pode ser linear ou exponencial. O critério de juntar duas regiões  $O_i$  e  $O_j$  está na dependência do sinal de  $E(K) > (O_i, O_j) - E(K)$ , no qual é apenas o decréscimo

de energia. Assim a equação descreve o critério de energia no modelo discreto (WANGENHEIM, 2007).

$$E(E(K) - \partial(O_i, O_j)) = \frac{|O_i| * |O_j|}{|O_i| + |O_j|} * \|u - g\|^2 - \lambda * l\partial(O_i, O_j) \quad (16)$$

(WANGENHEIM, 2007)

Pela equação (16),  $|O_i|$  e  $|O_j|$  são respectivamente as áreas das regiões  $O_i$  e  $O_j$ ;  $u_i$  e  $u_j$  correspondem ao valor de intensidade da região  $i$  e  $j$ . Finalmente  $\lambda * l\partial(O_i, O_j)$ , descreve a relação de adjacência das regiões  $O_i$  e  $O_j$ .

Dos algoritmos de crescimento de regiões em uso, em geral, Mumford&Shah é o que produz melhores resultados. Atualmente o algoritmo é amplamente utilizado e conhecido por ser um método significativamente preciso para segmentação por crescimento de regiões (WANGENHEIM, 2007).

### 3.5.3 Segmentação por Detecção de Borda

O processamento digital de imagens envolve a identificação e classificação de objetos em uma imagem. Na segmentação procura-se distinguir as partículas umas das outras e do fundo (GONZALEZ & WOODS, 2002). Desta forma a detecção de bordas é uma ferramenta muito útil no processo de análise de imagens, pois de uma forma ou de outra, a borda é a extremidade de um determinado objeto a ser isolado. A importância da detecção de borda dá-se devido ao fato de que ao se detectar uma borda pode-se localizar os objetos de uma imagem (PARKER, 1997). Esta distinção permitirá ao programa interpretar pixels contínuos e agrupá-los em regiões.

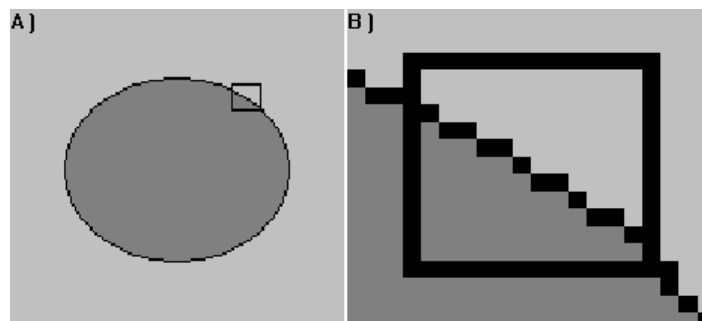


Figura 13 - Borda de um objeto, em A) imagem original, em B) zoom da imagem para observação de borda.

Considera-se borda, toda e qualquer mudança brusca da luminosidade em uma imagem. Com essa característica básica, define-se, praticamente, como localizar as extremidades de objetos em imagens como mostra a Figura 13. De uma maneira mais simplificada, borda é o contorno entre um objeto e seu fundo, sugere o limite entre objetos sobrepostos, sendo uma linha fechada, formando assim, a borda dos objetos (GONZALEZ & WOODS, 2002).

A detecção de borda teoricamente pode ser facilmente implementada. Tem-se um número limitado de algoritmos para esse fim, onde cada um, com seus qualificadores e especialidades, produzem diferentes resultados. A maioria dos algoritmos descritos para esta tarefa faz uso de operadores sensíveis à descontinuidade abrupta, os quais irão operar como detector de borda.

Os operadores responsáveis por essa função são os operadores de derivada, os quais irão atuar de maneira que onde a taxa de mudança dos níveis de cinza de uma imagem for grande, aquele pixel será considerado como uma pixel de borda, e nas demais áreas, será considerado como sendo do corpo da imagem em si. Caso seja determinado o conjunto de pontos onde a derivada da função é o ponto máximo, nestas serão localizadas as suas bordas. Dado que as imagens são em duas dimensões, é importante considerar mudanças nos níveis de cinza em muitas direções. Por esta razão, derivadas parciais das imagens são usadas com as respectivas direções X e Y (SEARA, 1998).

### **3.5.3.1 Detecção de Bordas pelas Máscaras de Sobel**

A detecção de borda pelo método de Sobel é simples e, no entanto, bastante eficiente do ponto de vista do relativo baixo custo computacional necessário para computar bordas.

Em geral podem-se computar bordas de uma imagem por meio de derivadas espaciais, ou seja, a taxa de variação em função da coordenada X e de Y. Para isso um operador comumente usado é o operador chamado de gradiente, que é um vetor que indica os locais nos quais os níveis de cinza sofrem maior variação (PEDRINI & SCHWARTZ, 2008).

$$\nabla f = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} \quad (17) \text{ (PEDRINI \& SCHWARTZ, 2008)}$$

(Gradiente, na forma matricial)

O vetor gradiente  $\nabla f$  na posição  $(x,y)$  pode ser expresso na forma matricial conforme equação (17). Onde  $G_x$  e  $G_y$  são respectivamente derivadas parciais para coordenada X e Y, ou seja, a variação em função de X e Y.

Para calcular as derivadas em X e Y utilizam-se então, para o método de Sobel, as máscaras dadas pela equação (18) e (19), e realiza-se uma operação de convolução da imagem  $I(x,y)$  com ambas as máscaras conforme equações (20) e (21).

$$h1 = \begin{bmatrix} +1 & 0 & -1 \\ +2 & 0 & -2 \\ +1 & 0 & -1 \end{bmatrix} \quad (18)$$

(Máscara para cálculo da variação em X, Máscara de Sobel.)

$$h2 = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \quad (19)$$

(Máscara para cálculo da variação em Y, Máscara de Sobel.)

Substituindo as matrizes pelas variáveis h1 e h2:

$$G_x(x, y) = h1 * I(x, y) \quad (20)$$

$$G_y(x, y) = h2 * I(x, y) \quad (21)$$

Tendo as derivadas calculadas, logo em seguida calcula-se a magnitude. A magnitude equivale a maior taxa de variação de uma imagem  $I(x,y)$  por unidade de distância na direção  $\nabla f$  (PEDRINI & SCHWARTZ, 2008). A magnitude  $G$  é dada pela equação (22).

$$G = \sqrt{G_x^2 + G_y^2} \quad (22)$$



A resposta da magnitude  $G$  é um escalar que indica o quanto de variação ocorre no ponto  $(x,y)$  da imagem  $I$ , sendo possível também calcular o ângulo da direção do vetor pela operação da equação (23).

$$\theta(x,y) = \arctan\left(\frac{Gy}{Gx}\right) \quad (23)$$

### 3.5.3.2 Gradientes pelas Máscaras de Simetria Positiva

Trata-se de máscaras calculadas usando a propriedade “Preservation of Positive Semi-definiteness”, definida pelo trabalho de (BROX, 2006). Esta propriedade garante que em nenhum caso a variação do gradiente possa ser anulado.

As equações (24) e (25) demonstram respectivamente as máscaras para o cálculo da variação do gradiente em X e em Y, onde  $a = 2(\sqrt{2}-1)$  e  $b = (2-\sqrt{2})$ .

$$Gx = 1/4 \begin{bmatrix} -b & 0 & b \\ -a & 0 & a \\ -b & 0 & b \end{bmatrix} \quad (24)$$

(Máscara para cálculo da variação em X.)

O interessante da resposta dessas máscaras é que junto com tensores de estrutura, o maior auto-vetor nos dá uma estimativa do quadrado do módulo do gradiente e isso é usado como um estimador de borda muito robusto (CASELLES, 2007).

$$Gy = 1/4 \begin{bmatrix} -b & -a & -b \\ 0 & 0 & 0 \\ b & a & b \end{bmatrix} \quad (25)$$

(Máscara para cálculo da variação em Y.)

### 3.5.3.3 Detecção de Bordas pelo Método de Canny

O detector de borda de Canny é um dos mais conhecidos e foi elaborado por Jonh Canny (PARKER, 1997) em 1986, sendo hoje amplamente utilizado em sistemas de processamento digital de imagens modernos.

O detector de bordas Canny é um operador gaussiano de derivada primeira que suaviza os ruídos e localiza as bordas (CANNY, 1986). A equação (26) calcula as derivadas para imagens bidimensionais em x e em y.

$$G(x, y) = \sigma^2 e^{-\left(\frac{x^2+y^2}{2\sigma^2}\right)} \quad (26) \text{ (PARKER, 1997)}$$

Seu sucesso deu-se por conseguir definir e implementar parâmetros que um bom detector de bordas deveria possuir. Na criação do algoritmo Canny, definiu-se um conjunto de requisitos que um detector de bordas deveria satisfazer e são eles (PARKER, 1997):

- Taxa de erro: o detector de bordas deveria detectar e achar somente bordas e nenhuma borda deveria faltar;
- Localização: a distância entre os pixels de borda encontradas pelo detector de bordas e a borda atual deveria ser a menor possível;
- Resposta: o detector de bordas não deveria identificar múltiplos pixels de borda onde somente exista um único pixel.

Para o detector de Canny uniformizar ruídos e localizar bordas eficientemente, depende da identificação do filtro que aperfeiçoe os três critérios para o detector de borda acima estabelecidos.

Para que fosse possível desenvolver este algoritmo, era necessário definir anteriormente o padrão das bordas ideais, onde a representação é realizada por função que possua apenas uma dimensão. Não é um modelo completamente exato, porém, representa os efeitos causados pelos ruídos (BUENO, 2000).

Segundo (BUENO, 2000), para que essas regras sejam satisfeitas, o detector Canny utiliza-se de quatro passos:

1. Uniformização ou homogeneidade da imagem: Como convoluções gaussianas de tamanho elevado levam muito tempo para serem calculadas, as mesmas são aproximadas por duas funções, uma no eixo x e outra y, estas gaussianas unidimensionais. Se borda for considerada uma dimensão que varia no contraste e esta ser “convolucionada” com a função de

uniformização de Gauss, o resultado sempre será uma variação contínua que se dará do valor inicial ao final, obtendo uma inclinação máxima, apenas onde encontrar um “degrau”.

2. Diferenciação : Depois de efetuada a convolução bidimensional, a imagem é separada nas direções x e y, para calcular o gradiente da superfície uniforme. Os valores de x são convolucionados utilizando a primeira derivada gaussiana com o alinhamento na direção de y; o inverso ocorre na convolução de y em x. Isto resultará na medida da intensidade que cada ponto possui na imagem.
3. Exclusão de pixels com baixa intensidade : Após os pontos já possuírem a medida da intensidade, o próximo passo é a localização das bordas. Isto se dá através da estimulação dos pontos com maior destaque, ou, inversamente, omitindo os não destacáveis. Os pontos em regiões distantes à borda são facilmente ocultados, porém, em contrapartida, os pontos próximos ou perpendiculares à borda devem ser avaliados para não eliminar a própria borda.
4. Threshold da borda: assim como na segmentação, na detecção da borda também se confronta com o problema do threshold. Mais comumente denominado de “limiarização” na detecção de bordas, este utiliza um método denominado de “hysteresis”. Alguns algoritmos de Canny possuem incorporados detectores de intensidade, os quais sugerem um bom limiar de threshold, porém falho em alguns casos. Para considerar borda, tudo o que estiver nos níveis de limiarização será considerado zero (preto), e inversamente, o que não estiver entre os níveis 255 (branco).

O detector de bordas de Canny merece grande consideração dentre o processo de detecção de bordas. A intensidade computacional do detector de bordas de Canny é relativamente alta, e os resultados são geralmente pós-processados para maior clareza. A Figura 14 demonstra os resultados da detecção de bordas pelo método de Canny.

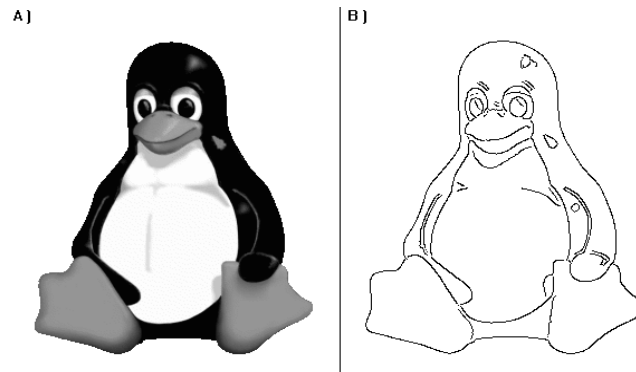


Figura 14 - Em A) imagem original, em B) imagem submetida à detecção de bordas do algoritmo de Canny.

### 3.6 Métricas de Qualidade em Imagens

Uma imagem pode sofrer degradação durante várias etapas de um sistema de processamento de imagens. Métricas de qualidade são técnicas empregadas para avaliar a similaridade de imagens que sofreram qualquer processo de transformação, em relação a uma imagem original sem ruído.

Algumas dessas métricas de qualidade são voltadas a avaliações subjetivas, as quais se baseiam na análise realizada por observadores humanos (PEDRINI & SCHWARTZ, 2008). Do outro lado tem-se técnicas objetivas cuja função em geral é comparar a quantidade de degradação da imagem de entrada no sistema sem intervenção humana. Ambas são úteis para casos diferenciados, mas em geral técnicas para comparação de imagens de especialistas humanos são muito mais complexas que uma avaliação do ponto de vista numérico.

#### 3.6.1 Erro Máximo

O erro máximo é a maior diferença absoluta entre cada ponto da imagem original e da imagem analisada,  $E$  é calculado com a equação (27) para duas imagens de entrada  $f$  e  $g$ .

$$E = \max_{x,y} |f(x,y) - g(x,y)| \quad (27)$$

Quanto menor o resultado dessa métrica significa que menos degradação a imagem de entrada  $f$  sofreu na saída em  $g$ . Esta técnica é muito sensível ao ruído de imagens (PEDRINI & SCHWARTZ, 2008).

### 3.6.2 Erro Médio Absoluto

Erro médio absoluto é calculado com a soma da diferença entre cada pixel da imagem original e da imagem aproximada, dividido pelo número de pixels da imagem, e dada para duas imagens  $f$  e  $g$  com dimensões  $X$  e  $Y$ , pela equação (28). Mais uma vez quanto menor o valor da saída desta técnica menor é a degradação do sinal da imagem original  $f$  (PEDRINI & SCHWARTZ, 2008).

$$E = \frac{1}{X * Y} \sum_{x=0}^{X-1} \sum_{y=0}^{Y-1} |f(x, y) - g(x, y)| \quad (28)$$

### 3.6.3 Erro Médio Quadrático

É a soma do quadrado das diferenças de cada ponto da imagem original  $f$  e da imagem aproximada  $g$ , dividida pelo número de pixels da imagem (ambas têm a mesma dimensão) (PEDRINI & SCHWARTZ, 2008). É expresso pela equação (29).

$$E = \frac{1}{X * Y} \sum_{x=0}^{X-1} \sum_{y=0}^{Y-1} |f(x, y) - g(x, y)|^2 \quad (29)$$

### 3.6.4 Medidas de Dissimilaridade em Imagens

A seguir serão apontados dois algoritmos amplamente utilizados para comparação entre resultados de segmentação. Estes algoritmos basicamente medem a dissimilaridade entre clusters de dados, gerando no final índices, este variam de 0 a 1. Por se tratar de medidas de dissimilaridade, quanto menor o índice, mais próximos são os clusters de dados, ou seja, mais similares.

#### 3.6.4.1 Índice Rand

Rand Index, desenvolvido por (RAND, 1971), é uma medida de similaridade para verificar a qualidade de algoritmos de clustering. Em 2006 foi revisado por (JIANG et. al., 2006).

A computação é realizada através da comparação com outros resultados de *clustering* ou com um padrão ouro. Para comparar dois resultados consideramos dois *clusters*  $C1 = \{c1_1, c1_2, \dots, c1_N\}$  e  $C2 = \{c2_1, c2_2, \dots, c2_M\}$  na mesma imagem  $P = \{p_1, p_2, \dots, p_K\}$  onde cada elemento de  $C1$  ou  $C2$  é um subconjunto de  $P$  e  $c1_j = \{p1_j, p2_j, \dots, pL_j\}$ , as seguintes quantidades são calculadas:

- $N_{11}$  – o número de *pixels* convergentes de ambos os *clusters*  $C1$  e  $C2$ .
- $N_{00}$  – o número de *pixels* divergentes de ambos os *clusters*  $C1$  e  $C2$ .

O índice Rand é definido pela Equação (30).

$$R(C1, C2) = 1 - \frac{N_{11} + N_{00}}{\frac{n(n-1)}{2}} \quad (30)$$

Para computar as quantidades de  $N_{11}$  e  $N_{00}$ , um deles deve iterar sobre toda a imagem, por cada pixel, para poder avaliar as condições definidas anteriormente. Uma solução mais prática é usar uma matriz de correspondência (JIANG et. al., 2006), a qual diminui ocorrências dos pixels na classe respectiva.

A matriz de correspondência é construída alocando cada cluster de  $C1$  para uma linha e cada cluster de  $C2$  para uma coluna. As células da matriz são então definidas como uma intersecção dos clusters especificando cada linha e coluna. Se a matriz de correspondência tem um tamanho  $k \times l$ , cada célula pode ser definida como  $m_{ij} = |c_i \cap c_j|$ ,  $c_i \in C1$ ,  $c_j \in C2$ . As quantidades de  $N_{11}$  e  $N_{00}$  podem ser computadas pelas condições da matriz de correspondência como segue:

$$N_{11} = \frac{1}{2} \left( \sum_{i=1}^k \sum_{j=1}^l m_{ij}^2 - n \right) \quad (31)$$

$$N_{00} = \frac{1}{2} \left( n^2 - \sum_{i=1}^k n_i^2 - \sum_{j=1}^l n_j^2 + \sum_{i=1}^k \sum_{j=1}^l m_{ij}^2 \right) \quad (32)$$

onde  $n$  é a cardinalidade de  $P$  e  $n_i$  e  $n_j$  são as cardinalidades dos *clusters*  $c1_i$  e  $c2_j$ .

### 3.6.4.2 Correspondência de Grafos Bipartidos (BGM)

O índice BGM calcula a correlação de um para um entre os clusters e ao mesmo tempo tenta maximizar a sua correlação. O BGM considera cada cluster de  $C1$  e  $C2$  como vértices de um grafo bipartido. Bordas são adicionadas entre cada vértice das duas partições e elas são estimadas como  $|c1_i \cap c2_j|$ , um valor que pode ser extraído diretamente da matriz de correspondência. Então o grafo bipartido com peso máximo é definido como o sub-grafo  $\{(c1_{i1}, c2_{j1}), \dots, (c1_{ir}, c2_{jr})\}$ , onde apenas as bordas de  $c1_i$  a  $c2_j$  com peso máximo estão presentes. Depois que todas as bordas com valores máximos foram encontradas, o peso geral do grafo é calculado pela soma de todos os pesos de bordas restantes (JIANG et. al., 2006).

