

**JOÃO MARCELO RIBEIRO**

**Segmentação de pele humana em imagens  
coloridas baseada em valores das médias da  
vizinhança em subimagens**

Dissertação apresentada à Escola de Engenharia  
de São Carlos da Universidade de São Paulo como  
requisito para obtenção do título de Mestre em  
Engenharia Elétrica.

Área de Concentração: Visão Computacional  
Orientador: Prof. Dr. Adilson Gonzaga

São Carlos  
2007

# **Livros Grátis**

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

Milhares de livros grátis para download.

*Dedico este trabalho à minha esposa, pois esteve sempre presente, incentivando-me com paciência e compreendendo que sua realização é um plano de aprendizado da vida.*

# Agradecimentos

Agradeço a todos que me ajudaram a concluir este trabalho, dentre os quais destaco:

Em primeiro lugar a Deus, onde sempre deposito minha confiança, que sabe, com certeza, o que é melhor para meu aprendizado e crescimento.

À Escola de Engenharia de São Carlos, da Universidade de São Paulo, pela oportunidade de realização deste mestrado.

Ao meu orientador, Prof. Dr. Adilson Gonzaga, pela paciência, compreensão, amizade e dedicada orientação em todas as etapas do trabalho possibilitando sua realização.

À minha esposa Luciana que me incentivou, esteve sempre presente, compreendeu e lutou junto para que eu pudesse alcançar este objetivo.

À toda minha família, em especial aos meus pais, que sempre me apoiaram e deram forças oferecendo palavras de motivação para a realização deste trabalho.

Aos colegas de pós-graduação pela ajuda nos momentos necessários.

Aos professores e funcionários do programa de pós-graduação, que sempre se mostraram disponíveis em ajudar.

Enfim, a todos que compreenderam a importância deste e que auxiliaram de alguma forma no desenvolvimento deste trabalho, direta ou indiretamente.



# Resumo

A segmentação de pele humana, em imagens coloridas, tem sido largamente estudada nos últimos anos servindo de fundamento para muitos outros estudos como, por exemplo, a detecção de faces. Dentre as inúmeras aplicações de trabalhos relativos à segmentação de pele humana está a de se localizar uma determinada pessoa em locais de grande concentração humana tais como: avenidas, terminais de ônibus, aeroportos, *shopping centers* e estádios. Assim, a necessidade de se obter um sistema que classifique de forma adequada a pele humana tornou-se a principal motivação para o desenvolvimento deste trabalho.

Assim, propõe-se uma metodologia para melhorar a segmentação de pele humana em imagens coloridas através de um algoritmo mais eficiente. O algoritmo é baseado na média de vizinhanças cujos valores limites, para definição do intervalo de cor equivalente à pele humana, são obtidas através de uma imagem padrão, gerada a priori, com amostras de pele humana. Esta imagem é chamada de "colcha de retalhos".

A metodologia tem como base de comparação trabalhos anteriores similares, principalmente o desenvolvido por Kovac et al. (2003). Os resultados mostram um desempenho superior da metodologia proposta.

**PALAVRAS-CHAVE:** Pele humana, segmentação, média de vizinhança, subimagens, resolução de imagens.



# Abstract

The segmentation of human skin, in colored images, has been studied broadly for the last years serving as foundation for many other studies as, for instance, the detection of faces. Among the countless applications of works related to the segmentation of human skin it is the one of localizing a certain person in places of great human concentration such as: avenues, bus terminals, airports, shopping centers and stadiums. This way, the need to obtain a system that classifies in an appropriate way the human skin became the main motivation for the development of this work.

This way, it is proposed a methodology to improve the segmentation of human skin in colored images through a more efficient algorithm. The algorithm is based on the average of neighborhoods whose limit values, for definition of the interval of equivalent color to the human skins, are obtained through an image pattern, generated in priori, with samples of human skin. This image is called "bedspread of remnants."

The methodology has as base of comparison similar previous works, mainly developed by Kovac et al. (2003). The results show a superior performance of the proposed methodology.

**KEY WORDS:** Human skin, segmentation, neighborhood average, sub-images, images resolution.





# Lista de Figuras

2.1	Diagrama do olho humano . . . . .	4
2.2	A composição de uma imagem e a localização de um pixel. . . . .	5
2.3	Elementos funcionais básicos de um sistema de processamento de imagens: Ar- mazemamento, Processamento, Aquisição, Exibição e Comunicação. . . . .	7
2.4	Posição: O mesmo objeto em posições diferentes. . . . .	7
2.5	Tamanho: O mesmo objeto em tamanhos diferentes. . . . .	8
2.6	Formato: Objetos diferentes pertencentes à mesma classe. . . . .	8
2.7	Perspectiva: Dificuldades para a visão artificial compreender a imagem. . . . .	8
2.8	Iluminação: O mesmo objeto com diferentes iluminações. . . . .	9
3.1	Espectro de cores visto quando a luz branca passa pelo prisma. . . . .	12
3.2	Cores Primárias e Secundárias. . . . .	13
3.3	Diagrama de Cromaticidade. Fonte: (GONZALES E WOODS, 1992). . . . .	14
3.4	Representação do Modelo através do cubo RGB. . . . .	15
3.5	Representação e localização de uma cor específica no cubo RGB. . . . .	15
3.6	O cone representando as cores do modelo HSV. . . . .	16
3.7	O modelo HSV e a representação das suas características. Fonte: (LEÃO ET AL., 2005). . . . .	18
4.1	Exemplo de vizinhança 3x3. . . . .	20
5.1	Algumas das imagens obtidas para auxiliar na escolha do modelo de cores. . . . .	31
5.2	<b>Colcha de retalhos</b> (ao centro) montada a partir de pedaços de imagens de pele humana. . . . .	31
5.3	Organização dos registros no banco de dados. . . . .	32

5.4	Pessoa em diferentes posições (ou expressões) utilizadas para a montagem da nova colcha de retalhos. . . . .	33
5.5	Nova <b>colcha de retalhos</b> montada da mesma forma que a anterior. . . . .	34
5.6	Nova <b>colcha de retalhos</b> utilizando uma subimagem para a obtenção das médias. . . . .	35
5.7	Características de cada pixel no formato <b>RGB</b> da <b>imagem original</b> (resolução 6x4) e o resultado da <b>ampliação</b> (resolução 12x8) e <b>redução</b> (resolução 3x2) desta imagem. . . . .	37
5.8	Algumas imagens e suas resoluções para a realização dos testes. . . . .	38
6.1	Resultados obtidos para a escolha do modelo de cores, conforme a quantidade de registros gerados em cada uma das bases (Colcha de Retalhos e Imagem Teste de Pele) considerando as coincidências entre elas. . . . .	40
6.2	Colcha de retalhos em dois momentos: (1) com os pixels que não atendem às especificações de pele devidamente identificados e (2) após a correção dos erros identificados. . . . .	40
6.3	Gráfico contendo os valores encontrados nas subimagens da colcha de retalhos com resolução de 900x1000 pixels. . . . .	41
6.4	Resultados da segmentação conforme os valores das médias obtidas dos diferentes tamanhos de vizinhança: 3x3, 5x5, 7x7 e 9x9 em imagem padrão de resolução 900x1000. . . . .	42
6.5	Resultados da segmentação conforme a resolução da colcha de retalhos: 160x120, 320x240, 640x480 e 900x1000 considerando uma subimagem de tamanho 3x3. . . . .	43
6.6	Valores das menores e maiores médias obtidas para várias resoluções e várias subimagens. . . . .	44
6.7	. . . . .	50
6.8	. . . . .	51
6.9	Resultado das imagens segmentadas conforme Kovac et al. (2003) . . . . .	52
6.10	Comparação de resultados de forma visual. . . . .	52

# Sumário

Dedicatória . . . . .	i
Agradecimentos . . . . .	iii
Resumo . . . . .	v
Abstract . . . . .	v
Lista de Figuras . . . . .	ix
Sumário . . . . .	xi
<b>1 Introdução</b>	<b>1</b>
1.1 Motivação . . . . .	1
1.2 Objetivos . . . . .	2
1.3 Organização desta Dissertação . . . . .	2
<b>2 Aparelhos Visuais</b>	<b>3</b>
2.1 Visão Natural x Visão Artificial . . . . .	3
2.2 Elementos de Visão Artificial . . . . .	5
2.3 Limites e Dificuldades . . . . .	6
2.3.1 Posição . . . . .	6
2.3.2 Tamanho e Formato . . . . .	7
2.3.3 Perspectiva . . . . .	8
2.3.4 Iluminação . . . . .	8
2.4 Áreas Relacionadas . . . . .	9
2.5 Considerações Finais . . . . .	9
<b>3 Modelos de Cores</b>	<b>11</b>
3.1 O Olho Humano e a Interpretação de Cores . . . . .	11

3.2	Fundamentos de cores . . . . .	11
3.2.1	Luz cromática e acromática . . . . .	12
3.2.2	Cores Primárias e Secundárias . . . . .	12
3.2.3	Distinção de Cores . . . . .	13
3.3	Modelos de Cores . . . . .	13
3.3.1	Modelo RGB . . . . .	13
3.3.2	Modelo CMYK . . . . .	15
3.3.3	Modelo HSV e HSI . . . . .	16
3.3.4	Outros Modelos de Cor . . . . .	17
3.4	Considerações Finais . . . . .	18
<b>4</b>	<b>Segmentação da Pele Humana em Imagens Coloridas</b>	<b>19</b>
4.1	Métodos para Segmentação . . . . .	19
4.2	Algoritmos Baseados em Vizinhança . . . . .	20
4.3	A Segmentação de Pele Humana . . . . .	20
4.4	Escolha do Modelo de Cor . . . . .	21
4.5	Métodos de Classificação de Pixel de Cor de Pele . . . . .	23
4.6	Considerações Finais . . . . .	25
<b>5</b>	<b>Metodologia Proposta</b>	<b>29</b>
5.1	Materiais . . . . .	29
5.2	Métodos . . . . .	29
5.2.1	Escolha do modelo de cores . . . . .	30
5.2.2	Geração da Colcha de Retalhos para a Base de Testes . . . . .	33
5.2.3	Cálculo das Médias Variando-se o Tamanho das Vizinhanças das Subimagens . . . . .	33
5.2.4	Cálculo das Médias de Vizinhanças em Imagens com Diversas Resoluções . . . . .	35
5.2.5	Segmentação Considerando-se o Tamanho da Vizinhança de Pixels nas Subimagens . . . . .	36
5.2.6	Segmentação Considerando a Resolução da Imagem da <i>Colcha de Retalhos</i> . . . . .	38

---

5.3	Considerações Finais . . . . .	38
<b>6</b>	<b>Resultados e Conclusões</b>	<b>39</b>
6.1	Resultados . . . . .	39
6.2	Análise dos Resultados . . . . .	42
6.2.1	Escolha do modelo de cores . . . . .	42
6.2.2	Colcha de Retalhos . . . . .	44
6.2.3	Tamanho da Vizinhança . . . . .	45
6.2.4	Resolução da Colcha de Retalhos . . . . .	45
6.2.5	Verificação completa . . . . .	45
6.3	Comparação dos Resultados . . . . .	46
6.3.1	Resultados comparados de forma visual (Qualitativa) . . . . .	46
6.3.2	Resultados comparados de forma numérica (Quantitativa) . . . . .	47
6.4	Conclusões . . . . .	47
6.5	Trabalhos Futuros . . . . .	48
	<b>Referências Bibliográficas</b>	<b>53</b>



# Capítulo 1

## Introdução

### 1.1 Motivação

A segmentação de pele humana em imagens coloridas pode ser muito útil para localizar pessoas em imagens como em estádios de futebol, teatros, cinemas, *shows*, aeroportos e outros locais com pouca ou grande concentração de gente. Em centros de compras, como *shopping centers* por exemplo, a localização de pessoas pode ser extremamente importante para: administradores, lojistas, e principalmente a segurança. Nesses locais não só a localização de determinadas pessoas pode ser importante, como também saber a quantidade das que estão presentes no local, além de saber em quais setores existe maior concentração podendo vir a ajudar nas orientações em caso de acidente ou até mesmo para questões estratégicas administrativas.

Muito se tem estudado sobre a segmentação de pele humana e suas aplicações. Encontra-se, na literatura, inúmeros trabalhos relativos ao uso de segmentação de pele, inclusive como um pré-processamento daquilo que se pretende realizar, como, por exemplo, em Cortez et al. (2006).

Mesmo sendo grandes motivos de estudos, a segmentação de pele em imagens coloridas possui alguns problemas de difícil solução e talvez por isso não obtenha a perfeição total. Fatores como a iluminação, têm sido bastante significativos no processamento de imagens, principalmente na segmentação de pele devido à sua pigmentação. Além disso, os constantes estudos sobre os tipos de pele (Carelli, 2006) elevam ainda mais a complexidade na sua classificação.



## 1.2 Objetivos

Como os métodos existentes na literatura que segmenta as regiões de pele humana não são eficientes para a detecção precisa de pessoas em cenas reais, o objetivo deste trabalho é desenvolver um método para identificar, com maior precisão, pele humana em imagens coloridas e em situações cotidianas.

Para isso propõe-se a geração de uma imagem padrão chamada de "colcha de retalhos" que fornecerá os parâmetros limites para identificação de pele humana em imagens gerais.

A segmentação será feita através de cálculos da média de cores em vizinhanças previamente estabelecidas. Pretende-se variar o tamanho da vizinhança e avaliar o efeito causado na eficiência da segmentação.

## 1.3 Organização desta Dissertação

Esta dissertação de mestrado é constituída por capítulos:

- Neste **primeiro capítulo**, foi realizada a introdução ao assunto, destacando a motivação e apresentando o objetivo desta pesquisa;
- no **Capítulo 2** são descritos o funcionamento e as diferenças entre o aparelho visual humano e o artificial;
- no **Capítulo 3** discute-se a percepção de cores, sua formação e os modelos mais comuns, como o RGB e o HSV;
- no **Capítulo 4** trata de segmentação de imagens, quais as finalidades, e quais as aplicações. Especificamente a segmentação de pele humana;
- no **Capítulo 5** apresentada a metodologia proposta e os algoritmos implementadas;
- no **Capítulo 6** são discutidos os resultados e apresentadas as conclusões;

# Capítulo 2

## Aparelhos Visuais

### 2.1 Visão Natural x Visão Artificial

Um dos principais objetivos do processamento de imagens é auxiliar o observador a interpretar o conteúdo de imagens identificando nestas aquilo que se pretende ou aquilo que ela contém. Assim, torna-se importante compreender basicamente o processo de percepção visual e sua composição.

Quanto à **visão natural**, pode-se dizer que o olho humano (figura 2.1), é basicamente uma esfera (o globo ocular) com um diâmetro médio de cerca de 20mm, composto por três membranas: (1) A **córnea**, que é um tecido resistente e transparente que cobre a superfície anterior do olho; (2) a **coróide**, que está situada abaixo da esclerótica e possui uma rede de vasos sanguíneos que servem como fonte de nutrição ao olho. Sua cobertura é fortemente pigmentada e sua extremidade anterior é dividida em *corpo ciliar* e *íris* (ou pupila) que contrai-se ou expande-se de forma a controlar a quantidade de luz que se espalha pelo globo ocular após entrar no olho; e (3) a **retina**, que é a membrana mais interna do olho e se estende sobre toda a porção posterior do olho.