## ***Capítulo 4***

### ***Desenvolvimento***

Este capítulo descreve detalhadamente o procedimento para o desenvolvimento da solução. As etapas deste capítulo são: uma visão geral sobre o filtro de difusão adotado, detalhes de implementação do filtro convencional, algumas dificuldades que motivaram a melhoria do filtro tradicional e por fim, duas abordagens propostas para melhoria do filtro tradicional.

#### ***4.1 Filtro de Difusão Anisotrópico com Lambda Adaptativo e Orientação por Mapas de Gradientes***

O filtro de difusão anisotrópico utilizado neste trabalho é uma interpretação das equações de (WEICKERT, 2001), com o uso de um kernel adaptativo ou matriz de difusibilidade, pela qual, com o uso de seus coeficientes resultará no valor do pixel sob processo de esmaecimento. O tensor de difusão é calculado para cada pixel, em cada iteração do algoritmo, e possui a capacidade de esmaecer fracamente quando este é um pixel em uma borda ou fortemente acoplado em regiões internas, de acordo com os valores de vizinhança do pixel.

Para melhor compreensão, nos tópicos a seguir são apresentados mais detalhes sobre como o filtro de difusão foi implementado.

##### **4.1.1 Aspectos de Implementação**

O filtro anisotrópico é calculado como uma máscara de convolução aplicada em cada ponto da imagem inicial, repetidas vezes. O número de repetições é um parâmetro definido pelo usuário como número de iterações. Outro importante valor especificado pelo usuário é o que se chama de lambda. O lambda determina dentro do algoritmo quão forte será a difusão, ou seja, grandes valores de lambda fazem com que a resposta de esmaecimento seja maior.



Como mencionado anteriormente, verifica-se para o cálculo do pixel as informações locais da imagem, isto significa que a vizinhança é avaliada no processamento de um pixel. Geralmente os 8 vizinhos são utilizados para isto, e em seguida uma operação matricial é realizada conforme a equação (33):

$$P = M * W \quad (33)$$

onde  $P$  é o valor de saída do produto das máscaras  $M$  e  $W$ .

Consideramos para essa operação as matrizes  $M$  e  $W$  das equações (34) e (35):

$$M = \begin{bmatrix} p_1 & p_2 & p_3 \\ p_4 & P & p_5 \\ p_6 & p_7 & p_8 \end{bmatrix} \quad (34)$$

(Matriz de pixels vizinhos de  $P$ .)

e

$$W = \begin{bmatrix} w_1 & w_2 & w_3 \\ w_4 & w_C & w_5 \\ w_6 & w_7 & w_8 \end{bmatrix} \quad (35)$$

(Matriz de pesos calculados para posterior aplicação no filtro.)

A matriz  $M$  detém os pixels  $p_1 \dots p_8$  vizinhos de  $P$ , e a matriz  $W$  são os pesos  $w_1 \dots w_8$ .

É importante mencionar que o kernel  $W$  é calculado para cada pixel a cada iteração, e este kernel adaptativo  $W$  é o tensor de difusão citado por (WEICKERT, 1998), onde cada coeficiente do tensor é calculado pela equação (36):

$$w(\lambda, \tau, \alpha, \beta)_i = \frac{1 - e^{-8*\tau*e^{-\frac{(|\sqrt[5]{\alpha-\beta}|}{\lambda})^5}}}{8} \quad (36)$$

onde  $\alpha$  e  $\beta$  são respectivamente vetores ou gradientes ( $\alpha$  é o pixel central  $f(x,y)$  e  $\beta$  é um dos 8 vizinhos) a variável  $\tau$ , é conhecida como “tamanho de passo de tempo” (*time step size*), esta variável decorre da necessidade de emular um processo físico de transporte de

massa por um processo iterativo de estabilização, onde  $\tau$  é a variação de tempo  $t$  utilizada para difusão. É definida nos intervalos de 0 a 0,5, como segue (WEICKERT, 1998):

$$\tau = \Delta t (t \mid 0 < t < 0.5)$$

A equação (36) processa os coeficientes de difusibilidade de dois gradientes em escala de cinza, mas pode ser estendida para um sistema de cores, por exemplo, o sistema RGB. Usando a métrica de distância linear euclidiana, fica então definida pela equação (37) em:

$$w(\lambda, \tau, \alpha, \beta)_i = \frac{1 - e^{-8*\tau*e}}{8} \left( \frac{\sqrt[5]{\sqrt[2]{(\alpha R - \beta R)^2 + (\alpha G - \beta G)^2 + (\alpha B - \beta B)^2}}}{\lambda} \right) \quad (37)$$

Na extensão para o caso de uso em imagens coloridas,  $\alpha R$  e  $\beta R$  possuem o valor da cor vermelha do ponto,  $\alpha G$  e  $\beta G$  possuem o valor da cor verde e  $\alpha B$  e  $\beta B$  possuem o valor de azul.

O coeficiente central  $wC$  de  $W$  é dado pela operação da equação :

$$wC = \sum_{i=1}^8 (1 - w_i) \quad (38)$$

Considerando o mapeamento de pontos da imagem em um plano cartesiano, o kernel  $W$  é criado da seguinte maneira (equação (39)):

$$W = \begin{pmatrix} (w_1(\lambda, \tau, x - 1, y + 1)) & (w_2(\lambda, \tau, x, y + 1)) & (w_3(\lambda, \tau, x + 1, y + 1)) \\ (w_4(\lambda, \tau, x - 1, y)) & (wC) & (w_5(\lambda, \tau, x + 1, y)) \\ (w_6(\lambda, \tau, x - 1, y - 1)) & (w_7(\lambda, \tau, x, y - 1)) & (w_8(\lambda, \tau, x + 1, y - 1)) \end{pmatrix} \quad (39)$$

A equação (39) define a criação do kernel ou tensor de difusão. Após ter sido criado o kernel para o pixel corrente, a operação de convolução da equação (33) pode ser realizada.

Abaixo, na Figura 15 pode ser visualizado o pseudocódigo do algoritmo da difusão anisotrópica aplicada neste trabalho.

```

inicio
ler lambda;
ler iterações;
imagem_atual = Ler_imagem_original;
Imagem_nova = Criar_imagem_vazia(0)
Para numero de iteracoes
inicio iteracoes
Para PIXELS em imagem_atual
ler p1(x - 1, y + 1) em imagem_atual;
ler p2(x, y + 1) em imagem_atual;
ler p3(x + 1, y + 1) em imagem_atual;
ler p4(x - 1, y) em imagem_atual;
ler P(x, y) em imagem_atual;
ler p5(x + 1, y) em imagem_atual;
ler p6(x - 1, y - 1) em imagem_atual;
ler p7(x, y - 1) em imagem_atual;
ler p8(x + 1, y - 1) em imagem_atual;
vizinhos_P = Vizinhos(p1,p2,p3,p4,p5,p6,p7,p8,P);
W = Criar_kernel(p1, p2, p3, p4, p5, p6, p7,p8, P, lambda);
pixel_novo = Convolucao(vizinhos_P,W);
Salva(imagem_nova, posicao_atual, pixel_novo);
fim PIXELS
imagem_atual = imagem_nova;
fim iterações
fim

```

Figura 15 – Pseudocódigo do filtro de difusão anisotrópico.

## 4.2 Contribuições do Trabalho

Do ponto de vista da aplicação do filtro de difusão como técnica de esmaecimento, somos capazes apontar duas principais características que podem ser melhoradas:

- para o caso onde muitas iterações são necessárias, existe a tendência de que regiões de bordas tênues não serem preservadas. Isto porque, no processo de estabilização da difusão em regiões de bordas tênues, ocorre muito antes do esmaecimento desejado. Assim ao longo do tempo essas bordas acabam sendo difundidas e não serão mais evidentes ou claras. Este problema é verificado quando um sistema de esmaecimento menos criterioso é aplicado, por exemplo, um esmaecimento por kernel gaussiano.

- quando o usuário pretende usar o filtro de difusão anisotrópico (dada suas características) como pré-segmentador, ele pode não esmaecer regiões nitidamente necessárias, assim o que se pretende é ter forte esmaecimento em regiões não fronteiriças ou de baixa variação e baixo esmaecimento em regiões de borda.

Para amenizar os efeitos citados acima duas estratégias são usadas: uso do mapa de pesos por gradientes para orientação da difusibilidade e adaptação do lambda, ambas tratadas com maiores detalhes a seguir.

### 4.2.1 Contribuição de Mapa de Pesos por Gradientes

Para compreender a aplicação de gradientes dentro da filtragem anisotrópica reveremos a equação (37) e dividi-la em duas partes funcionais:

$$C(\lambda, \alpha, \beta) = e^{-\left(\frac{\sqrt[5]{\sqrt{(\alpha R - \beta R)^2 + (\alpha G - \beta G)^2 + (\alpha B - \beta B)^2}}}{\lambda}\right)^5} \quad (40)$$

e

$$w(\lambda, \tau, \alpha, \beta)_i = \frac{1 - e^{-8*\tau*C(\lambda, \alpha, \beta)}}{8} \quad (41)$$

A equação (40) é uma função monotônica que irá retornar 0 ou 1 correspondendo à difusibilidade de  $\alpha$  para  $\beta$ . Esta parte da equação de difusibilidade será objeto de aplicação de gradientes; estes gradientes por sua vez, podem ser gerados por qualquer técnica a ser escolhida e correspondem à variação local em relação às coordenada  $x$  e  $y$  da imagem.

A computação das derivadas em relação a  $x$  e  $y$  já devem estar previamente calculadas ou pelo menos antes da realização da equação (40) elas devem ser calculadas. Neste trabalho são utilizadas as máscaras com simetria positiva definida pelo trabalho de (BROX, 2006) e apresentadas no item 3.5.3.2, página 29.

Sendo assim, de posse da magnitude do pixel  $\alpha$ ,  $G(\alpha)$  no caso de imagens tratadas no sistema de cores RGB, corresponde à magnitude dos 3 canais de cor. Estes valores para imagens coloridas variam geralmente de 0 a 255 por canal (256 níveis para cada canal ou 8 bits) e quanto maior for este valor, maior será a contribuição deste canal para a inibição ou excitação da difusão neste ponto. Logo, a equação (40) e (41) podem ser re-escritas respectivamente como:

$$C(\lambda, \alpha, \beta, G(\alpha)) = e^{-\left( \frac{\sqrt[5]{\sqrt[2]{((\alpha R - \beta R)^2 * G(\alpha R)) + ((\alpha G - \beta G)^2 * G(\alpha G)) + ((\alpha B - \beta B)^2 * G(\alpha B))}}}{\lambda} \right)} \quad (42)$$

e

$$w(\lambda, \tau, \alpha, \beta, G(\alpha))_i = \frac{1 - e^{-8*\tau*C(\lambda, \alpha, \beta, G(\alpha))}}{8} \quad (43)$$

Assim é adicionada a contribuição de peso das bordas a difusibilidade, ou mapa de gradientes, e os efeitos da aplicação desta técnica é visualizado em (d) na Figura 16.

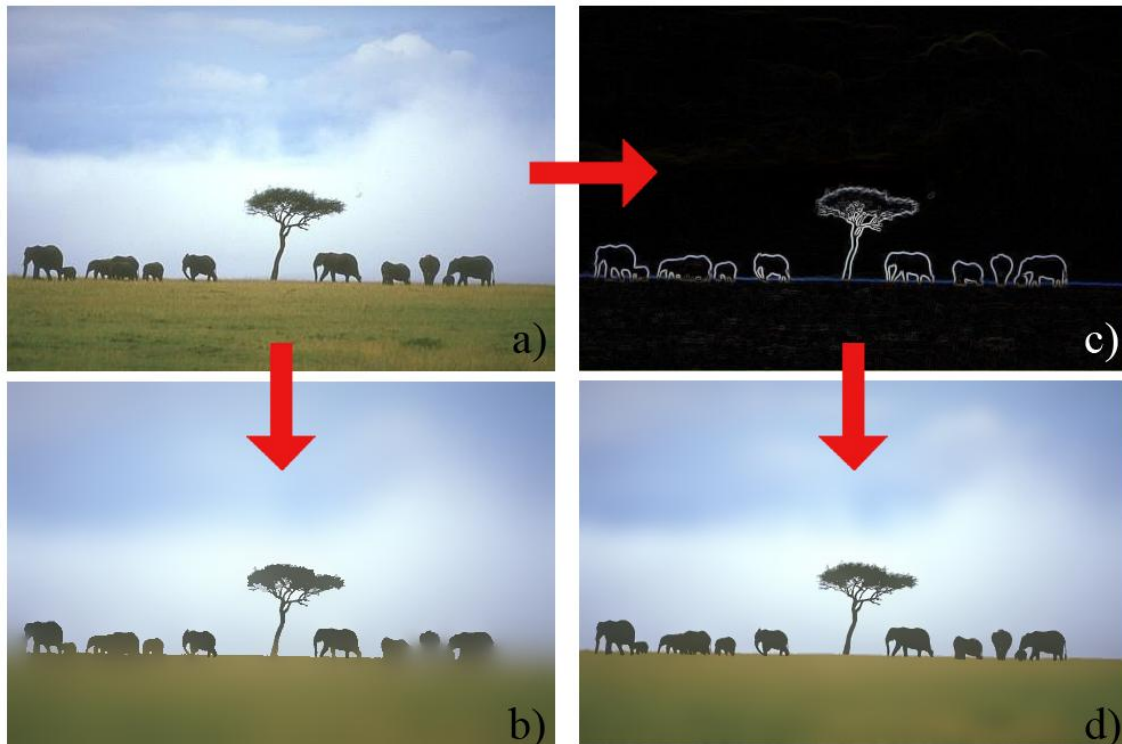


Figura 16 – Resultados da aplicação de filtro de difusão tradicional e orientado por mapa de gradientes. (a) imagem original; (b) imagem filtrada com o filtro tradicional; (c) imagem de gradiente; d) aplicação do filtro com os pesos de (c).

Na Figura 16 podemos ver em (a) a imagem original. Em (b) é o resultado do filtro tradicional com  $\lambda = 30$  e 300 iterações do algoritmo, em (c) é o resultado do cálculo dos gradientes coloridos de borda. A imagem em (d) corresponde à imagem filtrada com acoplamento de mapa de pesos da imagem (c) usando os mesmos parâmetros aplicados na filtragem da imagem (b).

#### 4.2.2 Lambda Adaptativo

Lambda ( $\lambda$ ) adaptativo ou  $\lambda$  dinâmico é uma característica da equação (40) que também exploramos. O parâmetro de  $\lambda$  determina o quanto a difusão pode ser forte ou fraca na avaliação do pixel  $\alpha$  e  $\beta$ . Quanto maior o valor de  $\lambda$  maior será o coeficiente de difusão naquela região ( $\lambda > 0 > \infty$ ).

Para atuar modificando o parâmetro  $\lambda$  a característica explorada na função  $C$  da equação (40) é a distância entre os pontos  $\alpha$  e  $\beta$ , deste ponto em diante mencionada como  $\Delta(\alpha, \beta)$  conforme equação (44). Esta é calculada por distância euclidiana (métrica linear de medida que pode ser usada dentro do modelo de representação do espaço de

cores RGB). Grandes variações de  $\Delta(\alpha, \beta)$  indicam uma dissimilaridade de cor muito forte, o que passa a ser um indicativo de possível borda. Então, basicamente o que temos que fazer é ajustar o  $\lambda$  (aumentando ou diminuindo) para modificar a resposta da função  $C$  da equação (40), atendendo os requisitos de baixo esmaecimento da borda.

$$\Delta(\alpha, \beta) = \sqrt[2]{(\alpha R - \beta R)^2 + (\alpha G - \beta G)^2 + (\alpha B - \beta B)^2} \quad (44)$$

Para avaliar o quanto uma dada distância varia, usamos a média estatística global ( $\mu$ ) e desvio padrão ( $\sigma$ ) de RGB da imagem. O uso de descritores estatísticos é justificável porque verifica-se que toda distribuição estatística tem tendência de convergir para o modelo normal (aumentando-se o número de amostras), e se os dados se distribuem de maneira semelhante à distribuição normal então os dados possuem a seguinte característica:

- 95% dos dados estão a 2 desvios padrão para mais e para menos, distantes da média, e;
- 99% dos dados estão a 3 desvios padrão para mais e para menos, distantes da média (BARBETTA, 2006);

Assim, o critério de modificação de  $\lambda$  varia de acordo com as medidas estatísticas previamente calculadas de média de cor de cada canal e desvio padrão. Estas são informações globais da imagem e não somente informações locais, como habitualmente são tratados os dados em operações de filtragem.

Logo, basicamente a estratégia adotada para reajuste do  $\lambda$  é a adição ou subtração de percentuais do valor de  $\lambda$  previamente especificado. O  $\lambda$  será re-escalado em um de 4 valores percentuais possíveis, valores estes que foram arbitrariamente escolhidos e adotados sem nenhum estudo sobre o impacto destes percentuais no processo de difusão.

Então para inibição do  $\lambda$ , ele é decrescido em 25 % ou 50 %, e para excitação o  $\lambda$  será acrescido também em 25% ou 50%. Para isto, considera-se a equação (45) do seguinte sistema de classificação  $A$  para soluções de  $\lambda$ :

$$\Lambda(\Delta(\alpha, \beta), \mu, \sigma, \lambda) = \left\{ \begin{array}{l} r_1 = (\lambda * 0.50) \leftrightarrow ((\Delta(\alpha, \beta) > \mu) \wedge (\Delta(\alpha, \beta) > (\mu + \sigma))) \\ r_2 = (\lambda * 0.75) \leftrightarrow ((\Delta(\alpha, \beta) > \mu) \wedge (\Delta(\alpha, \beta) < (\mu + \sigma))) \\ r_3 = \lambda \leftrightarrow (\Delta(\alpha, \beta) = \mu) \\ r_4 = (\lambda * 1.25) \leftrightarrow ((\Delta(\alpha, \beta) < \mu) \wedge (\Delta(\alpha, \beta) > (\mu - \sigma))) \\ r_5 = (\lambda * 1.50) \leftrightarrow ((\Delta(\alpha, \beta) < \mu) \wedge (\Delta(\alpha, \beta) < (\mu - \sigma))) \end{array} \right\} \quad (45)$$

Sendo assim o conjunto de soluções do sistema  $\Lambda$  assume um valor  $r_i$ , ou seja, uma escala de  $\lambda$  segundo  $\Delta(\alpha, \beta)$ ,  $\mu$  e  $\sigma$ , que será para  $r_1$  o  $\lambda$  menos 25 %, para  $r_2$  o  $\lambda$  menos 50 %, para  $r_3$  o  $\lambda$  mais 25 % e  $r_4$  o  $\lambda$  mais 50 %. Pela observação da Figura 7 podemos ver graficamente o comportamento do lambda adaptativo em função da média e desvio padrão de RGB.