É na retina que a luz do objeto é transformada em imagem. A luz incidente é focalizada pela **córnea** e pela **lente** do olho, criando a imagem do objeto que está sendo visto na **retina** (posicionado na parte traseira do globo ocular). Quando a luz passa pela córnea, ela atinge um foco na retina, gerando a imagem. A finalidade preliminar da lente é permitir que o olho focalize objetos em distâncias diferentes. (SHARMA, 2003).

Quanto à **visão artificial**, pode-se dizer que este tipo de visão é composta por elementos (mencionados de forma mais detalhada em 2.2) de um sistema de propósito geral capaz de desempenhar as operações de processamento de imagens. São eles:

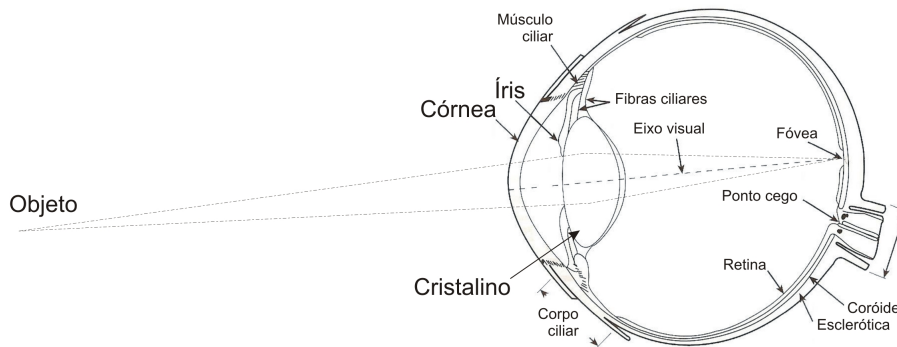


Figura 2.1: Diagrama do olho humano

(1) A **aquisição**, que é um dispositivo físico que tem como principal finalidade a obtenção da imagem propriamente dita; (2) o **armazenamento**, que, neste caso, é a introdução de dados ou informações em dispositivos físicos com a finalidade de recuperá-los posteriormente; (3) o **processamento**, que são os procedimentos, ou um conjunto de regras e operações bem definidas e ordenadas em um número finito de etapas, destinadas à solução de um problema (ou de uma classe de problemas); (4) a **comunicação**, que é a transmissão de informações entre sistemas através de sinais em fios ou de ondas eletromagnéticas; (5) a **exibição**, que é onde os sinais de saída, que foram enviados para um dispositivo de registro de imagem ou um módulo de exibição, são expostos.

Os computadores não compreendem imagens e sim uma **disposição de pontos**. As imagens são representadas por essas disposições, também chamadas de **matriz**, que podem ser dispostas de forma **bidimensional** ou **tridimensional**. Um ponto disposto em uma imagem bidimensional é chamado **pixel**.<sup>1</sup>

No caso mais simples os pixels estão localizados em uma grade retangular. A sua posição, na imagem, é dada na notação comum para matrizes. Ou seja, o primeiro elemento, ***M***, é a linha e o segundo elemento, ***N***, é a coluna. Se uma imagem digital possui  **$M \times N$**  pixels (de resolução), ela é representada por uma matriz deste tamanho e se o pixel for identificado com valores específicos para ***M*** e ***N***, significa que ele está localizado na **linha** de valor de ***M*** e na **coluna** de valor ***N*** dentro desta imagem. Conforme figura 2.2. (JAHNE, 2002).

<sup>1</sup>Um **pixel** é a menor unidade da imagem, ou seja, é um ponto pertencente a um imagem e o conjunto desses pontos (os **pixels**) formam a imagem, determinando assim a sua resolução.

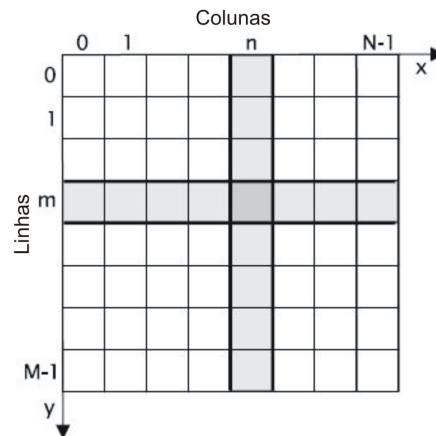


Figura 2.2: A composição de uma imagem e a localização de um pixel.

## 2.2 Elementos de Visão Artificial

Em Gonzales e Woods (1992), pode-se verificar os elementos de sistemas de processamento de imagens digitais.

Na **aquisição** de imagens, dois elementos são necessários para esta finalidade. "O primeiro é um *dispositivo físico* que seja sensível a uma banda do espectro de energia eletromagnética (como raios X, ultravioleta, visível, ou banda infravermelha) e que produza um sinal elétrico de saída proporcional a um nível de energia percebida. O segundo, chamado *digitalizador*, é um dispositivo para a conversão da saída elétrica de um dispositivo de sensoriamento físico para a forma digital. Como um exemplo, considere os aspectos básicos de um sistema de raios X. A saída de uma fonte de raios X é direcionada para um objeto e um meio sensível a raios X é colocado do outro lado do mesmo. O meio adquire assim uma imagem dos materiais (tais como ossos e tecidos), que possuam graus diferentes de absorção de raios X".

O **armazenamento** classifica-se em três principais categorias: (1) o *armazenamento para ser usado durante o processamento* - armazenamento por curto tempo, ocorrido geralmente através do uso da memória computacional; o (2) *armazenamento para acesso rápido ou "on-line"*, ocorrido geralmente na forma de discos magnéticos; e o (3) *armazenamento em arquivo*, que é um acesso que não se faz constantemente, ocorrido geralmente na forma de fitas magnéticas ou mídias usuais para aplicações de arquivamento de informações. O armazenamento é medido em diversas formas, ou seja, "uma imagem de 8 bits com resolução de 1024 x 1024 pixels requer um milhão de bytes para seu armazenamento. Assim, o provimento de armazenamento adequado é sempre um desafio para os projetos de sistemas de processamento de imagens".

O **processamento** ”envolve procedimentos que são geralmente expressos em forma algorítmicas. Assim, com exceção da aquisição e exibição de imagens, a maioria das funções de processamento de imagens pode ser implementada em *software*”. Além do *software*, outra dependência que se pode ter no processamento é com relação ao *hardware*. Ou seja, uma das razões que envolvem *hardware* especializado para processamento de imagens é a necessidade de velocidade em algumas aplicações. ”O processamento de imagens é caracterizado por soluções específicas. Desse modo, técnicas que funcionam bem em uma área podem se mostrar totalmente inadequadas em outra área”. Portanto, com o desenvolvimento cada vez maior da tecnologia, o que se faz atualmente é partir de um início mais avançado do que se fazia anteriormente.

A **comunicação** entre computadores - que é a transmissão de informações entre uma fonte e um destinatário - composta de hardware e software específicos para essa finalidade encontra-se disponível na maioria dos equipamentos. A comunicação quando realizada entre pontos a uma grande distância apresenta um desafio maior e mais complexo, pois as imagens digitais apresentam uma quantidade significativa de dados e sua transmissão pode sofrer distorções ou perdas de informações significativas para o seu processamento.

Com relação à **exibição**, os monitores são os principais dispositivos de saída usados em sistemas de processamentos de imagens. Também pode haver outros dispositivos de saída tais como as impressoras (*laser*, jato de tinta e matricial). Mas os monitores coloridos ou monocromáticos são os principais e os mais utilizados para esse fim. Todos os elementos podem ser representados conforme a figura 2.3.

## 2.3 Limites e Dificuldades

Os sistemas de processamento de imagens podem ter alguns limites e dificuldades para a visão artificial, tais como: (1) a posição dos objetos, (2) diferenças de tamanhos e formatos, (3) problemas de perspectivas e (4) iluminação.

### 2.3.1 Posição

O posicionamento dos objetos pode ser um fator que atrapalha bastante a visão artificial. A figura 2.4 mostra um mesmo objeto posicionado de maneiras diferentes. A visão natural consegue reconhecer que esses objetos são os mesmos, mesmo estando em posições diferentes, o que não acontece com a visão artificial. Para que um mesmo objeto em posições distintas seja reconhecido pela visão artificial é necessário extrair

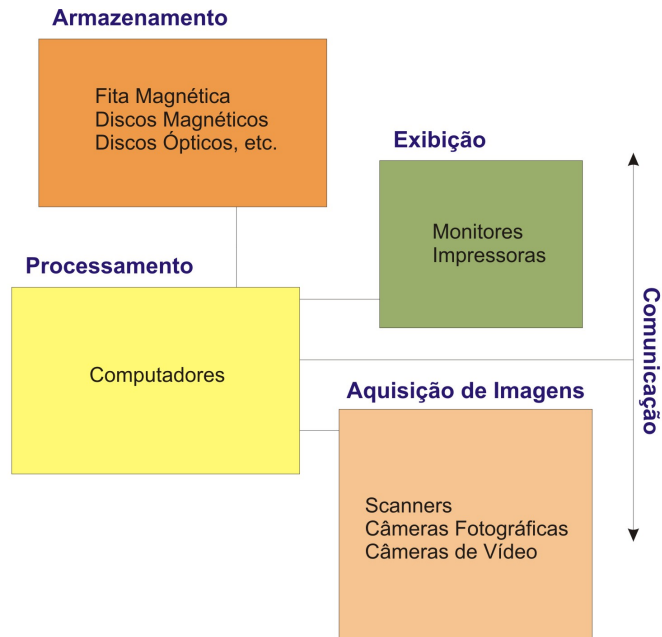


Figura 2.3: Elementos funcionais básicos de um sistema de processamento de imagens: Armazenamento, Processamento, Aquisição, Exibição e Comunicação.

indícios e primitivas que não dependem da posição do objeto. Esses indícios são chamados de *invariantes*.

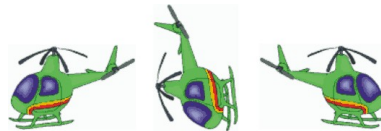


Figura 2.4: Posição: O mesmo objeto em posições diferentes.

### 2.3.2 Tamanho e Formato

O tamanho dos objetos também pode ser um problema para a visão artificial. Ou seja, reconhecer um objeto, ainda que seja o mesmo, mas de tamanhos diferentes, torna-se uma tarefa bastante difícil quando não realizada pela visão natural. Por exemplo, um processo de visão artificial dificilmente reconheceria que os elementos da figura 2.5 são os mesmos, pois suas dimensões são diferentes.

A mesma dificuldade aconteceria ao tentar reconhecer que os elementos da figura 2.6 pertencem à mesma classe. Ou seja, estes elementos, apesar de possuírem algumas características em comum, para a visão artificial eles são objetos bastante diferentes, inclusive quanto à sua classe.

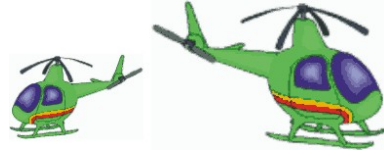


Figura 2.5: Tamanho: O mesmo objeto em tamanhos diferentes.



Figura 2.6: Formato: Objetos diferentes pertencentes à mesma classe.

### 2.3.3 Perspectiva

Outro problema comum é a perspectiva de uma imagem. Na figura 2.7, a visão natural consegue perceber facilmente que os objetos são os mesmos, e estão dispostos em uma superfície plana, onde o mais próximo está parcialmente escondendo o segundo. Pode-se perceber também que os dois não se tocam, mas para a visão artificial este parecer não seria tão simples e a disposição dos objetos, mesmo em distâncias diferentes, não seria tão perceptível.



Figura 2.7: Perspectiva: Dificuldades para a visão artificial compreender a imagem.

### 2.3.4 Iluminação

A visão natural possui uma capacidade enorme de se adaptar às variações de luz, capacidade esta que se torna uma problemática para a visão artificial. A iluminação é um dos pontos mais críticos, tornando-se um fator de extrema importância nessa área. Quando um mesmo objeto é disposto sob iluminações diferentes, é extremamente trabalhoso para a visão artificial reconhecer que eles são iguais, pois as tonalidades de cor dos pixels da imagem se apresentam diferentes. Para exemplificar esse problema, a figura 2.8, contendo o mesmo objeto sob diferentes iluminações, mostra que, para a visão artificial, as quatro imagens podem ser distintas, enquanto que para a visão natural os objetos são os mesmos.



Figura 2.8: Iluminação: O mesmo objeto com diferentes iluminações.

## 2.4 Áreas Relacionadas

No processamento de imagens, a visão artificial possui um relacionamento muito grande com diversas áreas. Na **inteligência artificial**, ela é destinada ao planejamento e elaboração de sistemas que trabalham com ações mecânicas, como a movimentação de um robô em determinado ambiente. Esse tipo de processamento necessita de dados provenientes destes sistemas, que agem como um sensor visual e provêem informações confiáveis sobre o ambiente para o robô. Outra aplicação da inteligência artificial, em que é utilizada a visão artificial, é o reconhecimento de padrões e técnicas de aprendizado de máquinas. Como consequência disso, a visão artificial é considerada parte da inteligência artificial.

Outro campo bastante relacionado com a visão artificial é a **física**, pois a maior parte das aplicações dessa área necessita trabalhar com as radiações eletromagnéticas que são refletidas nos objetos e capturadas pelos *sensores* de imagem para produzir as suas informações.

Como a inteligência artificial e a física, a **neurobiologia**, que estuda o sistema visual biológico, também é outro campo que se relaciona com a visão artificial. Nestes últimos tempos, muito se tem estudado sobre os olhos, os neurônios e a estrutura do cérebro. Essas pesquisas têm como finalidade desenvolver uma visão artificial que resolva alguns problemas que não são resolvidos pela visão natural.

## 2.5 Considerações Finais

Os sistemas de visão artificial, apesar de grande evolução da tecnologia nos últimos anos, não demonstram a mesma eficiência conseguida pelos sistemas de visão natural. Principalmente no que se refere ao processamento de imagens coloridas muito ainda



se deve fazer para aumentar a eficiência dos algoritmos de visão computacional.

# Capítulo 3

## Modelos de Cores

### 3.1 O Olho Humano e a Interpretação de Cores

A cor é um importante fator no processamento de imagens. Ela descreve de forma influente um objeto ou uma cena, simplificando e identificando as características dos mesmos. Da mesma forma que, em análise de imagens realizadas por seres humanos, a motivação para o uso de cores é que enquanto a visão natural (ou o olho humano) pode comparar cerca de duas dúzias de tons de cinza, ele também pode conhecer, de forma distinta, milhares de tons e intensidades de cores.

Até recentemente, o processamento de imagens coloridas era realizado com pseudo cores. Com o progresso da tecnologia ocorrido nos últimos anos, houve uma disponibilidade maior de equipamentos com cores verdadeiras (*True Color*) a um preço significativo. Como resultado disto, técnicas de processamento de imagens baseadas em cores, cada vez mais reais, estão se tornando expressivas em grande escala de aplicações.