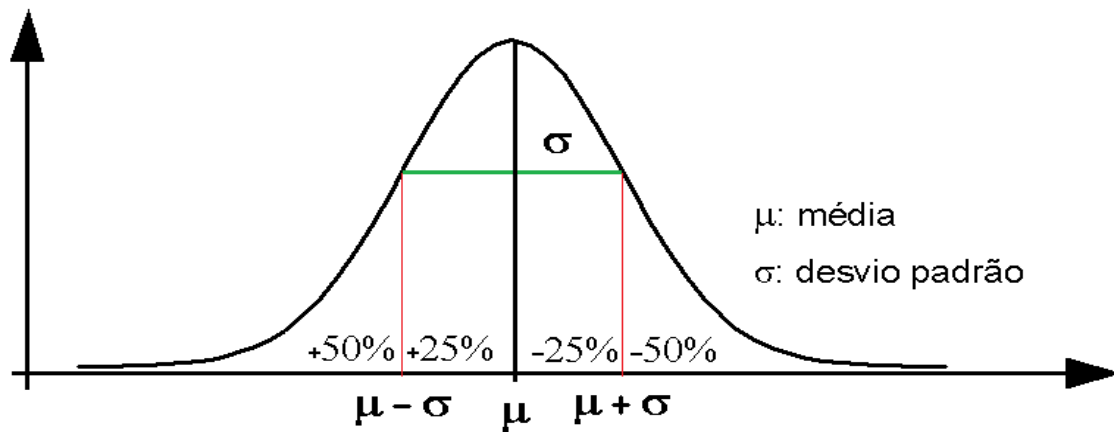


Figura 17 – Gráfico demonstrando o comportamento do lambda adaptativo de acordo com a média e desvio padrão.

Neste ponto está completa a adaptação de  $\lambda$  da equação (40), e esta terá embutido um sistema classificador  $\Lambda$  e fica como segue:

$$C(\lambda, \alpha, \beta, \mu, \sigma) = e^{-\left(\frac{|\sqrt[5]{\Delta(\alpha, \beta)}|}{\Lambda(\Delta(\alpha, \beta), \mu, \sigma, \lambda)}\right)} \quad (46)$$

O termo  $\Delta(\alpha, \beta)$  corresponde a equação (44).

### 4.2.3 Sistema de Equações Final

A versão final da equação (37) ou filtro de difusão anisotrópico tradicional, adicionando-se as modificações sugeridas na equação (42) e o classificador da equação



(45) embutido na equação (46), temos a versão final com todas as modificações como segue:

$$w(R)_i = \frac{1 - e^{-8*\tau*e}}{8} \frac{\left| \sqrt[5]{2 \sqrt{\left( (\alpha R - \beta R)^2 * G(\alpha R) \right) + \left( \alpha G - \beta G \right)^2 * G(\alpha G) + \left( \alpha B - \beta B \right)^2 * G(\alpha B)}} \right|}{\lambda \left\{ \begin{array}{l} r_1 = (\lambda * 0.50) \leftrightarrow \left( (\Delta(\alpha, \beta) > \mu) \wedge (\Delta(\alpha, \beta) > (\mu + \sigma)) \right) | \Delta(\alpha, \beta) = \sqrt{(\alpha R - \beta R)^2 + (\alpha G - \beta G)^2 + (\alpha B - \beta B)^2} \\ r_2 = (\lambda * 0.75) \leftrightarrow \left( (\Delta(\alpha, \beta) > \mu) \wedge (\Delta(\alpha, \beta) < (\mu + \sigma)) \right) | \Delta(\alpha, \beta) = \sqrt{(\alpha R - \beta R)^2 + (\alpha G - \beta G)^2 + (\alpha B - \beta B)^2} \\ r_3 = \lambda \leftrightarrow (\Delta(\alpha, \beta) = \mu) | \Delta(\alpha, \beta) = \sqrt{(\alpha R - \beta R)^2 + (\alpha G - \beta G)^2 + (\alpha B - \beta B)^2} \\ r_4 = (\lambda * 1.25) \leftrightarrow \left( (\Delta(\alpha, \beta) < \mu) \wedge (\Delta(\alpha, \beta) > (\mu - \sigma)) \right) | \Delta(\alpha, \beta) = \sqrt{(\alpha R - \beta R)^2 + (\alpha G - \beta G)^2 + (\alpha B - \beta B)^2} \\ r_5 = (\lambda * 1.50) \leftrightarrow \left( (\Delta(\alpha, \beta) < \mu) \wedge (\Delta(\alpha, \beta) < (\mu - \sigma)) \right) | \Delta(\alpha, \beta) = \sqrt{(\alpha R - \beta R)^2 + (\alpha G - \beta G)^2 + (\alpha B - \beta B)^2} \end{array} \right.} \quad (47)$$

Sendo  $R$  os parâmetros  $\lambda$ ,  $\tau$ ,  $\alpha$ ,  $\beta$ ,  $G(\alpha)$ ,  $\mu$  e  $\sigma$ , obtidos e calculados externamente ou durante o processamento do kernel  $W$ .

## ***Capítulo 5***

### ***Metodologia***

Este capítulo descreve brevemente o procedimento metodológico para o desenvolvimento da solução, testes da nova abordagem e validação. A metodologia está dividida em etapas que são: aplicação do filtro ADF convencional; cálculo dos gradientes de orientação; Aplicação dos gradientes; Adaptação do lambda, averiguação da funcionalidade com *ground truth* - GT e por fim, uma visão geral do processo é apresentada.

#### ***5.1 Conjunto de Imagens para Testes***

Para este trabalho, será adotado um conjunto de imagens bem estabelecido e conhecido dentro de ambientes de pesquisa em processamento de imagens. As imagens provêm de uma base de dados pública amplamente utilizada para testes de validação com o uso de GT. Os GTs são imagens de segmentação realizadas manualmente por pessoas, por esta razão considera-se como uma segmentação correta.

Este conjunto de imagens pertence ao *Berkeley Image Dataset* (<http://www.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/segbench/>), e são ricas em detalhes para testes em imagens outdoor. O conjunto de imagens oriundas deste local pode ser usado como base empírica para benchmarks de sistemas de segmentação e detecção de bordas (MARTIN, 2001). A base é pública e é composta de 300 imagens com os respectivos GTs. Para o teste aqui apresentado, foram selecionadas 40 imagens randomicamente da base de imagens de Berkeley. Abaixo segue Tabela 1 com as imagens selecionadas para o teste de validação corrente.



232038



216053



124084



310007



118035



97017



113016



113044



97033



19021



8023



28075



41004



71046



78019



86016



87065



123074



126039



140075



145053



145086



151087



156079



163085



169012



175043



176039



176035



183055



196015



219090



239096



351093



274007



271035



187071



167083



89072



100080

Tabela 1 – Imagens selecionadas randomicamente da base de dados de Berkeley para validação da metodologia proposta.

## 5.2 Aplicação dos filtros, Versão Tradicional e Modificada

De posse da imagem 124084 de Berkeley previamente selecionada, para exemplo, é aplicada então a filtragem com filtro de difusão anisotrópico na versão tradicional e modificada. Ambas as filtragens são realizadas com os mesmos parâmetros selecionados ad-hoc para cada imagem. Podemos ver na Figura 18: a) Imagem original; b) filtro tradicional; c) filtro modificado.



Figura 18 – Imagem 124084 de Berkeley, aplicação de ambas as versões do filtro com  $\lambda = 20$  e 90 iterações.

Para o teste de validação será adotado  $\lambda=39$  e 120 iterações do algoritmo, estes parâmetros são empíricos e selecionados de acordo com nossa experiência em processamento de imagens. As imagens, após serem filtradas servirão de entrada para o processo de segmentação posterior.

## 5.3 Segmentar as Imagens Filtradas

Depois de filtradas, as imagens passam por um processo de segmentação. Aqui adotamos a segmentação por crescimento de regiões pela equação de energia funcional de MS, da equação (15). A Figura 19 resume a aplicação da segmentação sobre as imagens b) e c) da Figura 18.



Figura 18-b) filtrada



Figura 18-c) filtrada

Figura 19 – Imagem original e imagens filtradas após segmentação com Mumford&Shah com  $\lambda = 62500$ .

Para o teste de validação o parâmetro da segmentação com Mumford&Shah(MS) será  $\lambda=2^{13}=8192$ . Esse é um parâmetro intermediário para segmentação, interessante

para os casos onde o filtro atua fazendo um excelente esmaecimento. O próximo passo será a aferição do resultado de segmentação segundo GT.

### ***5.4 Aplicação de Métricas de Qualidade em Imagens com Ground Truth***

Nesta etapa é aplicado sobre as imagens segmentadas em conjunto com as imagens de GT de *Berkeley* o algoritmo RAND e BGM. Estes têm a finalidade de gerar índices de dissimilaridade entre a segmentação computadorizada e a segmentação manual realizada pelos usuários do experimento de Berkeley. Na Figura 20 temos da esquerda para direita a imagem segmentada, imagem de GT e o índice dos dois algoritmos de dissimilaridade.

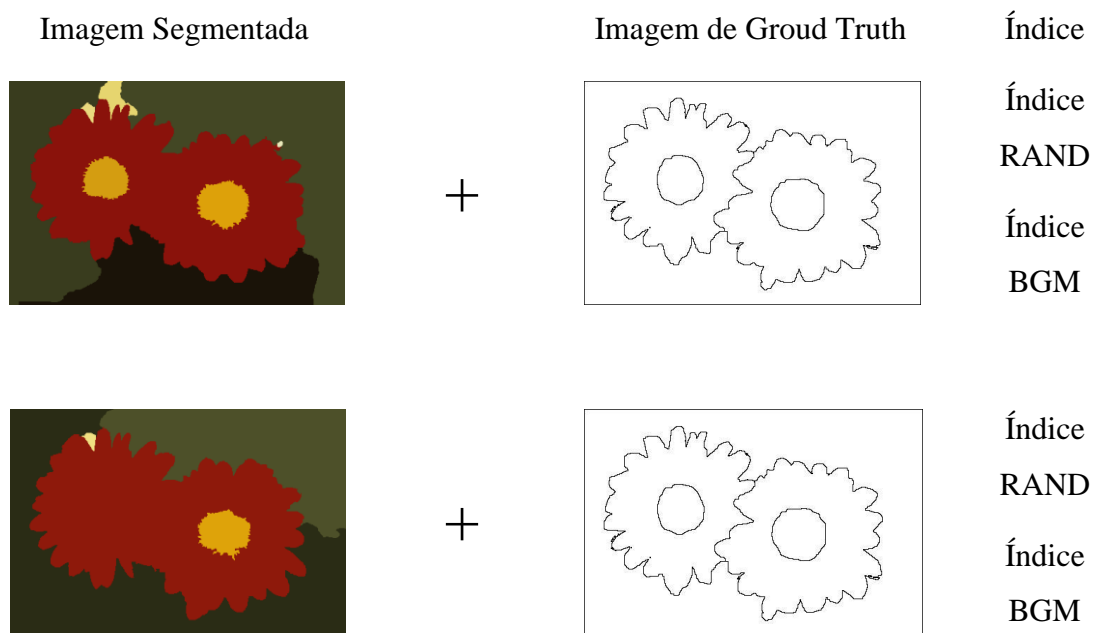


Figura 20 - Comparação da imagem segmentada com seu respectivo GT e índices RAND e BGM.

Quanto menor os índices mais similares são as segmentações em relação ao GT.

### ***5.5 Visão Geral do Processo de Teste***

Na Figura 21 podemos ter uma visão geral de todo o processo de validação, começando a partir da imagem original até a obtenção dos índices de similaridade.

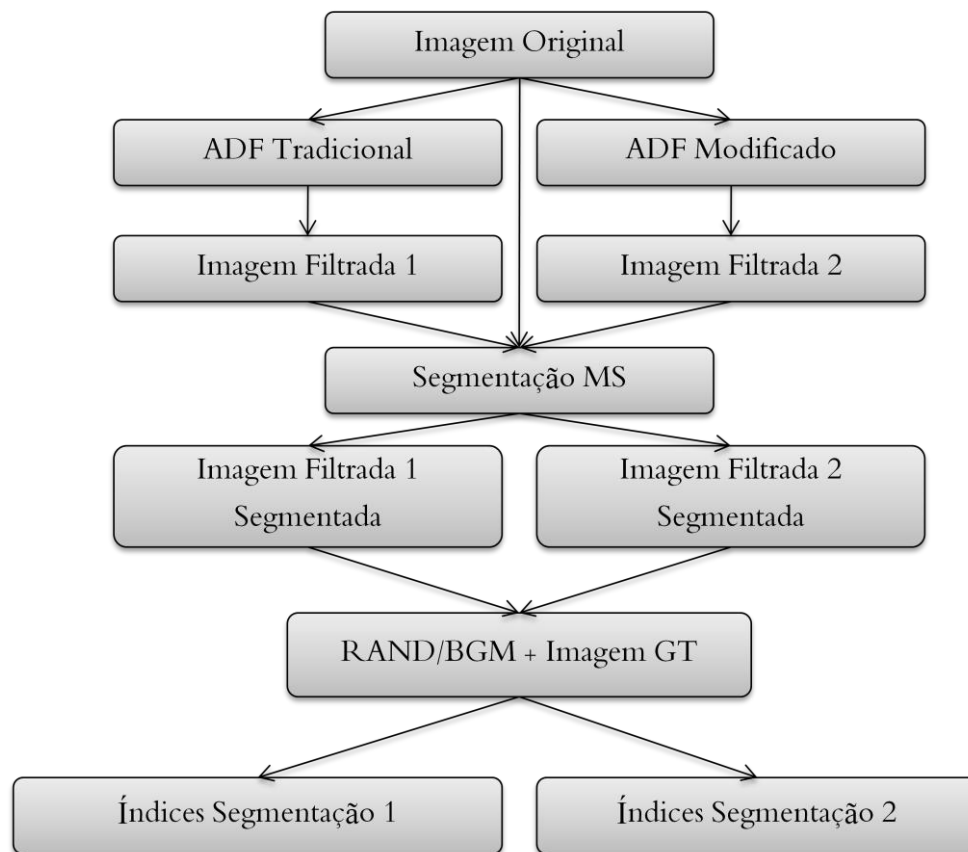


Figura 21 – Visão geral do processo de validação da metodologia.

O processo de validação começa com a imagem inicial filtrada com o filtro ADF tradicional e ADF modificado. Posteriormente as 2 imagens filtradas serão segmentadas com o método MS gerando 2 imagens segmentadas. Cada imagem segmentada será submetida ao teste com seus respectivos GTs. No final têm-se os índices de dissimilaridade do algoritmo RAND e BGM para cada imagem segmentada.

## Capítulo 6

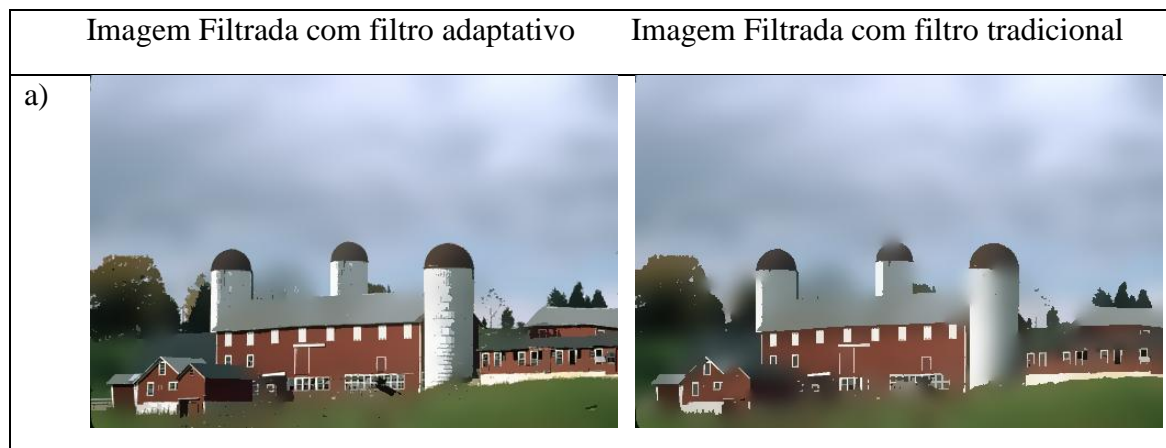
### Resultados

Neste capítulo são apresentados resultados obtidos no processamento das imagens pelo filtro de difusão anisotrópico tradicional e versão adaptada com as heurísticas embutidas no processo.

#### 6.1 Resultados da Aplicação do Filtro

Para aplicação do filtro de difusão anisotrópica foram realizadas filtragens de ambas as versões do filtro anisotrópico, a tradicional e a modificada. O parâmetro utilizado para todo o conjunto de validação foi  $\lambda=39$  e 120 iterações do filtro sobre a imagem. Em seguida um processo de segmentação por crescimento de região é aplicado com o parâmetro de  $\lambda=2^{13}=8192$ . Alguns aspectos visuais dos resultados são comentados a partir da Tabela 2.

Na Tabela 2 são apresentados 5 conjuntos de imagens filtradas com ambas as versões do filtro de difusão anisotrópico, rotuladas de (a) a (e). Estes casos podem ser considerados casos de sucesso da nova abordagem, preservando objetos claramente definidos onde na filtragem tradicional pode acontecer de objetos fundirem-se, mesmo quando estes são evidentemente diferentes. Logo abaixo de cada conjunto de imagem filtrada são comentados os principais aspectos considerados inconvenientes do filtro tradicional, os quais foram solucionados pelas heurísticas acima descritas.





As principais falhas desse conjunto ocorrem sobre os telhados das construções menores que não são preservadas. Em alguns casos desaparecem da imagem filtrada. Sobre a torre ao centro da imagem, o azul e tons marrons começam a se fundir no processo de difusão.



Nesta imagem observa-se o desaparecimento da floresta ao fundo e pequenos detalhes da construção, como portas e janelas, são abstraídos pelo filtro tradicional.



Neste caso, é evidente que muitos elementos foram fundidos e somente elementos altamente diferenciados puderam ser preservados durante o processo de difusão.



Para este conjunto, o gramado funde-se com a floresta ao fundo; a floresta funde-

se ao céu e mesmo elementos altamente dissimilares não são preservados, como a bolsa e chapéu. De maneira geral os objetos não são bem preservados pela versão tradicional do filtro.



Nestas figuras, vermelho do apoio funde-se com a figura do homem de preto, o rosto funde-se ao céu; as nuvens perdem muitos detalhes: suas bordas e a floresta começa a fundir-se com o céu ao lado da figura do cavalo. Fica impossível identificar o cavalo na filtragem tradicional.

Tabela 2 – Resultados da aplicação do filtro anisotrópico, casos de sucesso.

A Tabela 2 sumariza somente alguns caso onde muitas das características não podem ser preservadas por processos de esmaecimento tradicional e mesmo com filtro de difusão tradicional. Muitos outros casos de sucesso podem ser avaliados. Na Tabela 7 que está disponibilizada no Anexo C a partir da página 90.

Verificando-se os resultado também é possível apontar para alguns poucos casos onde as características da imagem dificultam a aplicação de qualquer processo de esmaecimento, mesmo ele sendo mais elaborado. Para estes casos um tratamento mais refinado faz-se necessário, como por exemplo, ajustes especializados do parâmetro de  $\lambda$  e do número de iterações.

Na Tabela 3 são apresentados 3 conjuntos de imagens, considerados casos onde a as imagens e segmentadas não apresentaram resultados significativos ao objetivo proposto, todos com um breve comentário. Todas as imagens da Tabela 3 apresentam baixa variação nos clusters de dados que representam os objetos como os animais e plantas bem como seu ambiente.









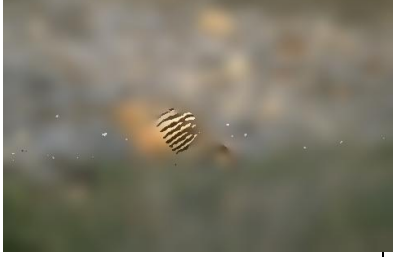
Imagem Original	Imagem Filtrada com filtro modificado	Imagem Filtrada com filtro tradicional
	a) 	
<p>Aqui mesmo o filtro adaptado funde as plantas rasteiras com o cacto, e outros elementos como o céu azul ao centro.</p>		
	b) 	
<p>Nestas imagens os membros do lagarto fundem-se às pedras, também dos anéis coloridos em seu corpo.</p>		
	e) 	
<p>Todos os elementos desta imagem foram fortemente difundidos entre ambiente e o pássaro.</p>		

Tabela 3 – Resultados da aplicação do filtro anisotrópico: casos de falha.

De maneira geral fica claro que as heurísticas do filtro de difusão anisotrópico são aplicáveis ao processo de filtragem para manter características importantes das imagens a serem processadas, porém em certos casos é necessário um ajuste fino de parâmetros para se alcançar bons resultados.

Para melhor esclarecimento sobre os resultados dos filtros, todos os testes de filtragem executadas podem ser visualizados no Anexo C.

## 6.2 Resultados da Validação e Segmentação

Após a execução da filtragem e em seguida à segmentação, o resultado deste processo, ou seja, as imagens segmentadas são conferidas juntamente com um conjunto de imagens de GT. O número de GTs varia de imagem para imagem. Para o conjunto de imagens do teste, o número de GTs variaram de 5 a 9 por imagem de entrada. A comparação é realizada com os algoritmos de comparação de cluster de dados RAND e BGM, os quais resultam em um índice de dissimilaridade na comparação da imagem segmentada com cada GT. Os resultados da validação podem ser acompanhados pela Figura 22 e Figura 23. Na Figura 22 são apresentados os índices RAND para o conjunto de validação, e na Figura 23 os índices para BGM.

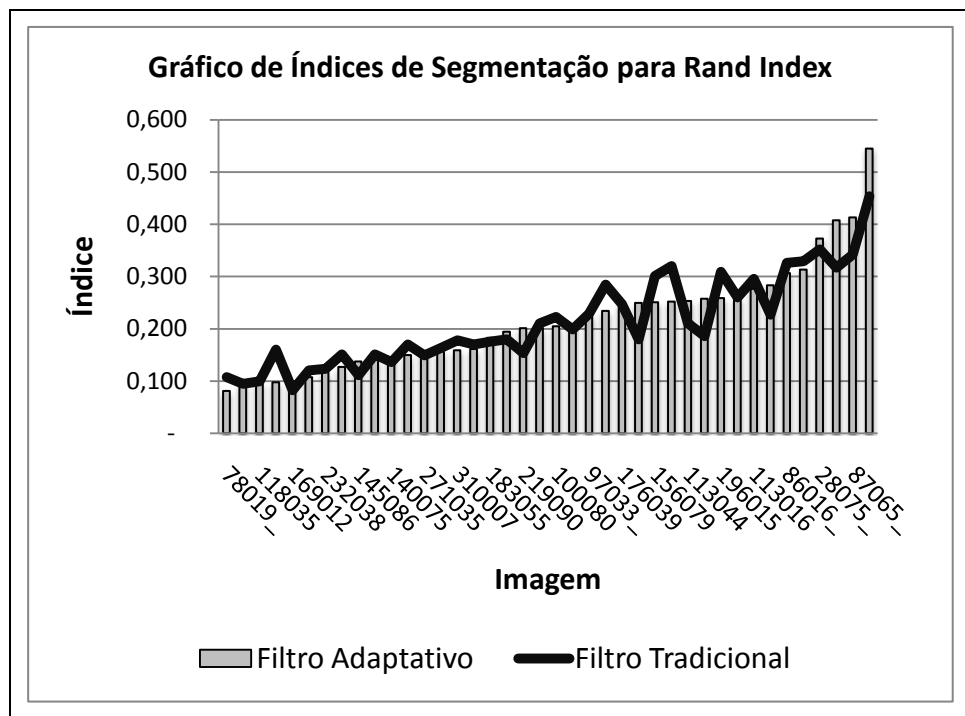


Figura 22 – Gráfico de índices RAND para o conjunto de validação.

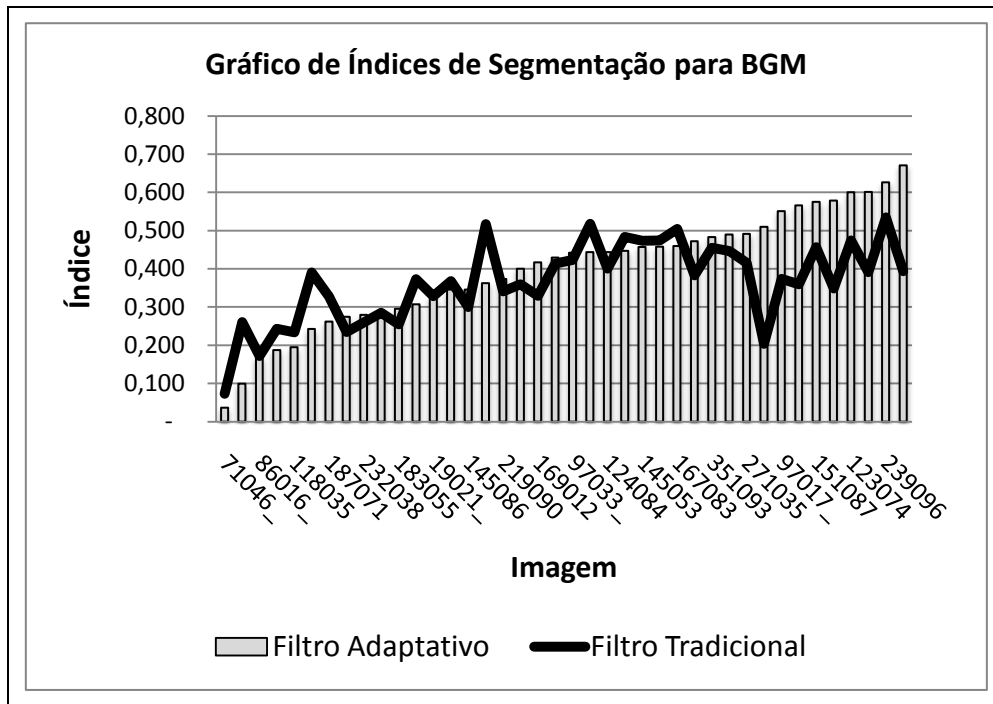
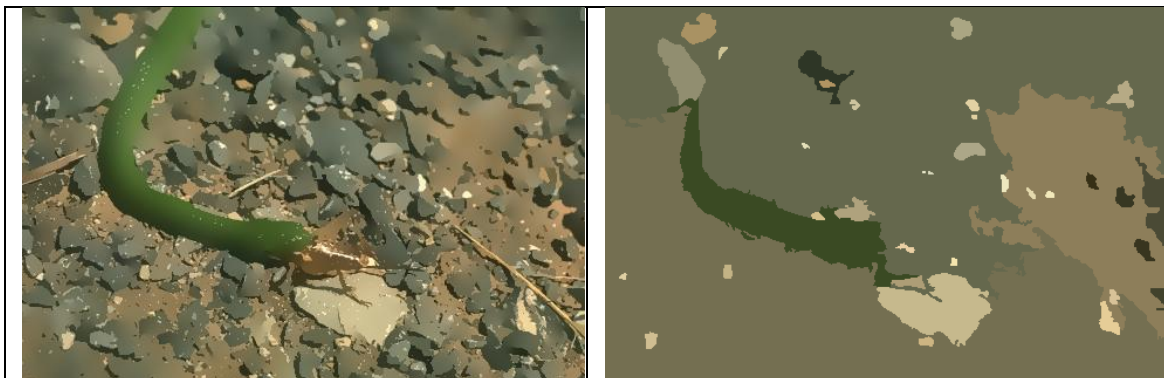


Figura 23 - Gráfico de índices BGM para o conjunto de validação.

Pode-se perceber pela leitura dos gráficos da Figura 22 e Figura 23, que existe grande flutuação entre os índices, entretanto os índices das imagens 175043 e 87065 respectivamente mostraram grande desequilíbrio na relação com as imagens segmentadas. Isto porque para ambos os filtros aplicados ao resultado de segmentação é inaplicável para qualquer tarefa de extração de dados ou reconhecimento, ou seja, apesar do índice indicar alguma correlação, este resultado de segmentação acaba sendo interessante.

A Tabela 4 apresenta 8 imagens, são elas: 2 imagens originais filtradas com 2 versões do filtro, e 4 imagens com resultados de segmentação não muito interessante, subsequentes às imagens filtradas.



a)175043 – filtrada com o filtro adaptativo	b)175043 – filtro adaptativo e segmentação
c)175043 – filtrada com o filtro tradicional	d)175043 – filtro tradicional e segmentação
e)87065 – filtrada com o filtro adaptativo	f)87065 – filtro adaptativo e segmentação
g)87065 – filtrada com o filtro tradicional	h)87065 – filtro tradicional e segmentação

Tabela 4 – Imagens com resultados menos relevantes de segmentação.

Na Tabela 4 as figuras (a), (c), (e) e (f) são resultados da aplicação dos filtros. Em (a) e (e), com apenas uma sub-segmentação, teríamos um resultado melhor do que foi apresentado na figuras (b) e (f). Para os casos (c) e (f) as imagens já foram deterioradas e não poderiam receber nenhum tratamento de segmentação. Portanto, os gráficos de índices apresentados anteriormente sofrem grande influência do parâmetro de segmentação adotado, e possuem comportamento muito semelhante para os índices das imagens que apresentam falhas na segmentação.

Em geral a média dos índices RAND e BGM estão expressos na Tabela 5:

	RAND	BGM
Filtro adaptativo	0,216	0,400
Filtro tradicional	0,212	0,364

Tabela 5 – Média dos índices RAND e BGM.

O que observa-se pelos índices médios é que o filtro tradicional aparentemente tem alguma vantagem sobre o filtro adaptativo. Porém, para o conjunto de 40 imagens filtradas, os índices RAND de 17 imagens filtradas com filtro adaptativo apresentaram índices superiores que nas imagens filtradas com filtro tradicional, o que vai na contramão do resultado da Tabela 5. Entretanto, para os índices BGM, 23 imagens filtradas com filtro adaptativo apresentaram índices maiores que nas imagens filtradas com filtro tradicional.

Para melhor esclarecimento sobre os resultados da validação, todos os índices e indicativos estatísticos, ou seja, os dados da bateria completa de testes podem ser visualizados no anexo B.

## *Capítulo 7*

### *Discussões e Conclusões*

Ao final do trabalho, é possível concluir que o filtro de difusão anisotrópico com as duas heurísticas definidas, ou seja, o uso de mapa de gradientes e a adaptação do parâmetro lambda foram suficientes para realizar de maneira eficiente a filtragem em imagens e preservar as bordas entre objetos, mesmo com muitas iterações e lambda alto.

Uma reflexão sobre os resultados dos índices de segmentação dá a entender que o filtro de difusão anisotrópico é inadequado para a fase de pré-processamento de imagens. No entanto, após inspeção visual dos resultados, os índices de segmentação são controversos com o resultado do filtro de difusão anisotrópico, em termos de filtragem, porque uma segmentação mais especializada é necessária para os casos que apresentam resultados inadequados. Também a título de teste de validação os parâmetros são fixos, para a segmentação e filtragem, o que comumente não acontece, sendo o parâmetro para cada imagem diferente na quase sua totalidade dos casos reais.

Entretanto, ao observar os resultados sob a ótica da qualidade da filtragem, fica evidente que as modificações do filtro o tornam extremamente atraentes como ferramenta definitiva para filtragem de imagens. Para uso em pré-segmentação, deve-se ter o cuidado para o ajuste do parâmetro de ambos algoritmos de filtragem e segmentação para os resultados esperados serem alcançados. Para os testes realizados para algumas imagens, a segmentação tornou os resultados não muito aplicáveis para qualquer processo de visão computacional.

Cabe ressaltar que a maneira utilizada para realizar a validação não foi muito interessante para se avaliar a qualidade da filtragem, porque foram usadas técnicas para avaliação de segmentações. Talvez uma outra maneira de se avaliar a qualidade de filtragem pudesse ter destacado melhor a qualidade da segmentação.

Os dois principais problemas apresentados na seção 4.2, foram sobrepostos com as táticas adotadas neste trabalho e vários outros trabalhos de pesquisa podem continuar a partir do que foi proposto aqui.



## ***7.1 Trabalhos Futuros***

Dentre os trabalhos futuros, as questões de pesquisa a seguir podem adicionar grandes funcionalidades e continuidade da pesquisa desenvolvida neste trabalho.

São elas:

1. Definir e encontrar melhores descritores de borda, pois o método adotado neste trabalho, apesar de superior aos comumente utilizados, tem tendência de detectar muito ruído nas imagens.
2. Utilizar outra métrica de distância em conjunto com outro espaço de representação de cores. A distância euclidiana como métrica de avaliação de cores não é exatamente adequada, porque não considera a incidência de luz sobre a informação de cor. Talvez a adoção de espaços mais especializados de cores resultem em excelentes resultados, por tratarem os canais de cores de maneira mais especializada.
3. Adicionar medidas de entropia, ou usar outra técnica para medir a variação dentro da região, sem considerar ruídos na imagem, desconsiderando baixas variações e falsas bordas.
4. O filtro de difusão adaptativo pode ser estendido para um processo de segmentação, ao se acoplar simples técnicas de segmentação durante a etapa de filtragem, como por exemplo, vizinhança por 8 conexos ou threshold com muitos níveis de seleção.

## ***Referências***

B.M. ter Haar Romeny (Ed). (1994). *Geometry-driven diffusion in computer vision*. Kluwer Dordrecht.

BARBETTA, P. A. (2006). *Estatística Aplicada às Ciências Sociais*. Florianópolis: Editora da UFSC.

BOTELHO, S. S. (2 de 3 de 2005). *Conceitos Básicos*. Acesso em 20 de 8 de 2008, disponível em Processamento Digital de Imagens: [http://www.ee.furg.br/~silviacb/DIP/Conceitos\\_Basicos.html](http://www.ee.furg.br/~silviacb/DIP/Conceitos_Basicos.html)

BROX, T. J. (2006). Nonlinear structure. *Image Vis. Comput* , 41–55.

BUENO, M. L. (12 de 2000). *Detecção de Bordas Através do Algoritmo de Canny*. Acesso em 20 de 7 de 2008, disponível em [wrg: www.inf.ufsc.br/~visao/2000/Bordas/index.html](http://www.inf.ufsc.br/~visao/2000/Bordas/index.html)

CANNY, J. F. (1986). A Computational Approach To Edge Detection. *IEEE Trans. Pattern Analysis and Machine Intelligence* , 679-714.

CASELLES, M. A.-F. (2007). Texture-Oriented Anisotropic Filtering and Geodesic Active Contours in Breast Tumor Ultrasound Segmentation. *J Math Imaging Vis* , 81–97.

COMUNELLO, E. (1999). Desenvolvimento de uma Metodologia Computacional para a Identificação, Contagem e Mensuração de Calcificações Relacionadas com Neurocisticercose em Imagens de Tomografia Computadorizada. Florianópolis, Santa Catarina, Brasil: UFSC.

COMUNELLO, E. (2004). The Cyclops Medical Image Interpretation System. *The Cyclops Medical Image Interpretation System*. Berlin, Alemanha: Verlag im Internet.

GONZALEZ, R. C., & WOODS, R. E. (2002). *Digital Image Processing, Second Edition*. New Jersey: Prentice Hall.

GONZALEZ, R. C., & WOODS, R. E. (2000). *Processamento de Imagens Digitais*. São Paulo: Edgard Blücher.

HUNT, R. W. (2004). *The Reproduction of Colour, 6th ed.* Chichester, UK: John Wiley & Sons.

JIANG et. al. (2006). Distance measures for image segmentation evaluation. *EURASIP Journal on Applied Signal Processing* , 1-10.

JIANG, X. e. (2006). Distance measures for image segmentation evaluation. *EURASIP Journal on Applied Signal Processing* , pp. 1–10.

MARTIN, D. F. (2001). A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. *In: Proc. 8th Int'l Conf. Computer Vision, vol. 2; 2001*, pp. 416–423.

MASCARENHAS, N. D., & VELASCO, F. R. (1989). *Processamento Digital de Imagens*. Buenos Aires: Kapelusz.

MATHWORKS. (2007). *Image Processing Toolbox*. Acesso em 20 de 8 de 2008, disponível em The MathWorks:  
<http://www.mathworks.com/access/helpdesk/help/toolbox/images/index.html?/access/helpdesk/help/toolbox/images/f8-15484.html&http://www.mathworks.com/cgi-bin/texis/webinator/search/>

MUMFORD, D., & SHAH, J. (1989). Optimal approximations by piecewise smooth functions and associated variational problems. *Communications On Pure & Applied Mathematics* , 557-684.

P. Perona, J. M. (30 de 11 de 1987). Scale space and edge detection using anisotropic diffusion. *Proc. IEEE Comp. Soc. Workshop on Computer Vision* , pp. 16-22.

PACIORNIK, S., & MAURICIO, M. H. (2004). Digital Imaging. In: G. V. Handbook, *Metallography and Microstructures* (pp. p. 368-402).

PARKER, J. R. (1997). *Algorithms for Image Processing and Computer Vision*. New York: John Wiley & Sons.

PAULO, P. e. (2004). *Segmentação por Crescimento de Regiões*. Acesso em 20 de 8 de 2008, disponível em Visão Computacional: <http://www.inf.ufsc.br/~visao/regiongrow.pdf>

PEDRINI, H., & SCHWARTZ, W. R. (2008). *Análise de Imagens Digitais*. São Paulo: Thomson Learning.