### 3.2 Fundamentos de cores

Quando o feixe de luz solar passa através de um prisma de vidro, um espectro contínuo de cores, variando do violeta ao vermelho, é gerado ao invés do branco. Este espectro de cores pode ser dividido em seis regiões: **violeta**, **azul**, **verde**, **amarelo**, **laranja** e **vermelho** (Figura 3.1). Cada uma destas regiões (ou cada uma destas cores) mistura-se com as demais de forma suave, não terminando de forma repentina. Assim, basicamente, as cores são determinadas pela natureza da luz que reflete no objeto.

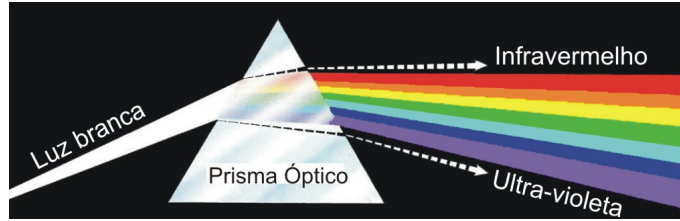


Figura 3.1: Espectro de cores visto quando a luz branca passa pelo prisma.

### 3.2.1 Luz cromática e acromática

A luz, que é essencial para a ciência das cores, pode ser **acromática** ou **cromática**. A luz acromática consiste na luz sem cores e o seu único atributo será sua **intensidade** de luz, onde o termo *nível de cinza* refere-se a essa medida de intensidade variando de forma escalar do preto aos cinzas, e ao branco. A luz cromática, ao contrário da luz acromática, possui três atributos que descrevem sua qualidade: (1) a **Radiância**, que é quantidade total de energia que flui de uma fonte de luz; (2) a **Luminância**, que é a medida da quantidade de energia que um observador percebe da fonte de luz; e (3) o **Brilho**, que incorpora a noção acromática de intensidade. Apesar de ser praticamente impossível de se medir o brilho, este é um fator importante na descrição da sensação de cores (GONZALES E WOODS, 1992).

### 3.2.2 Cores Primárias e Secundárias

”Devido à estrutura do olho humano, todas as cores são vistas como combinações variáveis das três chamadas **cores primárias**: **vermelho** (**R**, do inglês *”red”*), **verde** (**G**, do inglês *”green”*), e **azul** (**B**, do inglês *”blue”*) (Figura 3.2). Para o propósito de padronização, a Comissão Internacional de Iluminação (CIE - *”Comission Internationale de l’Eclairage”*) designou em 1931 os seguintes valores de comprimento de onda específicos para as três cores primárias medidos em  $nm$ <sup>1</sup>: azul = 435,8 nm, verde = 546,1 nm, e vermelho = 700 nm”.

A mistura entre as cores primárias gera as **cores secundárias**: O **Ciano**, que é a mistura das cores verde e azul; o **Magenta**, que é a mistura das cores vermelho e azul; e **Amarelo**, que é a mistura das cores vermelho e verde (Figura 3.2). Uma das possibilidades de se obter a cor branca é pela mistura, em intensidades máximas, das três cores primárias.

<sup>1</sup>O **Nanômetro** ( $nm$ ) é uma unidade de comprimento definida pelo sistema internacional de unidades e usada, dentre outras coisas, para medição de comprimentos de onda de luz visível (400 nm a 700 nm). Um  $nm$  equivale a um milionésimo de milímetros ou  $1,0 \times 10^{-9}$  metros.

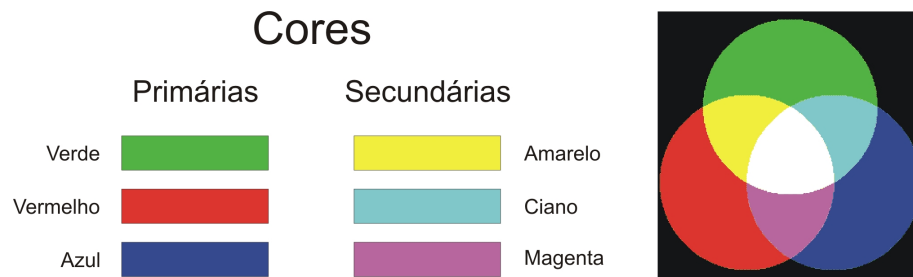


Figura 3.2: Cores Primárias e Secundárias.

### 3.2.3 Distinção de Cores

”A distinção entre as cores primárias de luz e as cores primárias (subtrativas) de pigmentos ou corantes é importante. No último caso, uma cor primária é definida como sendo aquela que subtrai ou absorve uma cor primária da luz e reflete ou transmite as outras duas. Portanto, as cores primárias dos pigmentos são magenta, ciano e amarelo, e as cores secundárias são vermelho, verde e azul”.

Para distinguirmos uma cor da outra, utilizamos as seguintes características: O (1) **brilho**, que abrange a noção cromática da intensidade de luz; a (2) **matiz**, que é a cor dominante ou a cor percebida pelo observador; e a (3) **saturação**, que se refere à pureza ou à quantidade de luz branca mesclada com a matiz. A saturação refere-se às cores puras do espectro, ou seja, cores como o lilás são menos saturadas devido à mistura das cores violeta e branco.

Quando medimos a matiz e saturação em conjunto, chamamos isso de cromaticidade. E uma outra forma de especificar cores é o *diagrama de cromaticidade* (figura 3.3), que é útil para a mistura de cores pois ”um segmento de reta juntando quaisquer dois pontos do diagrama define todas as variações de cores diferentes que podem ser obtidas através da combinação aditiva destas duas cores” (GONZALES E WOODS, 1992).

## 3.3 Modelos de Cores

### 3.3.1 Modelo RGB

Baseado num sistema de coordenadas cartesianas, esse modelo é um dos sistemas de cores mais utilizados. O subespaço de cores deste modelo é um cubo (figura 3.4) onde os valores das cores primárias **vermelho**, **verde** e **azul** (RGB) estão dispostos em três vértices; os valores das cores secundárias **ciano**, **magenta** e **amarelo** estão dispostos em outros três vértices; o **preto** está no vértice de origem e o **branco** está