RABELO, A. B. (2002). *Análise da Utilização de Filtros Lineares e Não-Lineares na Recuperação de Imagens Degradadas*. Campinas.

RAND, W. M. (1971). Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*. Vol. 66. , 846-850.

SCHALKOFF, R. J. (1989). *Digital Image Processnig and Computer Vision*. John Wiley.

SEARA, D. M. (12 de 1998). *ALGORITMOS PARA DETECÇÃO DE BORDAS*. Acesso em 20 de 7 de 2008, disponível em Visao Computacional: <http://www.inf.ufsc.br/~visao/1998/seara/index.html>

TORRAS, C. (1992). *Computer Vision Theory and Industrial Applications*. New York: Springer – Verlag.

WANGENHEIM, A. V. (2007). *Visão Computacional*. Acesso em 10 de Dezembro de 2007, disponível em Seminário Introdução à Visão Computacional: <http://www.inf.ufsc.br/~visao/>

WEICKERT, J. (1998). *Anisotropic Diffusion in Image Processing*. Stuttgart: Teubner-Verlag.

WEICKERT, J. (2001). *Applications of nonlinear diffusion in image processing and computer vision*. Comenianae: Acta Math. Univ.

WIKIPEDIA. (2008). *Color*. Acesso em 15 de Janeiro de 2008, disponível em Wikipedia: <http://en.wikipedia.org/wiki/Color>

## *Anexos*

### **A. Artigos Publicados**

Foram publicados artigos 2 artigos sobre o filtro de difusão anisotrópica. Um no Congresso: The XXI Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAPI) - <http://www.gpec.ucdb.br/sibgrapi2008/> e outro no congresso: 2008 IEEE 11th International Conference on Computational Science and Engineering - <http://www.icmc.usp.br/~cse08/>. Respectivamente intitulados, Color Gradient Map-Oriented Anisotropic Diffusion Filtering e An Anisotropic Diffusion Filtering Implementation to Execute in Parallel Distributed Systems, de maneira que o tema parece ter grande relevância na area científica.

#### Artigo 1

### Color Gradient Map-Oriented Anisotropic Diffusion Filtering

#### Abstract

Anisotropic diffusion filtering is a well-established technique for image enhancement that works by means of diffusion functions. They are able to smooth images without destroying edge information. However, when many filtering iterations are applied, edges gradually fade away and are ultimately smoothed by the process. We propose the adoption of a color gradient map to guide the smoothing so that clearly-defined edges are preserved even after many iterations. Preliminary experiments show good results when compared with the regular anisotropic diffusion filter.

#### 1. Introduction

The anisotropic diffusion filter [2, 3] is a powerful image processing technique for noise removal. The filter works by performing smoothing on the image but at the same time preserving the boundaries between different regions. The process is controlled by a diffusion matrix that measures the color variation on the neighborhood of a hot spot. When the filter is applied over many iterations, the diffusion matrix slowly becomes adapted to the new local colors.

When the original image contains tenuous edges, the smoothing process will gradually erase all edge information as iterations are sequentially applied. In this work, we propose an improvement to the anisotropic diffusion filter that introduces a color gradient map, which behaves as static boundary

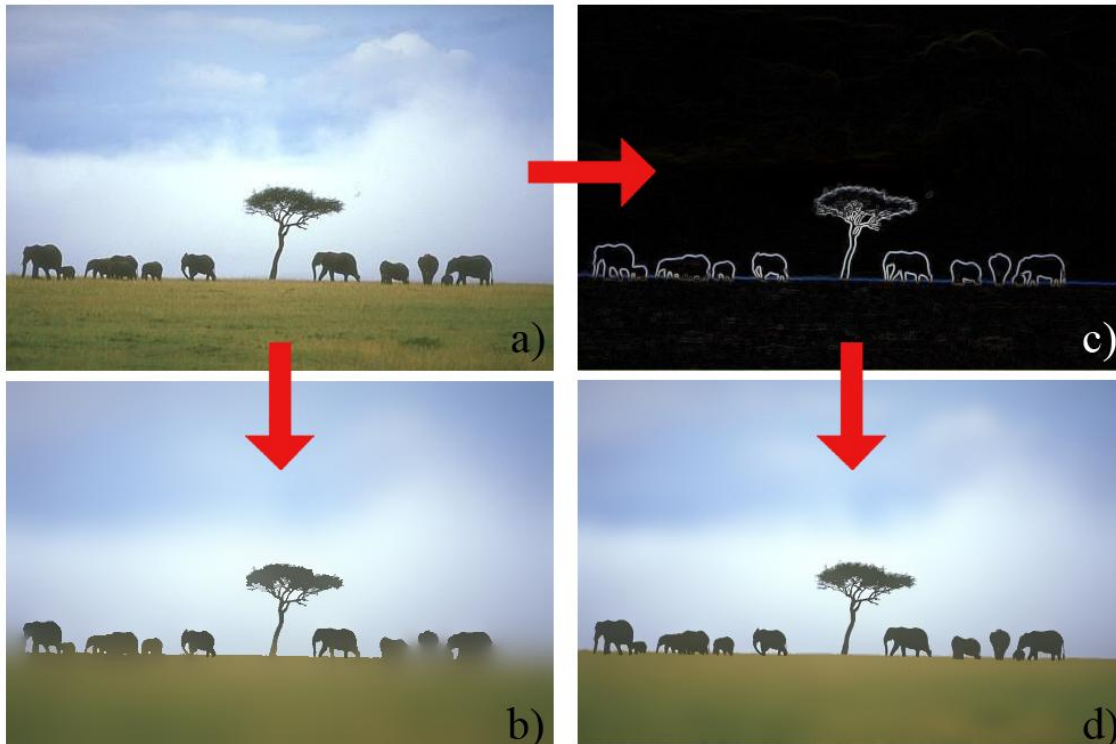
evidence, enhancing the response of the filter when applied several times to an image containing tenuous edges.

## 2. Orienting the Anisotropic Diffusion Filter

The anisotropic diffusion filter used in this paper has been extensively discussed [2], and can be regarded as a convolution technique with an adaptive matrix-valued kernel that performs a special smoothing on images: It inhibits the smoothing on edge pixels and stimulates it on internal regions. The basic diffusion equation [1] for an image  $I(x, y)$  with  $M$  channels and a signal initialized with  $u(x, y, 0) = I(x, y)$  is

$$\partial_t u_i = \text{div} \left( D \left( \sum_{k=1}^M \nabla u_k \nabla u_k^T \right) \nabla u_i \right) \quad (1)$$

where  $D$  is a matrix-valued function, and  $i = 1, \dots, M$  are the individual channels.



**Figure 1. Overview of traditional and gradient map-oriented filtering. a) is the original image; b) is the traditional filtering; c) is the gradient map generated from a); and d) is the filtering oriented by the gradient map shown in c).**

As the process of diffusion is carried on through several iterations, the edges of the original image gradually fade away, because the function  $D$  takes into account only the results from the previous iteration. After a certain amount of iterations, even edges that were initially well defined may become blurred. In order to keep well-defined edges present in the original image, we propose the introduction of

a static factor  $G_i$  in Equation 2 that depends on the original image only, thus avoiding the effects introduced by the repeated application of the diffusion filter. This static factor is calculated from a color gradient map produced from the original image and remains constant throughout all iterations. An overview of this process is shown in Figure 1, where both version of the filter is applied over 300 iterations.

In order to accommodate  $G_i$ , Equation 2 can be rewritten as

$$\partial_t u_i = \text{div} \left( D \left( \sum_{k=1}^M \nabla u_k \nabla u_k^T \right) \nabla u_i \right) \quad (1)$$

The gradient map  $G$  is calculated by a convolution operation using the following masks:

$$Gx = 1/4 \begin{bmatrix} -b & 0 & b \\ -a & 0 & a \\ -b & 0 & b \end{bmatrix} \text{ and } Gy = 1/4 \begin{bmatrix} -b & -a & -b \\ 0 & 0 & 0 \\ b & a & b \end{bmatrix}$$

where  $a = 2(\sqrt{2}-1)$  and  $b = (2-\sqrt{2})$ . The modulus of the vector  $(I_x, I_y)$  is then used as an estimation of the gradient for each channel.

Figure 2 shows an example of the color gradient map-oriented filter. While the internal regions were greatly smoothed, the distinction between them remains clear.

### 3. Conclusion and Discussions

The anisotropic diffusion filter is a powerful tool to improve the quality of images because of its sophisticated border-preserving smoothing. We have shown how this filter can be further improved by using a color gradient map that remains unchanged over iterations. This way, we ensure the preservation of edges that were well defined in the original image.

### References

- [1] T. Brox. From Pixels to Regions: Partial Differential Equations in Image Analysis. PhD thesis, Saarland University, 2005.
- [2] J. Weickert. Anisotropic Diffusion in Image Processing. Teubner-Verlag, 1998.
- [3] J. Weickert. Applications of nonlinear diffusion in image processing and computer vision. Acta Math. Univ. Comenianae, 70:33–50, 2001.



Figure 2. On the left, original images; on the right, images filtered with the approach we propose.

## Artigo 2

### **An Anisotropic Diffusion Filtering Implementation to Execute in Parallel Distributed Systems**

Antonio C. Sobieranski<sup>1</sup>, Leandro Coser<sup>1</sup>, M.A.R. Dantas<sup>2</sup>, Aldo v. Wangenheim<sup>1</sup>, Eros Comunello<sup>3</sup>

<sup>1</sup>LAPIX - Image Processing and  
Graphic

Computing Lab

The Cyclops Group -  
[www.cyclops.ufsc.br](http://www.cyclops.ufsc.br)

Federal University of Santa Catarina –  
Brazil

<sup>2</sup>LaPeSD – Distributed  
Systems Research Lab

[www.lapesd.inf.ufsc.br](http://www.lapesd.inf.ufsc.br)

Federal University of Santa  
Catarina – Brazil

[mario@inf.ufsc.br](mailto:mario@inf.ufsc.br)

<sup>3</sup>Graduate Program in  
Applied

Computer Science

Universidade do Vale do  
Itajaí – Brazil

[eros.com@univali.br](mailto:eros.com@univali.br)



{asobieranski, awangenh, coser}

@cyclops.ufsc.br

## Abstract

*In this paper, we present a parallelization of a filtering algorithm related to non-linear anisotropic diffusion, used to enhance the performance of an application in a parallel distributed system. The anisotropic diffusion is a well-established technique for image enhancement by means of diffusivity functions, which act as border attenuators. However, it requires a high computational cost when a large amount of data is used. The proposed implementation was parallelized considering both point-to-point and collective communications, adopting the MPI paradigm. Results from both approaches indicate that the proposed algorithm has reached interesting levels of performance (81% and 93% of efficiency, respectively) when compared to the execution of one process in a single computer node. In addition, our results indicate an enhancement of around 21% utilizing the collective communication strategy when compared to point-to-point communication.*

**Keywords:** *digital image processing, anisotropic diffusion filter, parallel distributed systems, message passing interface (MPI), point-to-point and collective communication.*

## 1. Introduction

Filtering techniques applied to ordinary images can achieve different levels of performance according to the computation complexity of the method. These techniques aim to reduce the noise from the image, thus improving the quality of the final result. An example is the convolution mask approach that filters an image in a linear and isotropic form, softening a specific point in accordance to its neighborhood. This technique has a major drawback: it does not consider boundaries and region limits; this is the isotropic characteristic of this approach. Other filters that use the linear method consider a matrix with information

related to region limits. However, these filters require a large number of iterations [1].

A research work presented by Joachim Weickert [2] proposes the use of matrix-valued diffusion tensors which simulates an anisotropic paradigm. This process is similar to physical phenomena, such as fluid dynamic, mass transportation and gas dispersion. These phenomena are inserted inside the isotropic and anisotropic approaches and can be represented through *partial differential equations* (PDEs). Several research works (e.g. [4, 15, 16, 17]), using the anisotropic filter, indicate that this filter reaches more precise results when compared to isotropic models, with fewer requirements for computational efforts. Weickert's work is employed in the pre-processing stage of many applications that require hundreds of segmentation steps (e.g., magnetic resonance imaging and computer tomography) for ordinary image processing tasks.

One important aspect related to the anisotropic filtering is the wide use of this approach, such as in the production of processed videos in real time. On the other hand, a well-known drawback of this type of filtering is the high consumption of computation resources.

The cluster and grid computing paradigms are being considered an attractive solution to tackle the problem in many organizations. Some works aimed at improving the performance of image processing filters. In [4], a high performance implementation of a diffusion anisotropic filter which targets 3D images is reported. The parallel approach used for volumetric data from medical images, in applications such as magnetic resonance imaging and computer tomography, represents an interesting strategy to achieve effective results. A framework developed for image search based on its contents is presented in [5]. The proposal employs ordinary filtering techniques based on space domain, as suggested in [6]. The strategy considers a distributed processing which works concurrently on the image. A grid computing approach designed and implemented for digital image processing (*DIP*) is shown in [7].

In this article, a parallelization of an anisotropic diffusion filter algorithm to execute in a distributed parallel configuration is presented. The main target of this parallelization are bi-dimensional image applications, such as medical images, outdoors scenes, noise reduction and aerospace applications. In addition,

we show a comparison related to point-to-point and collective communications using the message passing interface (*MPI*) for the proposal.

The experimental environment was characterized by the heterogeneity of the machines (i.e., processors and memory features) and also the versions of operating systems.

The paper is organized as follows. Section 2 describes theoretical aspects related to image filtering and distributed parallelization. The parallelization details of the proposal are pointed out at Section 3. The experiment descriptions are discussed in Section 4, where we also present the methodology used to obtain our results. Results from the anisotropic filtering parallelization are shown in Section 5. Finally, in Section 6 we present conclusions and future work related to this research.

## 2. Image Filtering and Distributed Parallelization

*In this section we present two relevant aspects for this research work. The first element is how anisotropic diffusion filtering is characterized. The latter tackles important elements for a parallel implementation of the algorithm.*

### 2.1. Theoretical Aspects of Anisotropic Diffusion Filtering

As it was mentioned before, isotropic filters and other linear operations do not consider aspects such as borders and region boundaries. The main reason for this limitation lies in the scalar space used by these strategies to analyze signals and multi-scale images. As a result, images have a poor resolution, because they represent a Gaussian convolution of the original image [2].

Enhancements to these PDE approaches were initially suggested by Perona and Malik in 1987 [3]. This work proposes a non-linear diffusion filter concept, i.e., an anisotropic strategy, which produces a reduction in the diffusion coefficient (scalar value) until they are encapsulated by a border [2]. This filter is presented in [3] as:

$$g(s) = \frac{1}{1 + s^2/\lambda}, \text{ to } \lambda > 0. \quad (1)$$

It guarantees that thick borders will be clearer when the diffusion filter is used. Perona and Malik also propose a discrete model of the diffusion filter:



$$(2)$$

$\mathbf{I}(\mathbf{s}, t)$  is the discrete image in spatial and temporal terms;  $\mathbf{s}$  is the pixel position in a discrete bi-dimensional grid,  $t$  shows the discrete time step ( $t \geq 0$ )  $\lambda$  is a constant which determines the speed of diffusion and  $|\eta_s|$  represents a set of spatial neighbors from an  $\mathbf{s}$  pixel.

In the Perona-Malik formalization, there is the so-called edge stopping function,  $\mathbf{g}$ , which controls the diffusion intensity in accordance to the point gradient that should suffer a specific diffusion. The edge function stopping has a  $\lambda$  parameter which, in conjunction with the gradient, indicates whether diffusion is strong or weak. In [3] there are suggestions for these functions:

$$g_1(x) = \frac{1}{1 + \frac{x^2}{2\sigma^2}} \quad (3)$$

$$g_2(x) = e^{-\left[ \frac{x}{p_2} \right]^2} \quad (4)$$

Weickert [2, 14] argues that this technique can be expanded to an anisotropic process utilizing adaptive diffusion tensors. The Weickert proposal is similar to the Perona and Malik diffusion model. In this second proposal, a discrete solution is found by iterative processes. The discrete form from the anisotropic process is provided by the following discrete model:



$$(5)$$

The formula presented above can be understood as: the  $\mathbf{k}$  superior index is the solution at the time  $\mathbf{k}\tau$ , with  $\tau$  indicating the stopping time; the set  $\mathbf{N}(\mathbf{i}, \mathbf{j})$  contains the neighborhood of the pixel  $(\mathbf{i}, \mathbf{j})$ ; the  $|\mathbf{m}, \mathbf{n} - (\mathbf{i}, \mathbf{j})|$  expression indicates the linear distance between pixels  $(\mathbf{i}, \mathbf{j})$  and  $(\mathbf{m}, \mathbf{n})$  (generally, the linear norm of the vector);  $(\mathbf{g}(\mathbf{m}, \mathbf{n}))\mathbf{k}$  provides an approximation of the local diffusion coefficient  $((\mathbf{i} + \mathbf{m})/2, (\mathbf{j} + \mathbf{n})/2)$  at the time

$\mathbf{k}\tau$ . This means that the diffusion rate that belongs the connection between the gradient  $(\mathbf{i},\mathbf{j})$  and  $(\mathbf{m},\mathbf{n})$ , where the gradient magnitude  $\mathbf{g}$  can be estimated from discrete data using a mask.

Weickert considers the Perona-Malik model as an isotropic model, since it utilizes a scalar value for diffusion, and not a diffusion tensor (represented by a matrix-valued). The Perona-Malik experiment is clearly unpredictable, because the boundaries only stay stable after a great number of iterations. However, the border detection employing by this model is more accurately than the linear Canny approach when used as border detector.

This can be still verified in cases where the maximum coefficients are not used. The main reason for that is that both the interactive diffusion and the border detection are within a single process, in contrast to several independent processes. The interior of a segment is always preserved, while the diffusion borders are inhibited.

The anisotropic diffusion filter proposed by Weickert has provided very useful results. However, due to its computational complexity, it works on the image with a predefined number of iterations, determined by the  $\lambda$  scalar parameter, which controls how strong the diffusivity must be.

## 2.2. Message Passing Interface Libraries

Multicomputer configurations are well known as loosely coupled architectures that more suitable to applications developed using the message passing paradigm.

The work presented in [12] shows that even mobile devices can be employed to access large multicomputers for the execution of processes, such as a complex visual applications. On the other hand, PVM (*Parallel Virtual Machine*) and MPI (*Message Passing Interface*) are the most widely accepted message passing libraries in the engineering and scientific communities. Both PVM and MPI have interesting features to implement an application to be executed in [13]. Environment transparencies, scalability, portability and predefined communication calls are some examples of features available for these two message passing libraries.

In this paper, we consider the usage of the MPI because it is considered a standard library. In our research work we considered the use of the following communication paradigms [9]:

- Point-to-Point: communications occur only across two processes in a specific call;
- Collective: a communication can occur among a number of processes at the same

time, using a collective call primitive. As an example a process can gather information from several processes employing a single call.

Messages can also be classified as blocked and non-blocked [8]. The first call waits for communication to finish, while the latter does not.

In this research work, in order to reach an effective high performance data communication inside the application, we have considered the use of both communication strategies, i.e., the collective and point-to-point. In addition, we considered the MPICH [9] and LAM [10] implementations of the standard MPI. These implementations have some differences as presented in [9] and [10] and we employed them where they were more suitable.

## 3. The Proposal Anisotropic Diffusion Filter Parallelization

In this section, first we present some characteristics related to the anisotropic diffusion filtering parallelization. In the second part of the section, we describe the distributed system configuration, in terms of message passing software library and hardware.

### 3.1. Parallelization characteristics

As it was mentioned before, the anisotropic diffusion filtering is a powerful tool to improve the quality of any image. The quality process is characterized by the image smoothing and the sophisticated border preservation. However, to achieve this image smoothing an iterative process that has a high computational cost is necessary.

Independently from the process execution environment, it is possible to consider some aspects to the parallelization of the anisotropic diffusion filtering. The following aspects were observed in the implementation of this work:

- A performance enhancement is achieved by dividing the image in small parts so that each part can be processed by a computational node from a distributed system configuration. Figure 1 illustrates an input from an image that is divided into nine pieces;
- Network latency and application granularity. As demonstrated in [14], this is a relevant aspect to be considered when implementing distributed parallel applications. Therefore, we have implemented two versions of the parallel code. The first code employs point-

to-point communication calls. The other version uses collective communications. This approach could indicate which solution would be better suited to execute our application in the distributed parallel configuration;



**Figure 1. An input figure divided in nine processes [11]**

- In the point-to-point communication version, the image is divided into pieces  $h/p$  pixels high ( $h$  is the image height and  $p$  the number of proceses). A load balancing approach was considered when the image division creates an odd number of parts. The master-slave approach was used as the parallel programming paradigm. When each slave finishes their task, a process collects all pieces to build up the final image;
- The collective communication has a similar programming paradigm, however the scatter and gather primitive calls were utilized. In other words, the image division operation is realized transparently for the application programmer, who only needs to define the image block size (the MPI primitives are like a black box, since division occurs internally). Similar to the point-to-point communication parallel approach, when the division reaches an odd number of parts, the master node takes care of those parts that were not sent to any slave.

**Table 1. Hardware and software characteristics of the experimental configuration**

Processor	Memory	NIC	OS
Intel(R) Pentium(R) 4 CPU 2.40GHz GenuineIntel	0,7 GB	100T	Gentoo 2007.1 (i686), kernel 2.6.22

### 3.2. Distributed System Configuration

The LAM [10], an MPI implementation (version 7.1.4). was used to parallelize the distributed message passing communication. Table 1 shows the eight nodes used in our configuration and illustrates the general characteristics of the machines from the loosely coupled multicomputer configuration. This environment is characterized by the heterogeneity in term of hardware and operating systems versions. In terms of hardware, it is possible to notice different processors (e.g., Pentium and AMD) and network interfaces (e.g., 100T and 1000T). On the other hand, 32- and 64-bit software systems were available. The Linux kernel 2.6 can be considered as the only homogenous characteristic of the configuration.

### 4. Experiments Description

Our experiments were characterized by the execution of two parallel versions of the anisotropic diffusion filtering. The first parallel code was implemented using point-to-point communications and followed the aspects mentioned in section 3.1. The other version was implemented using collective communications. In addition, the following points were observed:

- Image size in terms of width x height;
- Number of processors available in the configuration;
- Type of the original communication that exists in the anisotropic diffusion filtering algorithm (i.e. point-to-point or collective).

All experiments considered six types of images with the following dimensions: 320x240, 800x600, 1024x768, 1280x1024, 1600x1200 and 3888x2592. Each image was submitted to the anisotropic diffusion filtering process using 30 iterative operations and a  $\lambda$  neighborhood factor of 15. Because the algorithm's execution time is proportional to the number of iterations, the use of a fixed pattern is sufficient and stable to draw comparison results between the parallel executions.

Intel(R) Pentium(R) 4 CPU 2.40GHz GenuineIntel	1,5 GB	100T	Ubuntu (i686) kernel 2.6.20
AMD Athlon(tm) 64 Processor 3200+ AuthenticAMD	1 GB	1000T	Slackware (i686) kernel 2.6.13
AMD Athlon(tm) 64 Processor 3500+ AuthenticAMD	2 GB	1000T	Gentoo (i686) Kernel 2.6.20
AMD Athlon(tm) 64 X2 Dual Core Processor 4400+ AuthenticAMD	2 GB	1000T	Gentoo (i686) Kernel 2.6.22
AMD Athlon(tm) 64 X2 Dual Core Processor 4400+ AuthenticAMD	2 GB	1000T	Gentoo (i686) Kernel 2.6.22
AMD Athlon(tm) 64 X2 Dual Core Processor 4400+ AuthenticAMD	2 GB	1000T	Gentoo (i686) Kernel 2.6.22
AMD Athlon(tm) 64 Processor 3000+ AuthenticAMD	1 GB	100T	Gentoo (x86_64) Kernel 2.6.22
AMD Athlon(tm) 64 Processor 3000+ AuthenticAMD	1 GB	100T	Gentoo (x86_64) kernel 2.6.22

The efficiency is characterized in this paper as the total execution time (including time to send and receive data) to finalize the anisotropic diffusion filtering process, observing the previous described parameters. The execution time has two main components: the time spent to execute the diffusion process and the time related to the data communication among nodes. In addition, we computed the required time to divide parts of the image in the point-to-point implementation. This procedure was necessary to have a fair comparison with the collective implementation, where the scatter and gather primitive calls are transparent for the application programmer. The time required to upload and download images from (or to) the disk to the main memory was not considered.

In addition, program executions adopted an exponential image division in terms of number of nodes. As a result, we have seven executions, e.g., from 20 to 26, where 20 represents a sequential execution in a single node.

The final number of experiments was eighty four tests, from six images, two diffusion filtering implementations and seven executions related to the number of nodes.

## 5. Empirical Results

In our experiments, we have executed the two anisotropic diffusion filtering implementations considering different image sizes and different number of processing nodes, as we illustrate in this section.

The two parallel implementations have presented an interesting degree of efficiency when compared to the sequential version. This aspect indicates that in the end of the day a clearer image was obtained in a shorter time.

Table 2 shows different sizes of images, number of processes and their respective executions for point-to-point and collective communications. This table shows the high performance of the parallel implementation when compared to the sequential algorithm in both implementations. It is interesting to notice that the scatter and gather primitive calls have a high latency for two communication nodes. However, when one more node is aggregated to the configuration there is a sharp drop in the execution time. This pattern continues to occur when more computational nodes are added to the configuration. Similar to the research presented in [14], we have also noticed the high latency related to the interconnection network. Because of the latency, a plateau level is observed in terms of execution time.

**Table 2. Average time for different image size and number of processes**

**Point-to-Point Communication**

Image Size	Number of processors						
	1	2	4	8	16	32	64
380x240	9.480 s	4.948 s	5.074 s	2.321 s	2.324 s	1.764 s	1.870 s
800x600	47.442 s	24.913 s	23.261 s	11.533 s	11.550 s	8.545 s	9.250 s
1024x768	82.156 s	41.341 s	43.678 s	19.689 s	21.431 s	16.210 s	16.240 s
1280x1024	132.420 s	66.021 s	66.916 s	31.531 s	31.823 s	23.964 s	25.270 s
1600x1200	195.340 s	98.056 s	93.180 s	47.443 s	46.161 s	34.911 s	37.680 s
3888x2592	992.115 s	526.217 s	480.525 s	255.232 s	243.852 s	179.747 s	191.910 s

**Collective Communication**

Image Size	Number of processors						
	1	2	4	8	16	32	64
380x240	9.480 s	5.485 s	1.490 s	0.630 s	0.391 s	0.309 s	0.311 s
800x600	47.442 s	27.132 s	6.822 s	3.692 s	3.689 s	1.740 s	1.747 s
1024x768	82.156 s	36.000 s	13.110 s	7.615 s	7.700 s	5.672 s	3.060 s
1280x1024	132.420 s	61.000 s	20.473 s	10.991 s	11.160 s	7.893 s	8.614 s
1600x1200	195.340 s	80.000 s	29.552 s	17.366 s	15.919 s	12.048 s	13.517 s
3888x2592	992.115 s	440.000 s	140.262 s	79.029 s	79.830 s	60.081 s	63.547 s

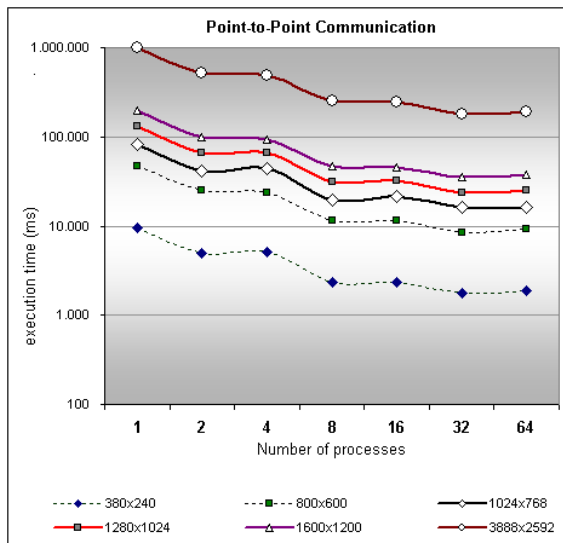
Figures 2 and 3 illustrates in logarithmic scale the performance of the point-to-point and collective implementation, respectively, considering different image sizes and number of processors.

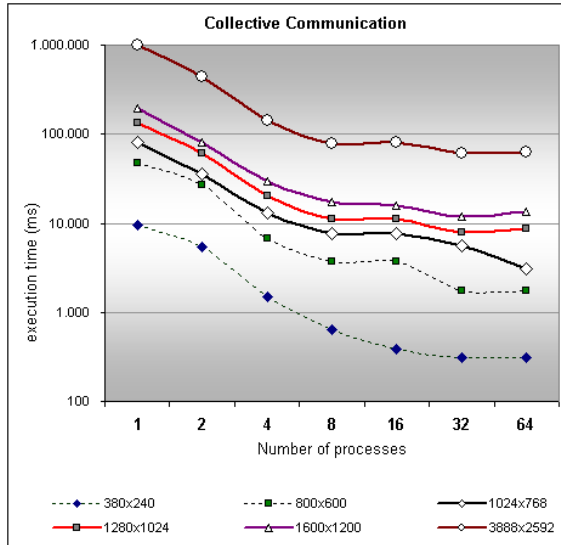
Observing Figure 2, it is possible to notice a low variation in the execution time between two and four nodes (and eight and sixteen). There is also a significant increase from four to eight nodes (and from sixteen to thirty-two). On the other hand, from thirty-two to sixty-four there is

a plateau in terms of execution time. This behavior occurs because of the high latency from the interconnection network.

A similar comment can be drawn in relation to Figure 3, where the collective communication implementation is presented. The two main differences in this figure in comparison to the Figure 2 are the high latency for two nodes and low execution time to execute the algorithm when three (or more) nodes are considered.

**Figure 2. Image sizes and number of processes for the point-to-point implementation.**

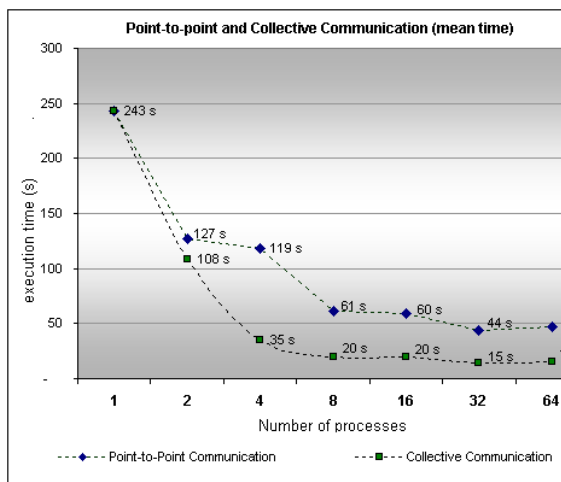




**Figure 3. Image sizes and number of processes for the point-to-point implementation.**

Figure 4 shows a comparison between the point-to-point and collective implementations. This chart illustrates all executions. This figure shows an average of 21.5 % decrease in execution time.

Considering the sequential execution, represented by one node, our implementations have reached a performance gain of 81.9% in the point-to-point implementation and 93.8% in the collective. These two numbers represent the best execution times in both implementations.



**Figure 4. Image sizes and number of processes for the**

## point-to-point and collective communications implementation.

### 6. Conclusions and Future Works

In this paper we have presented the utilization of anisotropic diffusion filtering to enhance the execution time of digital image processing. Our research work was characterized by the implementation of two communication models of the anisotropic diffusion filter. These algorithms adopted the message passing programming approach as the processing model to be executed in a distributed loosely coupled environment. The first implementation considered point-to-point communication primitive calls and the second collective communications. These two main goals were to observe how difficult these two approaches would be to implement and how they perform in term of execution time. The implementation issue was related to image division among processing nodes. On the other hand, the execution time considered the size and decrease time to execute these applications.

Our experiments indicate that both approaches have interesting performance results in terms of efficiency. The point-to-point algorithm has achieved a performance gain of 81.9% and the collective implementation 93.8% in relation to the sequential execution. In both cases 64 processes were used. In addition, the collective communication has presented an enhanced performance in comparison to the point-to-point approach and it was more easily implemented using pre-existent collective calls (e.g. scatter and gather).

As a future work, we are planning to utilize different types of interconnection networks to draw some pictures related to the scalability of the node configuration. In addition, we are also planning to provide a framework for the execution of the present application on a mobile device. Another feature is an implementation of two-dimensional anisotropic diffusion filters on GPUs, and studies of performance between multicomputers and the stream programming model.

### References

- [1] B.M. ter Haar Romeny (Ed), Geometry- driven diffusion in computer vision, Kluwer Dordrecht, (1994).

- [2] Weickert, J, Anisotropic Diffusion in Image Processing, Teubner, Stuttgart, (1998).
- [3] P. Perona, J. Malik, Scale space and edge detection using anisotropic diffusion, Proc. IEEE Comp. Soc. Workshop on Computer Vision (Miami Beach, Nov. 30 - Dec. 2, 1987), IEEE Computer Society Press, Washington, 16 - 22, (1987)
- [4] Andrés Bruhn, Tobias Jakob, Markus Fischer, Timo Kohlberger, Joachim Weickert, Ulrich Brning and Christoph Schnrr. High performance cluster computing with 3-D nonlinear diffusion filters. Real-Time Imaging 10 (2004) 41-51.
- [5] Rahman Tashakkori, Steven H. Heffner, High-Performance Image Content-Based Search. IPCV - International Conference on Image Processing, Computer Vision, & Pattern Recognition (2006) 145-150
- [6] Gonzalez, Digital Image Processing, 2<sup>o</sup> Edition, (1993).
- [7] Zhanfeng Shena, Jiancheng Luoa, Chenghu Zhoua, Guangyu Huangc, Weifeng Mad, Dongping Minga. System design and implementation of digital-image processing using computational grids. Computers & Geosciences 31 (2005) 619-630.
- [8] I. Foster. Designing and Building Parallel Programs. (1995).
- [9] W. Gropp, E. Lusk, N. Doss, A. Skjellum. "A Highperformance, Portable Implementation of the MPI Message Passing Interface Standard". Parallel Computing, 22(6):789-828, (1996).
- [10] Greg Burns, Raja Daoud, James Vaigl. "LAM: An Open Cluster Environment for MPI". Proceedings of Supercomputing Symposium, (1994).
- [11] The Berkeley Segmentation Dataset and Benchmark, <http://www.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/segbench/>
- [12] Anubis G. M. Rossetto, Vinicius C. M. Borges, A. P. C. Silva, M. A. R. Dantas: SuMMIT - A framework for coordinating applications execution in mobile grid environments. GRID (2007): 129-136
- [13] M. A. R. Dantas, Ed Zaluska: Experiences with porting an engineering application onto workstation clusters using PVM and MPI. EUROSIM (1996): 229-236
- [14] Luiz C. Pinto, Rodrigo P. Mendonça, M.A.R. Dantas: Impact of Interconnection Networks and Application Granularity to Compound Cluster Environments, Paper submitted to the ISCC (2008).
- [15] Weickert, J.: Applications of nonlinear diffusion in image processing and computer vision. Acta Mathematica Universitatis Comenianae, vol. 70, No. 1, (2001): 33-50.
- [16] Chung, D. H.; Sapiro, G.: Segmenting skin lesions with partial differential equations based image processing algorithm. IEEE Transactions on Medical Imaging, vol. 19, issue 7, (2000): 763-767
- [17] Demirkaya, O.: Anisotropic diffusion filtering of PET attenuation data to improve emission images. Physics in Medicine and Biology, vol. 47, (2002): 271-278.

## B. Tabela de Dados

Abaixo a Tabela 6 completa de dados gerados pelo processo de validação, todos os índices de cada GT é comparado com cada segmentação. Também está calculado a média de índices para cada segmentação, número de segmentos da imagem da imagem segmentada (N Segs), número de segmentos de cada GT (N Segs GT) e o número de GTs comparados com o arquivo de entrada (N GTs).