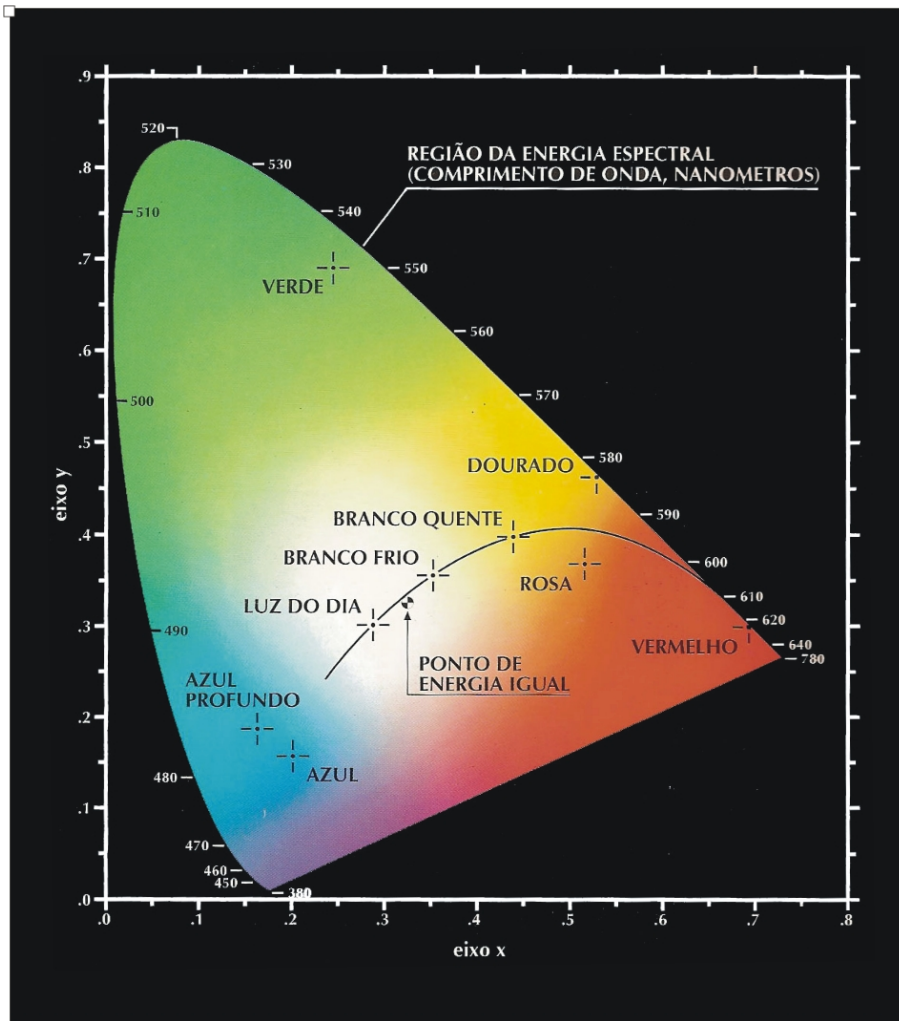


Figura 3.3: Diagrama de Cromaticidade. Fonte: (GONZALES E WOODS, 1992).

no vértice mais distante da origem.

As cores no cubo RGB são especificadas por valores que podem variar entre 0 e 1, sendo que a relação desses valores representam também seu posicionamento (ou a coordenada) dentro do cubo relativo às retas  $(x,y,z)$ . Ou seja, a cor *preta* por estar na origem, é representada pela coordenada  $(0,0,0)$ , e a cor *branca* que está no vértice oposto à cor *preta*, é representada pela coordenada  $(1,1,1)$ . As demais cores são formadas pela relação (ou mistura) dos valores entre 0 e 1 das cores que as compõem como a cor *ciano*, por exemplo, que tem coordenadas  $(1,1,0)$ . Ou seja, é obtida pelas luzes *azul* e *verde*, em proporções iguais, sem a mistura do *vermelho*. No sistema RGB, os tons de *cinza* possuem pontos com os mesmo valores em **R**, **G** e **B** e vão desde a origem (*preto*) até o ponto mais distante (*branco*). Assim, pode-se formar a cor desejada partindo deste princípio, como mostra a figura 3.5.

O número de bits utilizado para representar os pixels (picture element) no sistema RGB é chamado de *pixel depth* (profundidade do ponto). Como exemplo,

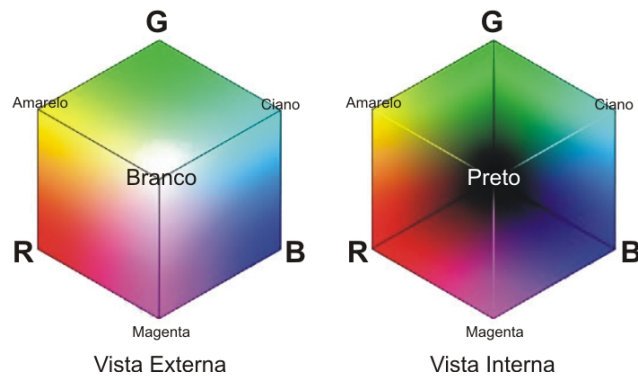


Figura 3.4: Representação do Modelo através do cubo RGB.

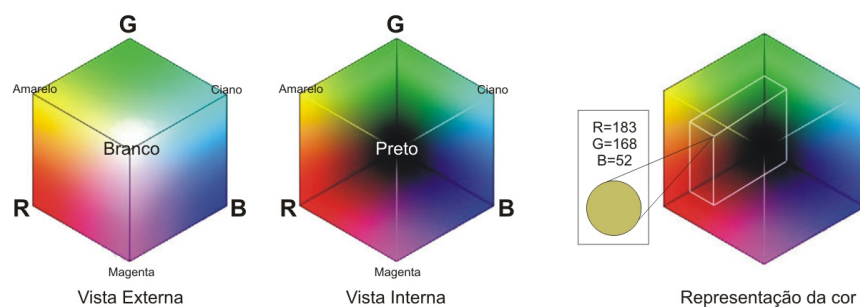


Figura 3.5: Representação e localização de uma cor específica no cubo RGB.

suponha-se que uma imagem possua as cores *vermelha*, *verde* e *azul* com 8 bits. Nesse contexto, cada pixel (**R**, **G**, **B**) da imagem possui 24 bits (três canais de cores). Quando uma imagem RGB possui 24 bits ela é chamada de imagem *full-color* (todas as cores). O número total de cores em uma imagem de 24 bits é  $(2^8)^3 = 16,777,216$  milhões de cores.

### 3.3.2 Modelo CMYK

As cores secundárias também são conhecidas como cores subtrativas. Alguns dispositivos, tais como as impressoras coloridas que requerem dados CMY de entrada, convertem de RGB para CMY antes de imprimirem. O sistema CMY, que significa: **ciano** (**C**, do inglês "cyan"), **magenta** (**M**, do inglês "magenta") e o **amarelo** (**Y**, do inglês "yellow"), representa a cor não absorvida de cada pigmento. Como a mistura das cores *ciano*, *magenta* e *amarelo* resulta numa tonalidade de *marrom* ou *cinza*, essa deficiência de cor é compensada pelo acréscimo da tinta *preta*, como quarto pigmento básico a essa mistura, tornando assim o **CMYK** (abrangendo as cores: *Cyan*, *Magenta*, *Yellow* e *Black*).

Como a letra **B** poderia ser confundida com *Blue*, a letra **K** foi adotada re-

presentando a cor preta *Black*. Existem muitas teorias para a adoção dessa letra e a mais coerente é o fato de, caso as três cores CMY se unam, esta geraria uma cor chave (do inglês **Key**). Outro motivo do acréscimo da cor preta está relacionado ao seu custo, pois como é a cor mais utilizada, evita-se a utilização das demais cores para gerá-la. Fraser et al. (2005).

Como pode ser visto na equação 3.1 "a luz refletida de uma superfície coberta com *ciano* puro não contém *vermelho* (isto é,  $C=1-R$ ); do mesmo modo, *magenta* pura não reflete *verde*; e *amarelo* puro não reflete *azul*. A equação revela também que os valores RGB podem ser obtidos facilmente a partir de um conjunto de valores CMY, através da subtração de valores CMY individuais de 1" (GONZALES E WOODS, 2001).

$$\begin{bmatrix} C \\ M \\ Y \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (3.1)$$

### 3.3.3 Modelo HSV e HSI

O modelo HSV, também conhecido como modelo HSB ou HSI, é um modelo de representação matemática das cores, baseando-se na maneira mais próxima da percepção humana, enquanto que os modelos RGB e CMYK baseiam-se nas técnicas físicas de reprodução de cor, respectivamente, por emissão ou absorção de luz.

O modelo HSV é ilustrado graficamente por um cone (figura 3.6) em que cada parte do cone representa os componentes de cor utilizados nesse sistema. Assim, os componentes de cor são: (1) o **matiz** (**H**, do inglês "hue"), a **saturação** (**S**, do inglês "saturation") e a **luminância** (**V**, do inglês "value").