Nome da Imagem De segmentação	Rand Index	BGM	N Segs GT	N Segs	N GTs
100080_IT_120_LB_39_FLAG_3_ms	0,19	0,24	7,00	10	5
	0,17	0,17	35,00		
	0,35	0,47	4,00		
	0,15	0,15	27,00		
	0,17	0,19	12,00		
<b>Médias</b>	<b>0,20</b>	<b>0,24</b>	<b>17,00</b>		
<b>SD</b>	<b>0,08</b>	<b>0,13</b>	<b>13,40</b>		
100080_IT_120_LB_39_FLAG_4_ms	0,22	0,39	7,00	7	5
	0,16	0,30	35,00		
	0,38	0,57	4,00		



	0,15	0,29	27,00		
	0,21	0,39	12,00		
<b>Médias</b>	<b>0,22</b>	<b>0,39</b>	<b>17,00</b>		
<b>SD</b>	<b>0,09</b>	<b>0,11</b>	<b>13,40</b>		
113016_IT_120_LB_39_FLAG_3_ms	0,48	0,60	3,00	40	6
	0,15	0,33	31,00		
	0,49	0,60	7,00		
	0,18	0,38	11,00		
	0,18	0,37	24,00		
	0,18	0,38	13,00		
<b>Médias</b>	<b>0,28</b>	<b>0,44</b>	<b>14,83</b>		
<b>SD</b>	<b>0,16</b>	<b>0,12</b>	<b>10,63</b>		
113016_IT_120_LB_39_FLAG_4_ms	0,50	0,66	3,00	24	6
	0,15	0,37	31,00		
	0,50	0,66	7,00		
	0,21	0,47	11,00		
	0,21	0,46	24,00		
	0,21	0,47	13,00		
<b>Médias</b>	<b>0,30</b>	<b>0,52</b>	<b>14,83</b>		
<b>SD</b>	<b>0,16</b>	<b>0,12</b>	<b>10,63</b>		
113044_IT_120_LB_39_FLAG_3_ms	0,19	0,58	10,00	44	6
	0,19	0,58	15,00		
	0,39	0,66	5,00		
	0,19	0,58	9,00		
	0,16	0,54	17,00		
	0,40	0,67	6,00		
<b>Médias</b>	<b>0,25</b>	<b>0,60</b>	<b>10,33</b>		
<b>SD</b>	<b>0,11</b>	<b>0,05</b>	<b>4,80</b>		
113044_IT_120_LB_39_FLAG_4_ms	0,14	0,35	10,00	17	6
	0,14	0,35	15,00		
	0,35	0,47	5,00		
	0,14	0,36	9,00		
	0,13	0,33	17,00		
	0,36	0,48	6,00		
<b>Médias</b>	<b>0,21</b>	<b>0,39</b>	<b>10,33</b>		
<b>SD</b>	<b>0,11</b>	<b>0,07</b>	<b>4,80</b>		
118035_IT_120_LB_39_FLAG_3_ms	0,09	0,20	14,00	31	5
	0,10	0,21	19,00		
	0,07	0,08	47,00		
	0,16	0,30	12,00		
	0,06	0,19	21,00		
<b>Médias</b>	<b>0,10</b>	<b>0,19</b>	<b>22,60</b>		
<b>SD</b>	<b>0,04</b>	<b>0,08</b>	<b>14,12</b>		
118035_IT_120_LB_39_FLAG_4_ms	0,09	0,24	14,00	34	5

	0,12	0,30	19,00		
	0,06	0,08	47,00		
	0,18	0,35	12,00		
	0,05	0,19	21,00		
<b>Médias</b>	<b>0,10</b>	<b>0,23</b>	<b>22,60</b>		
<b>SD</b>	<b>0,05</b>	<b>0,10</b>	<b>14,12</b>		
123074_IT_120_LB_39_FLAG_3_ms	0,26	0,60	12,00	20	5
	0,25	0,60	15,00		
	0,25	0,60	9,00		
	0,26	0,62	6,00		
	0,25	0,59	14,00		
<b>Médias</b>	<b>0,25</b>	<b>0,60</b>	<b>11,20</b>		
<b>SD</b>	<b>0,00</b>	<b>0,01</b>	<b>3,70</b>		
123074_IT_120_LB_39_FLAG_4_ms	0,32	0,47	12,00	11	5
	0,32	0,46	15,00		
	0,32	0,47	9,00		
	0,33	0,49	6,00		
	0,32	0,48	14,00		
<b>Médias</b>	<b>0,32</b>	<b>0,47</b>	<b>11,20</b>		
<b>SD</b>	<b>0,00</b>	<b>0,01</b>	<b>3,70</b>		
124084_IT_120_LB_39_FLAG_3_ms	0,36	0,52	5,00	48	5
	0,11	0,18	49,00		
	0,35	0,51	9,00		
	0,36	0,52	11,00		
	0,39	0,49	46,00		
<b>Médias</b>	<b>0,31</b>	<b>0,44</b>	<b>24,00</b>		
<b>SD</b>	<b>0,12</b>	<b>0,15</b>	<b>21,59</b>		
124084_IT_120_LB_39_FLAG_4_ms	0,37	0,47	5,00	43	5
	0,12	0,14	49,00		
	0,36	0,46	9,00		
	0,37	0,47	11,00		
	0,42	0,47	46,00		
<b>Médias</b>	<b>0,33</b>	<b>0,40</b>	<b>24,00</b>		
<b>SD</b>	<b>0,12</b>	<b>0,15</b>	<b>21,59</b>		
126039_IT_120_LB_39_FLAG_3_ms	0,21	0,08	32,00	19	5
	0,20	0,12	16,00		
	0,21	0,12	20,00		
	0,39	0,09	32,00		
	0,42	0,07	56,00		
<b>Médias</b>	<b>0,28</b>	<b>0,10</b>	<b>31,20</b>		
<b>SD</b>	<b>0,11</b>	<b>0,02</b>	<b>15,59</b>		
126039_IT_120_LB_39_FLAG_4_ms	0,22	0,28	32,00	17	5
	0,23	0,32	16,00		
	0,23	0,32	20,00		

	0,23	0,20	32,00		
	0,23	0,19	56,00		
<b>Médias</b>	<b>0,23</b>	<b>0,26</b>	<b>31,20</b>		
<b>SD</b>	<b>0,00</b>	<b>0,06</b>	<b>15,59</b>		
140075_IT_120_LB_39_FLAG_3_ms	0,15	0,40	24,00	49	5
	0,20	0,49	11,00		
	0,17	0,45	10,00		
	0,12	0,41	14,00		
	0,08	0,26	67,00		
<b>Médias</b>	<b>0,15</b>	<b>0,40</b>	<b>25,20</b>		
<b>SD</b>	<b>0,04</b>	<b>0,08</b>	<b>24,01</b>		
140075_IT_120_LB_39_FLAG_4_ms	0,14	0,36	24,00	30	5
	0,18	0,42	11,00		
	0,15	0,39	10,00		
	0,11	0,35	14,00		
	0,10	0,27	67,00		
<b>Médias</b>	<b>0,14</b>	<b>0,36</b>	<b>25,20</b>		
<b>SD</b>	<b>0,03</b>	<b>0,06</b>	<b>24,01</b>		
145053_IT_120_LB_39_FLAG_3_ms	0,23	0,62	11,00	104	5
	0,08	0,41	34,00		
	0,07	0,40	29,00		
	0,08	0,41	57,00		
	0,08	0,44	32,00		
<b>Médias</b>	<b>0,11</b>	<b>0,46</b>	<b>32,60</b>		
<b>SD</b>	<b>0,07</b>	<b>0,09</b>	<b>16,41</b>		
145053_IT_120_LB_39_FLAG_4_ms	0,24	0,63	11,00	88	5
	0,09	0,43	34,00		
	0,08	0,41	29,00		
	0,10	0,45	57,00		
	0,09	0,45	32,00		
<b>Médias</b>	<b>0,12</b>	<b>0,47</b>	<b>32,60</b>		
<b>SD</b>	<b>0,07</b>	<b>0,09</b>	<b>16,41</b>		
145086_IT_120_LB_39_FLAG_3_ms	0,14	0,38	21,00	57	5
	0,15	0,38	25,00		
	0,13	0,32	40,00		
	0,13	0,33	24,00		
	0,14	0,33	28,00		
<b>Médias</b>	<b>0,14</b>	<b>0,35</b>	<b>27,60</b>		
<b>SD</b>	<b>0,01</b>	<b>0,03</b>	<b>7,37</b>		
145086_IT_120_LB_39_FLAG_4_ms	0,12	0,33	21,00	51	5
	0,12	0,32	25,00		
	0,10	0,28	40,00		
	0,11	0,29	24,00		
	0,11	0,28	28,00		

<b>Médias</b>	<b>0,11</b>	<b>0,30</b>	<b>27,60</b>		
<b>SD</b>	<b>0,01</b>	<b>0,03</b>	<b>7,37</b>		
151087_IT_120_LB_39_FLAG_3_ms	0,20	0,56	62,00	64	5
	0,21	0,62	27,00		
	0,21	0,56	43,00		
	0,22	0,58	35,00		
	0,20	0,55	55,00		
<b>Médias</b>	<b>0,21</b>	<b>0,58</b>	<b>44,40</b>		
<b>SD</b>	<b>0,01</b>	<b>0,03</b>	<b>14,28</b>		
151087_IT_120_LB_39_FLAG_4_ms	0,20	0,44	62,00	39	5
	0,20	0,51	27,00		
	0,20	0,44	43,00		
	0,20	0,46	35,00		
	0,19	0,44	55,00		
<b>Médias</b>	<b>0,20</b>	<b>0,46</b>	<b>44,40</b>		
<b>SD</b>	<b>0,00</b>	<b>0,03</b>	<b>14,28</b>		
156079_IT_120_LB_39_FLAG_3_ms	0,17	0,31	15,00	48	5
	0,20	0,30	28,00		
	0,16	0,30	25,00		
	0,53	0,62	8,00		
	0,19	0,28	35,00		
<b>Médias</b>	<b>0,25</b>	<b>0,36</b>	<b>22,20</b>		
<b>SD</b>	<b>0,16</b>	<b>0,15</b>	<b>10,71</b>		
156079_IT_120_LB_39_FLAG_4_ms	0,25	0,51	15,00	32	5
	0,20	0,39	28,00		
	0,24	0,48	25,00		
	0,57	0,75	8,00		
	0,23	0,46	35,00		
<b>Médias</b>	<b>0,30</b>	<b>0,52</b>	<b>22,20</b>		
<b>SD</b>	<b>0,15</b>	<b>0,14</b>	<b>10,71</b>		
163085_IT_120_LB_39_FLAG_3_ms	0,29	0,36	7,00	15	7
	0,25	0,36	13,00		
	0,28	0,36	3,00		
	0,25	0,37	6,00		
	0,25	0,35	20,00		
	0,25	0,37	9,00		
	0,23	0,25	18,00		
<b>Médias</b>	<b>0,26</b>	<b>0,35</b>	<b>10,86</b>		
<b>SD</b>	<b>0,02</b>	<b>0,04</b>	<b>6,36</b>		
163085_IT_120_LB_39_FLAG_4_ms	0,27	0,39	7,00	9	7
	0,26	0,38	13,00		
	0,27	0,38	3,00		
	0,26	0,38	6,00		
	0,26	0,37	20,00		

	0,26	0,39	9,00		
	0,25	0,29	18,00		
<b>Médias</b>	<b>0,26</b>	<b>0,37</b>	<b>10,86</b>		
<b>SD</b>	<b>0,01</b>	<b>0,04</b>	<b>6,36</b>		
167083_IT_120_LB_39_FLAG_3_ms	0,21	0,47	5,00	181	6
	0,16	0,43	28,00		
	0,21	0,46	6,00		
	0,21	0,47	8,00		
	0,21	0,47	5,00		
	0,21	0,47	5,00		
<b>Médias</b>	<b>0,20</b>	<b>0,46</b>	<b>9,50</b>		
<b>SD</b>	<b>0,02</b>	<b>0,02</b>	<b>9,14</b>		
167083_IT_120_LB_39_FLAG_4_ms	0,22	0,51	5,00	197	6
	0,17	0,48	28,00		
	0,22	0,51	6,00		
	0,22	0,51	8,00		
	0,22	0,51	5,00		
	0,22	0,51	5,00		
<b>Médias</b>	<b>0,21</b>	<b>0,50</b>	<b>9,50</b>		
<b>SD</b>	<b>0,02</b>	<b>0,01</b>	<b>9,14</b>		
169012_IT_120_LB_39_FLAG_3_ms	0,13	0,50	14,00	154	9
	0,13	0,49	12,00		
	0,07	0,34	33,00		
	0,14	0,52	14,00		
	0,13	0,50	14,00		
	0,08	0,36	24,00		
	0,07	0,33	21,00		
	0,08	0,37	16,00		
	0,07	0,34	19,00		
<b>Médias</b>	<b>0,10</b>	<b>0,42</b>	<b>18,56</b>		
<b>SD</b>	<b>0,03</b>	<b>0,08</b>	<b>6,67</b>		
169012_IT_120_LB_39_FLAG_4_ms	0,11	0,40	14,00	61	9
	0,11	0,40	12,00		
	0,06	0,26	33,00		
	0,11	0,41	14,00		
	0,11	0,41	14,00		
	0,06	0,28	24,00		
	0,06	0,26	21,00		
	0,06	0,28	16,00		
	0,06	0,26	19,00		
<b>Médias</b>	<b>0,08</b>	<b>0,33</b>	<b>18,56</b>		
<b>SD</b>	<b>0,03</b>	<b>0,07</b>	<b>6,67</b>		
175043_IT_120_LB_39_FLAG_3_ms	0,62	0,57	3,00	38	8
	0,62	0,57	6,00		

	0,53	0,54	14,00		
	0,57	0,55	5,00		
	0,51	0,53	15,00		
	0,62	0,57	4,00		
	0,32	0,21	208,00		
	0,57	0,55	6,00		
<b>Médias</b>	<b>0,54</b>	<b>0,51</b>	<b>32,63</b>		
<b>SD</b>	<b>0,10</b>	<b>0,12</b>	<b>71,00</b>		
175043_IT_120_LB_39_FLAG_4_ms	0,43	0,22	3,00	17	8
	0,43	0,22	6,00		
	0,45	0,20	14,00		
	0,40	0,21	5,00		
	0,46	0,19	15,00		
	0,43	0,22	4,00		
	0,62	0,15	208,00		
	0,40	0,21	6,00		
<b>Médias</b>	<b>0,45</b>	<b>0,20</b>	<b>32,63</b>		
<b>SD</b>	<b>0,07</b>	<b>0,03</b>	<b>71,00</b>		
176035_IT_120_LB_39_FLAG_3_ms	0,10	0,21	39,00	38	6
	0,19	0,47	10,00		
	0,22	0,39	8,00		
	0,13	0,31	15,00		
	0,11	0,25	26,00		
	0,10	0,21	24,00		
<b>Médias</b>	<b>0,14</b>	<b>0,31</b>	<b>20,33</b>		
<b>SD</b>	<b>0,05</b>	<b>0,11</b>	<b>11,67</b>		
176035_IT_120_LB_39_FLAG_4_ms	0,11	0,29	39,00	41	6
	0,19	0,51	10,00		
	0,24	0,48	8,00		
	0,14	0,37	15,00		
	0,11	0,29	26,00		
	0,12	0,29	24,00		
<b>Médias</b>	<b>0,15</b>	<b>0,37</b>	<b>20,33</b>		
<b>SD</b>	<b>0,05</b>	<b>0,10</b>	<b>11,67</b>		
176039_IT_120_LB_39_FLAG_3_ms	0,20	0,43	7,00	38	5
	0,45	0,60	3,00		
	0,21	0,44	8,00		
	0,18	0,39	6,00		
	0,21	0,44	5,00		
<b>Médias</b>	<b>0,25</b>	<b>0,46</b>	<b>5,80</b>		
<b>SD</b>	<b>0,12</b>	<b>0,08</b>	<b>1,92</b>		
176039_IT_120_LB_39_FLAG_4_ms	0,20	0,43	7,00	20	5
	0,44	0,63	3,00		
	0,20	0,44	8,00		

	0,19	0,44	6,00		
	0,20	0,44	5,00		
<b>Médias</b>	<b>0,25</b>	<b>0,47</b>	<b>5,80</b>		
<b>SD</b>	<b>0,11</b>	<b>0,09</b>	<b>1,92</b>		
183055_IT_120_LB_39_FLAG_3_ms	0,17	0,28	7,00	17	5
	0,12	0,18	52,00		
	0,30	0,44	7,00		
	0,16	0,28	7,00		
	0,17	0,29	18,00		
<b>Médias</b>	<b>0,18</b>	<b>0,30</b>	<b>18,20</b>		
<b>SD</b>	<b>0,07</b>	<b>0,09</b>	<b>19,49</b>		
183055_IT_120_LB_39_FLAG_4_ms	0,16	0,24	7,00	8	5
	0,11	0,14	52,00		
	0,29	0,42	7,00		
	0,15	0,23	7,00		
	0,17	0,25	18,00		
<b>Médias</b>	<b>0,18</b>	<b>0,25</b>	<b>18,20</b>		
<b>SD</b>	<b>0,07</b>	<b>0,10</b>	<b>19,49</b>		
187071_IT_120_LB_39_FLAG_3_ms	0,08	0,23	14,00	27	6
	0,09	0,29	9,00		
	0,09	0,28	11,00		
	0,17	0,34	9,00		
	0,09	0,28	8,00		
	0,06	0,15	35,00		
<b>Médias</b>	<b>0,10</b>	<b>0,26</b>	<b>14,33</b>		
<b>SD</b>	<b>0,04</b>	<b>0,06</b>	<b>10,35</b>		
187071_IT_120_LB_39_FLAG_4_ms	0,15	0,30	14,00	19	6
	0,16	0,35	9,00		
	0,15	0,33	11,00		
	0,23	0,40	9,00		
	0,15	0,34	8,00		
	0,13	0,23	35,00		
<b>Médias</b>	<b>0,16</b>	<b>0,33</b>	<b>14,33</b>		
<b>SD</b>	<b>0,03</b>	<b>0,06</b>	<b>10,35</b>		
19021_IT_120_LB_39_FLAG_3_ms	0,14	0,35	15,00	77	6
	0,14	0,36	14,00		
	0,14	0,35	15,00		
	0,11	0,26	33,00		
	0,11	0,32	21,00		
	0,11	0,31	45,00		
<b>Médias</b>	<b>0,13</b>	<b>0,33</b>	<b>23,83</b>		
<b>SD</b>	<b>0,02</b>	<b>0,04</b>	<b>12,59</b>		
19021_IT_120_LB_39_FLAG_4_ms	0,17	0,37	15,00	43	6
	0,17	0,38	14,00		

	0,17	0,37	15,00		
	0,14	0,26	33,00		
	0,13	0,30	21,00		
	0,13	0,29	45,00		
<b>Médias</b>	<b>0,15</b>	<b>0,33</b>	<b>23,83</b>		
<b>SD</b>	<b>0,02</b>	<b>0,05</b>	<b>12,59</b>		
196015_IT_120_LB_39_FLAG_3_ms	0,39	0,60	25,00	48	5
	0,20	0,45	38,00		
	0,23	0,46	21,00		
	0,22	0,44	51,00		
	0,26	0,51	18,00		
<b>Médias</b>	<b>0,26</b>	<b>0,49</b>	<b>30,60</b>		
<b>SD</b>	<b>0,08</b>	<b>0,07</b>	<b>13,72</b>		
196015_IT_120_LB_39_FLAG_4_ms	0,40	0,50	25,00	31	5
	0,27	0,42	38,00		
	0,28	0,43	21,00		
	0,27	0,40	51,00		
	0,32	0,47	18,00		
<b>Médias</b>	<b>0,31</b>	<b>0,45</b>	<b>30,60</b>		
<b>SD</b>	<b>0,06</b>	<b>0,04</b>	<b>13,72</b>		
216053_IT_120_LB_39_FLAG_3_ms	0,21	0,47	15,00	72	5
	0,20	0,44	20,00		
	0,19	0,42	36,00		
	0,17	0,37	61,00		
	0,20	0,45	25,00		
<b>Médias</b>	<b>0,19</b>	<b>0,43</b>	<b>31,40</b>		
<b>SD</b>	<b>0,02</b>	<b>0,04</b>	<b>18,28</b>		
216053_IT_120_LB_39_FLAG_4_ms	0,20	0,45	15,00	56	5
	0,18	0,43	20,00		
	0,18	0,41	36,00		
	0,15	0,36	61,00		
	0,18	0,42	25,00		
<b>Médias</b>	<b>0,18</b>	<b>0,41</b>	<b>31,40</b>		
<b>SD</b>	<b>0,02</b>	<b>0,04</b>	<b>18,28</b>		
219090_IT_120_LB_39_FLAG_3_ms	0,19	0,36	12,00	54	5
	0,20	0,37	13,00		
	0,21	0,38	8,00		
	0,20	0,38	7,00		
	0,21	0,38	8,00		
<b>Médias</b>	<b>0,20</b>	<b>0,37</b>	<b>9,60</b>		
<b>SD</b>	<b>0,01</b>	<b>0,01</b>	<b>2,70</b>		
219090_IT_120_LB_39_FLAG_4_ms	0,14	0,33	12,00	52	5
	0,15	0,34	13,00		
	0,16	0,35	8,00		



	0,15	0,34	7,00		
	0,16	0,35	8,00		
<b>Médias</b>	<b>0,15</b>	<b>0,34</b>	<b>9,60</b>		
<b>SD</b>	<b>0,01</b>	<b>0,01</b>	<b>2,70</b>		
232038_IT_120_LB_39_FLAG_3_ms	0,12	0,32	16,00	31	5
	0,14	0,35	26,00		
	0,10	0,20	31,00		
	0,12	0,27	26,00		
	0,11	0,26	20,00		
<b>Médias</b>	<b>0,12</b>	<b>0,28</b>	<b>23,80</b>		
<b>SD</b>	<b>0,02</b>	<b>0,06</b>	<b>5,85</b>		
232038_IT_120_LB_39_FLAG_4_ms	0,12	0,28	16,00	19	5
	0,14	0,32	26,00		
	0,12	0,22	31,00		
	0,12	0,24	26,00		
	0,12	0,23	20,00		
<b>Médias</b>	<b>0,12</b>	<b>0,26</b>	<b>23,80</b>		
<b>SD</b>	<b>0,01</b>	<b>0,04</b>	<b>5,85</b>		
239096_IT_120_LB_39_FLAG_3_ms	0,21	0,68	22,00	63	5
	0,24	0,70	24,00		
	0,11	0,58	17,00		
	0,13	0,60	21,00		
	0,12	0,57	18,00		
<b>Médias</b>	<b>0,16</b>	<b>0,63</b>	<b>20,40</b>		
<b>SD</b>	<b>0,06</b>	<b>0,06</b>	<b>2,88</b>		
239096_IT_120_LB_39_FLAG_4_ms	0,23	0,61	22,00	35	5
	0,23	0,58	24,00		
	0,12	0,48	17,00		
	0,14	0,52	21,00		
	0,14	0,49	18,00		
<b>Médias</b>	<b>0,17</b>	<b>0,53</b>	<b>20,40</b>		
<b>SD</b>	<b>0,05</b>	<b>0,06</b>	<b>2,88</b>		
271035_IT_120_LB_39_FLAG_3_ms	0,13	0,47	32,00	64	5
	0,12	0,44	52,00		
	0,34	0,73	9,00		
	0,10	0,43	34,00		
	0,09	0,39	45,00		
<b>Médias</b>	<b>0,16</b>	<b>0,49</b>	<b>34,40</b>		
<b>SD</b>	<b>0,11</b>	<b>0,14</b>	<b>16,38</b>		
271035_IT_120_LB_39_FLAG_4_ms	0,12	0,37	32,00	37	5
	0,12	0,37	52,00		
	0,33	0,66	9,00		
	0,09	0,35	34,00		
	0,09	0,33	45,00		

<b>Médias</b>	<b>0,15</b>	<b>0,42</b>	<b>34,40</b>		
<b>SD</b>	<b>0,10</b>	<b>0,14</b>	<b>16,38</b>		
274007_IT_120_LB_39_FLAG_3_ms	0,27	0,24	64,00	45	5
	0,29	0,25	71,00		
	0,18	0,12	31,00		
	0,22	0,17	60,00		
	0,21	0,16	21,00		
<b>Médias</b>	<b>0,23</b>	<b>0,19</b>	<b>49,40</b>		
<b>SD</b>	<b>0,04</b>	<b>0,05</b>	<b>22,01</b>		
274007_IT_120_LB_39_FLAG_4_ms	0,32	0,29	64,00	22	5
	0,33	0,30	71,00		
	0,24	0,18	31,00		
	0,27	0,23	60,00		
	0,26	0,21	21,00		
<b>Médias</b>	<b>0,28</b>	<b>0,24</b>	<b>49,40</b>		
<b>SD</b>	<b>0,04</b>	<b>0,05</b>	<b>22,01</b>		
28075_IT_120_LB_39_FLAG_3_ms	0,52	0,56	4,00	22	6
	0,53	0,56	4,00		
	0,52	0,56	4,00		
	0,25	0,40	8,00		
	0,24	0,40	7,00		
	0,18	0,36	9,00		
<b>Médias</b>	<b>0,37</b>	<b>0,47</b>	<b>6,00</b>		
<b>SD</b>	<b>0,17</b>	<b>0,09</b>	<b>2,28</b>		
28075_IT_120_LB_39_FLAG_4_ms	0,50	0,48	4,00	11	6
	0,50	0,48	4,00		
	0,50	0,48	4,00		
	0,22	0,31	8,00		
	0,22	0,31	7,00		
	0,18	0,23	9,00		
<b>Médias</b>	<b>0,35</b>	<b>0,38</b>	<b>6,00</b>		
<b>SD</b>	<b>0,16</b>	<b>0,11</b>	<b>2,28</b>		
310007_IT_120_LB_39_FLAG_3_ms	0,16	0,28	14,00	22	5
	0,16	0,27	13,00		
	0,16	0,28	8,00		
	0,16	0,28	9,00		
	0,16	0,31	9,00		
<b>Médias</b>	<b>0,16</b>	<b>0,28</b>	<b>10,60</b>		
<b>SD</b>	<b>0,00</b>	<b>0,01</b>	<b>2,70</b>		
310007_IT_120_LB_39_FLAG_4_ms	0,18	0,28	14,00	10	5
	0,18	0,28	13,00		
	0,18	0,28	8,00		
	0,18	0,28	9,00		
	0,18	0,30	9,00		

<b>Médias</b>	<b>0,18</b>	<b>0,28</b>	<b>10,60</b>		
<b>SD</b>	<b>0,00</b>	<b>0,01</b>	<b>2,70</b>		
351093_IT_120_LB_39_FLAG_3_ms	0,29	0,56	9,00	128	5
	0,11	0,44	17,00		
	0,14	0,48	19,00		
	0,13	0,48	13,00		
	0,12	0,46	20,00		
<b>Médias</b>	<b>0,16</b>	<b>0,48</b>	<b>15,60</b>		
<b>SD</b>	<b>0,08</b>	<b>0,05</b>	<b>4,56</b>		
351093_IT_120_LB_39_FLAG_4_ms	0,29	0,55	9,00	124	5
	0,11	0,39	17,00		
	0,14	0,46	19,00		
	0,14	0,47	13,00		
	0,12	0,41	20,00		
<b>Médias</b>	<b>0,16</b>	<b>0,45</b>	<b>15,60</b>		
<b>SD</b>	<b>0,07</b>	<b>0,06</b>	<b>4,56</b>		
41004_IT_120_LB_39_FLAG_3_ms	0,27	0,61	6,00	21	5
	0,17	0,45	10,00		
	0,27	0,62	4,00		
	0,27	0,61	9,00		
	0,27	0,60	13,00		
<b>Médias</b>	<b>0,25</b>	<b>0,58</b>	<b>8,40</b>		
<b>SD</b>	<b>0,04</b>	<b>0,07</b>	<b>3,51</b>		
41004_IT_120_LB_39_FLAG_4_ms	0,20	0,38	6,00	19	5
	0,10	0,22	10,00		
	0,20	0,39	4,00		
	0,20	0,38	9,00		
	0,19	0,37	13,00		
<b>Médias</b>	<b>0,18</b>	<b>0,35</b>	<b>8,40</b>		
<b>SD</b>	<b>0,04</b>	<b>0,07</b>	<b>3,51</b>		
71046_IT_120_LB_39_FLAG_3_ms	0,07	0,04	12,00	11	5
	0,10	0,04	19,00		
	0,10	0,03	27,00		
	0,09	0,06	5,00		
	0,10	0,02	29,00		
<b>Médias</b>	<b>0,09</b>	<b>0,04</b>	<b>18,40</b>		
<b>SD</b>	<b>0,01</b>	<b>0,01</b>	<b>10,09</b>		
71046_IT_120_LB_39_FLAG_4_ms	0,08	0,08	12,00	9	5
	0,10	0,07	19,00		
	0,10	0,05	27,00		
	0,10	0,11	5,00		
	0,10	0,05	29,00		
<b>Médias</b>	<b>0,09</b>	<b>0,07</b>	<b>18,40</b>		

	<b>SD</b>	<b>0,01</b>	<b>0,02</b>	<b>10,09</b>		
78019_IT_120_LB_39_FLAG_3_ms		0,08	0,28	42,00	220	5
		0,08	0,29	67,00		
		0,11	0,29	28,00		
		0,06	0,26	37,00		
		0,07	0,26	35,00		
	<b>Médias</b>	<b>0,08</b>	<b>0,27</b>	<b>41,80</b>		
	<b>SD</b>	<b>0,02</b>	<b>0,01</b>	<b>14,96</b>		
78019_IT_120_LB_39_FLAG_4_ms		0,10	0,24	42,00	186	5
		0,09	0,23	67,00		
		0,13	0,24	28,00		
		0,11	0,23	37,00		
		0,10	0,23	35,00		
	<b>Médias</b>	<b>0,11</b>	<b>0,23</b>	<b>41,80</b>		
	<b>SD</b>	<b>0,01</b>	<b>0,01</b>	<b>14,96</b>		
8023_IT_120_LB_39_FLAG_3_ms		0,33	0,53	4,00	33	5
		0,34	0,54	7,00		
		0,32	0,52	12,00		
		0,33	0,52	21,00		
		0,72	0,72	4,00		
	<b>Médias</b>	<b>0,41</b>	<b>0,57</b>	<b>9,60</b>		
	<b>SD</b>	<b>0,18</b>	<b>0,08</b>	<b>7,16</b>		
8023_IT_120_LB_39_FLAG_4_ms		0,24	0,31	4,00	17	5
		0,24	0,31	7,00		
		0,23	0,29	12,00		
		0,24	0,30	21,00		
		0,63	0,58	4,00		
	<b>Médias</b>	<b>0,32</b>	<b>0,36</b>	<b>9,60</b>		
	<b>SD</b>	<b>0,18</b>	<b>0,12</b>	<b>7,16</b>		
86016_IT_120_LB_39_FLAG_3_ms		0,30	0,14	24,00	14	5
		0,32	0,29	4,00		
		0,30	0,14	23,00		
		0,31	0,14	41,00		
		0,30	0,14	17,00		
	<b>Médias</b>	<b>0,31</b>	<b>0,17</b>	<b>21,80</b>		
	<b>SD</b>	<b>0,01</b>	<b>0,07</b>	<b>13,37</b>		
86016_IT_120_LB_39_FLAG_4_ms		0,33	0,14	24,00	4	5
		0,33	0,30	4,00		
		0,32	0,15	23,00		
		0,33	0,14	41,00		
		0,32	0,14	17,00		
	<b>Médias</b>	<b>0,33</b>	<b>0,17</b>	<b>21,80</b>		
	<b>SD</b>	<b>0,00</b>	<b>0,07</b>	<b>13,37</b>		
87065_IT_120_LB_39_FLAG_3_ms		0,55	0,72	8,00	135	5

	0,49	0,69	23,00		
	0,25	0,64	7,00		
	0,40	0,62	53,00		
	0,38	0,68	3,00		
<b>Médias</b>	<b>0,41</b>	<b>0,67</b>	<b>18,80</b>		
<b>SD</b>	<b>0,11</b>	<b>0,04</b>	<b>20,57</b>		
87065_IT_120_LB_39_FLAG_4_ms	0,44	0,45	8,00	40	5
	0,38	0,40	23,00		
	0,24	0,32	7,00		
	0,33	0,37	53,00		
	0,32	0,43	3,00		
<b>Médias</b>	<b>0,34</b>	<b>0,39</b>	<b>18,80</b>		
<b>SD</b>	<b>0,07</b>	<b>0,05</b>	<b>20,57</b>		
89072_IT_120_LB_39_FLAG_3_ms	0,16	0,46	13,00	34	5
	0,13	0,40	30,00		
	0,15	0,46	17,00		
	0,17	0,50	10,00		
	0,14	0,42	19,00		
<b>Médias</b>	<b>0,15</b>	<b>0,45</b>	<b>17,80</b>		
<b>SD</b>	<b>0,01</b>	<b>0,04</b>	<b>7,66</b>		
89072_IT_120_LB_39_FLAG_4_ms	0,18	0,50	13,00	29	5
	0,15	0,42	30,00		
	0,17	0,48	17,00		
	0,19	0,55	10,00		
	0,16	0,46	19,00		
<b>Médias</b>	<b>0,17</b>	<b>0,48</b>	<b>17,80</b>		
<b>SD</b>	<b>0,01</b>	<b>0,05</b>	<b>7,66</b>		
97017_IT_120_LB_39_FLAG_3_ms	0,24	0,52	15,00	75	6
	0,30	0,64	12,00		
	0,24	0,48	26,00		
	0,29	0,60	11,00		
	0,26	0,59	18,00		
	0,22	0,48	34,00		
<b>Médias</b>	<b>0,26</b>	<b>0,55</b>	<b>19,33</b>		
<b>SD</b>	<b>0,03</b>	<b>0,07</b>	<b>8,98</b>		
97017_IT_120_LB_39_FLAG_4_ms	0,16	0,35	15,00	78	6
	0,22	0,45	12,00		
	0,16	0,31	26,00		
	0,21	0,42	11,00		
	0,19	0,41	18,00		
	0,17	0,30	34,00		
<b>Médias</b>	<b>0,19</b>	<b>0,37</b>	<b>19,33</b>		
<b>SD</b>	<b>0,03</b>	<b>0,06</b>	<b>8,98</b>		
97033_IT_120_LB_39_FLAG_3_ms	0,33	0,57	13,00	94	5

	0,18	0,44	32,00		
	0,32	0,51	22,00		
	0,16	0,36	24,00		
	0,15	0,33	25,00		
<b>Médias</b>	<b>0,23</b>	<b>0,44</b>	<b>23,20</b>		
<b>SD</b>	<b>0,09</b>	<b>0,10</b>	<b>6,83</b>		
97033_IT_120_LB_39_FLAG_4_ms	0,32	0,52	13,00	54	5
	0,18	0,40	32,00		
	0,31	0,50	22,00		
	0,16	0,34	24,00		
	0,17	0,35	25,00		
<b>Médias</b>	<b>0,23</b>	<b>0,42</b>	<b>23,20</b>		
<b>SD</b>	<b>0,08</b>	<b>0,08</b>	<b>6,83</b>		

Tabela 6 – Dados gerados durante a validação.

### C. Todo o Conjunto de Imagens Filtradas

Abaixo a Tabela 7 contendo todas as imagens do teste e suas respectivas filtragens com o método tradicional e método modificado.

Imagem Original



232038

Filtro Adaptativo



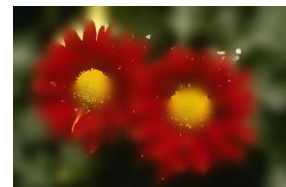
Filtro Tradicional



216053



124084





310007



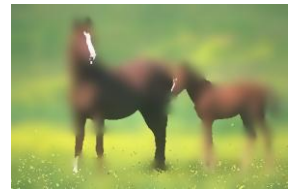
118035



97017



113016



113044



97033





19021



8023



28075



41004



71046



78019







86016



87065



123074



126039



140075



145053





145086



151087



156079



163085



169012



175043





176039



176035



183055



196015



219090



239096





351093



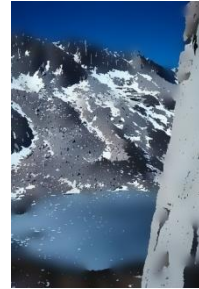
274007



271035



187071



167083



89072



100080

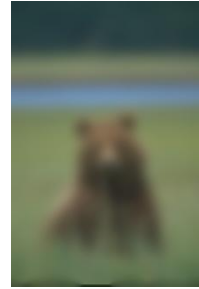
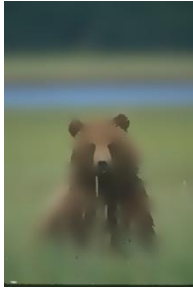


Tabela 7 – Imagens originais seguidas da filtragem pelo método tradicional e em seguida pelo método adaptado.

# Livros Grátis

( <http://www.livrosgratis.com.br> )

Milhares de Livros para Download:

[Baixar livros de Administração](#)

[Baixar livros de Agronomia](#)

[Baixar livros de Arquitetura](#)

[Baixar livros de Artes](#)

[Baixar livros de Astronomia](#)

[Baixar livros de Biologia Geral](#)

[Baixar livros de Ciência da Computação](#)

[Baixar livros de Ciência da Informação](#)

[Baixar livros de Ciência Política](#)

[Baixar livros de Ciências da Saúde](#)

[Baixar livros de Comunicação](#)

[Baixar livros do Conselho Nacional de Educação - CNE](#)

[Baixar livros de Defesa civil](#)

[Baixar livros de Direito](#)

[Baixar livros de Direitos humanos](#)

[Baixar livros de Economia](#)

[Baixar livros de Economia Doméstica](#)

[Baixar livros de Educação](#)

[Baixar livros de Educação - Trânsito](#)

[Baixar livros de Educação Física](#)

[Baixar livros de Engenharia Aeroespacial](#)

[Baixar livros de Farmácia](#)

[Baixar livros de Filosofia](#)

[Baixar livros de Física](#)

[Baixar livros de Geociências](#)

[Baixar livros de Geografia](#)

[Baixar livros de História](#)

[Baixar livros de Línguas](#)

[Baixar livros de Literatura](#)  
[Baixar livros de Literatura de Cordel](#)  
[Baixar livros de Literatura Infantil](#)  
[Baixar livros de Matemática](#)  
[Baixar livros de Medicina](#)  
[Baixar livros de Medicina Veterinária](#)  
[Baixar livros de Meio Ambiente](#)  
[Baixar livros de Meteorologia](#)  
[Baixar Monografias e TCC](#)  
[Baixar livros Multidisciplinar](#)  
[Baixar livros de Música](#)  
[Baixar livros de Psicologia](#)  
[Baixar livros de Química](#)  
[Baixar livros de Saúde Coletiva](#)  
[Baixar livros de Serviço Social](#)  
[Baixar livros de Sociologia](#)  
[Baixar livros de Teologia](#)  
[Baixar livros de Trabalho](#)  
[Baixar livros de Turismo](#)