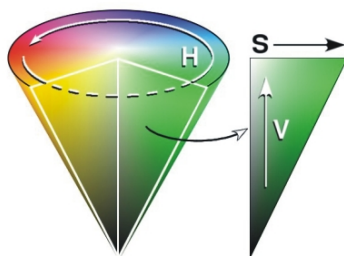


Figura 3.6: O cone representando as cores do modelo HSV.

- **Hue:** É o matiz de cores, ou seja, são todas as cores principais do espectro de cores visível. São identificadas (ou estão representadas) na base do cone

correspondendo às arestas do cone que variam de 0 a 360 graus. Ou seja, seu valor varia entre o *vermelho* (ângulo 0) passando pelo *amarelo* (ângulo 60), *verde* (ângulo 120), *ciano* (ângulo 180), *azul* (ângulo 240), *magenta* (ângulo 300) e novamente o *vermelho* (ângulo 360).

- **Saturation:** É a saturação ou o elemento que dá vida às cores, adicionando ou subtraindo o fator "cor" dos tons. Está relacionada ao quanto de cor *branca* está sendo utilizado (com esta variando o seu valor de 0 a 1) indicando assim a pureza de um determinado matiz. Quanto mais próxima do contorno do cone o valor da saturação estiver, mais pura é a matiz, ou seja, quanto mais próxima do valor 1, mais viva é a matiz. Ao contrário disso, quanto mais próxima do centro do cone o valor da saturação estiver, mais impura é a matiz, ou seja, mais próxima do valor 0, significando que a cor é totalmente *branca* e, neste caso, não importa muito o valor da matiz.
- **Value:** É o valor percentual de iluminação geral da cor, correspondendo à intensidade da cor. Está relacionado ao quanto de cor *preta* está sendo utilizado (com esta variando o seu valor também de 0 a 1) indicando assim a pureza de um determinado matiz e da sua saturação. Quanto mais próxima da base do cone o valor da luminância estiver, maiores serão os valores do matiz e da saturação, ou seja, quanto mais próxima do valor 1, maiores serão as intensidades das cores. Ao contrário disso, quanto mais próxima do pico do cone o valor da luminância estiver, menores serão os valores do matiz e da saturação, ou seja, quanto mais próxima do valor 0, menores serão as intensidades de cores. As cores claras e escuras são, respectivamente, bem iluminadas e pouco iluminadas. Caso o valor da saturação seja 0 (mais próxima do centro do cone), a iluminação criará tons de *cinza*, *preto* e *branco*. E caso a cor seja bem saturada, a iluminação se encarrega de escurecer ou clarear a cor, gerando os tons.

Pode-se representar essas características do cone na figura 3.7.

### 3.3.4 Outros Modelos de Cor

No modelo de cor **YCbCr**, a luminância ou luminosidade da cor é representada pela componente Y (que poder ser vista como um tom de cinza) enquanto a crominância (matiz e saturação indicando o tom da cor) é representada pelas componentes Cb e Cr. O Cb coloca a cor numa escala entre o azul e o amarelo e o Cr indica o local da cor entre vermelho e verde.





Tonalidade

Brilho / Luminosidade

Saturação

Figura 3.7: O modelo HSV e a representação das suas características. Fonte: (LEÃO ET AL., 2005).

Semelhante ao modelo YCbCr, o modelo de cor YIQ possui a componente **Y** indicando a luminância ou a luminosidade da cor. Já as componentes **I** e **Q** indicam a cromaticidade.

Criado pela *Commission International de l'Eclairage*, o modelo de cores CIELAB possui a componente **L** indicando a luminosidade variando de 0 (preto) a 100 (branco). As outras 2 componentes **a\*** e **b\*** indicam respectivamente as cores vermelho (variando até o verde) e amarelo (variando até o azul) (SHARMA, 2003).

### 3.4 Considerações Finais

Os modelos de cores são empregados na tentativa de representar de forma matemática as cores existentes, oriundas de reflexões da luz sobre os objetos. A diversidade de modelos propostos (RGB, HSV, HSI, CieLab, YCbCr, etc.) objetivam a dedicação da aplicação. Assim, o RGB é mais apropriado para definir cores em equipamentos de visualização como os monitores de vídeo. O HSV e HSI tentam interpretar o processo de percepção humana das cores. Na segmentação de imagens coloridas, no entanto, não se pode afirmar que este ou aquele modelo seja mais eficiente.

## Capítulo 4

# Segmentação da Pele Humana em Imagens Coloridas

### 4.1 Métodos para Segmentação

Para o processamento e análise de imagens, geralmente o primeiro passo é a sua segmentação. "A segmentação subdivide uma imagem em suas partes ou objetos constituintes. O nível a que essa subdivisão deve ser realizada depende do problema a ser resolvido. Ou seja, a segmentação deve parar quando os objetos de interesse na aplicação tiverem sido isolados" (GONZALES E WOODS, 1992).

Os algoritmos de segmentação para imagens monocromáticas são geralmente baseados em uma das seguintes propriedades básicas de valores de níveis de cinza: **descontinuidade** e **similaridade**. Enquanto a **descontinuidade** consiste no particionamento da imagem baseando-se na mudança repentina dos tons de cinza dos *pixels* (tendo como propósito detectar pontos isolados e linhas e bordas nas imagens) a **similaridade** consiste no particionamento, baseando-se na classificação dos pixels que possui os tons de cinza com as mesmas características ou com as características mais próximas uns dos outros como a textura ou a cor do pixel.

Em uma imagem colorida, a segmentação realizada por meio da descontinuidade e/ou da similaridade sofre o mesmo processo das imagens em tons de cinza. A diferença é que esses processos em imagens coloridas são baseados conforme um modelo de cor específico. Assim, se o modelo de cor da imagem for o RGB, utiliza-se o conjunto desses valores para o mesmo pixel (vermelho  $\mathbf{R}(x,y)$ , verde  $\mathbf{G}(x,y)$  e azul  $\mathbf{B}(x,y)$ ).

## 4.2 Algoritmos Baseados em Vizinhança

O **crescimento** da região é um dos algoritmos por **similaridade** utilizados para segmentar imagens. Esse crescimento em torno de um pixel central (ou pixel principal, denominado também de pixel "semente") dentro de um domínio espacial, constitui-se de um processo que apura as características de um determinado número de pixels vizinhos a este pixel principal. E esse conjunto, ou essa região de pixels, em torno do pixel principal (figura 4.1), é chamado de **vizinhança**.

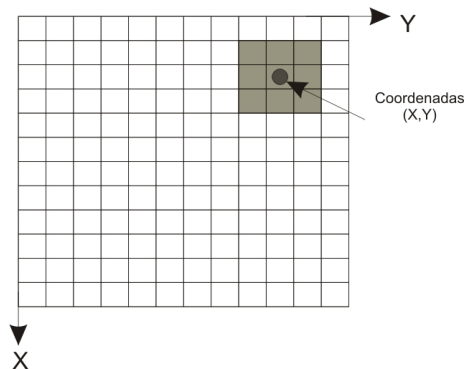


Figura 4.1: Exemplo de vizinhança 3x3.

Assim, a vizinhança de um pixel é determinada em uma **subimagem** pertencente a um domínio espacial. Ou seja, esta subimagem é composta por um determinado número de pixels onde serão realizados os processamentos de extração das suas características e verificados a pertinência do pixel "semente" dentro do que se pretende segmentar. "O centro da subimagem é movido de pixel a pixel iniciando-se no canto superior esquerdo e aplicando-se o operador para cada posição (x,y) para obter  $g$  naquela posição" (GONZALES E WOODS, 1992).

Existem diversos tamanhos de subimagens, o que determina também o tamanho da sua vizinhança. As mais utilizadas possuem tamanhos regulares pois, embora outras formas de vizinhança, tal como aproximações de um círculo, sejam às vezes utilizadas, matrizes retangulares e quadradas são bem mais comuns devido à facilidade de implementação.

## 4.3 A Segmentação de Pele Humana

A segmentação de pele humana representa um importante papel em inúmeras aplicações de processamento de imagens, abrangendo análise de gestos, detecção de faces ou partes do corpo e diversas áreas de interação entre homem-computador. A pele humana é composta pelas camadas **epiderme**, **derme** e **hipoderme**, e conforme

ela é exposta aos raios luminosos suas cores são alteradas devido às características de filtragem de luz realizadas pelas camadas da pele.

Dessa forma, a segmentação de cor de pele em imagens pode se tornar uma tarefa extremamente complexa e desafiadora, visto que esta cor em uma determinada imagem é bastante sensível a diversos fatores, tais como:

- **Iluminação:** Sua variação é um dos maiores problemas relacionados ao desempenho de segmentação de pele humana pois produz características inconstantes da cor de pele.
- **Etnia:** Pessoas de grupos étnicos diferentes (como, por exemplo: Caucasianos, Asiáticos, e Africanos) possuem diferentes cores de pele, caracterizados por brancos, amarelos e negros.
- **Características individuais:** A idade, o sexo e regiões do corpo também afetam a característica da cor da pele.
- **Câmera Fotográfica:** A cor de pele pode diferenciar-se quando a imagem da mesma pessoa é obtida por câmeras diferentes devido às características do sensor do equipamento.
- **Outros fatores:** As cores de fundo da imagem, as sombras contidas na imagem e, principalmente, a aparência da pessoa (ocasionadas por: maquiagem, cabelo, óculos, etc.), também influenciam na característica da cor de pele.

O fato de escolher uma técnica (ou método) de segmentação, em relação a uma outra, está diretamente relacionado ao problema que se pretende resolver. E para a segmentação de pele humana, um dos principais desafios é fazer com que a classificação dos pixels coloridos das imagens seja o mais similar possível àquela percebida pela visão natural.

## 4.4 Escolha do Modelo de Cor

O primeiro passo, talvez o principal, para a classificação de cor de pele em imagens, é a escolha de um modelo de cor. Foi observado que a pele contém um nível de cor vermelha significativa. Assim, alguns trabalhos tentam utilizar esta característica para um resultado mais satisfatório.

Geralmente o modelo RGB é o modelo escolhido pela grande maioria, mas qualquer outro espaço pode ser obtido a partir de uma transformação linear ou não linear do RGB para a detecção de pele. Dessa forma, alguns trabalhos utilizam,

por exemplo, o modelo *rgb Normalizado*, ao invés do modelo de cores RGB, por mostrar-se mais adequado para a segmentação de pele. Um dos principais motivos da utilização do modelo *rgb Normalizado* é que este reduz as diferenças nos pixels com cor de pele, relativos às condições de iluminação e às diferenças étnicas. Por estas vantagens, inúmeros trabalhos foram desenvolvidos com este modelo de cores. (YANG ET AL., 1998), (YANG E AHUJA, 1999), (BERGASA ET AL., 2000), (BROWN ET AL., 2001), (CAETANO E BARONE, 2001), (OLIVER ET AL., 1997), (KIM ET AL., 1998), (SCHWERDT E CROWELY, 2000), (SEBE ET AL., 2004), (SORIANO ET AL., 2003), (STÖRRING ET AL., 2003) e (WANG E SUNG, 1999). "Além disso, os agrupamentos de cor de pele no espaço *rgb Normalizado* apresentam uma variância relativamente menor do que os grupos correspondentes no RGB e, portanto, mostra-se adequado para a detecção e modelagem de cor de pele" (KAKUMANU ET AL., 2006).

O modelo RGB não representa diretamente as características perceptivas de cores como no modelo HSV ou HSI. Assim, estes modelos de cores podem formar uma boa escolha para métodos de detecção de pele, pois não variam com alta intensidade de luz branca ou luz ambiente, por exemplo. Inúmeros trabalhos utilizam o modelo HSV para segmentação de pele, como: Brown et al. (2001), Garcia e Tziritas (1999), McKenna et al. (1998), Saxe e Foulds (1996), Sobottka e Pitas (1998), Thu et al. (2002), Wang e Yuan (2001) e Zhu et al. (2004).

Quando os componentes de *luminância*<sup>1</sup> e *crominância*<sup>2</sup> são explicitamente separados, como no modelo de cores YCbCr, eles se tornam favoráveis para detecção de pele. O modelo de cores YCbCr, por exemplo, representa a cor, como luminância, em Y e a crominância em Cb e Cr. A cor neste modelo é obtida pela soma ponderada dos valores RGB e a crominância é obtida através da subtração da componente luminância dos valores B e R. O espaço YCbCr é uma das escolhas mais populares em detecção de pele e foi usado em diversos trabalhos, como: Hsu et al. (2002), Chai e Bouzerdoum (2000), D. Chai (1998), K.W.Wong et al. (2003).

A forma como duas cores diferem em aparência, para um observador humano, é representada pela uniformidade perceptiva. Assim, espaços de cor uniformes foram definidos de maneira que todas as cores são organizadas por suas diferenças perceptivas, mas isto é obtido a altos custos de transformações computacionais. Os modelos de cores CIE-Lab e CIE-Luv representam a luminância em L, e a crominância, representada em ab ou uv, é obtida através de mapeamento das coordenadas XYZ. Para a segmentação de pele, o modelo CIE-Lab foi usado nos trabalhos de Cai e Goshtasby (1999) e Kawato e Ohya (2000). Já o modelo CIE-Luv foi usado

---

<sup>1</sup>**Luminância** refere-se ao valor da intensidade luminosa emitida por uma superfície, ou seja, as luzes branca e preta.

<sup>2</sup>**Crominância** refere-se ao valor das cores, ou seja, é o conjunto de cores combinadas.

no trabalho de Yang e Ahuja (1999).

## 4.5 Métodos de Classificação de Pixel de Cor de Pele

A detecção de pele humana em imagens pode ser dividida em duas classes: (1) pixel-pele e (2) pixel-não-pele; e para abordar este problema, pesquisadores utilizam diferentes técnicas procurando obter um melhor resultado.

O **thresholding**, por exemplo, do espaço de cor de pele agrupa uma pequena região de pixels pertencentes a um determinado modelo de cor considerando que as imagens foram obtidas em ambientes de iluminação controlada. Esse é um dos métodos mais fáceis e mais utilizados na classificação de pixels de cor de pele.

No trabalho desenvolvido por Dai e Nakano (1996), por exemplo, foram utilizados valores fixos na componente I ( $RI = 0,50$ ) do modelo de cor YIQ para detectar pixels-pele em imagens que continham pessoas com pele amarela. Em Wang e Yuan (2001) foram utilizados valores fixos na componente  $rg$  do modelo de cores  $rgb$  Normalizado ( $r = 0.36$  a  $0.46$ ,  $g = 0.28$  a  $0.36$ ) e também valores na componente HSV ( $H = 0,50$ ,  $S = 0.20$  a  $0.68$  e  $V = 0.35$  a  $1.0$ ). Os trabalhos de Sobottka e Pitas (1998) e Sobottka e Pitas (1996) também utilizaram o modelo de cores HSV com valores fixos na componente HS deste modelo ( $H = 0,50$  e  $S = 0.23$  a  $0.68$ ) para detectar pixel-pele em imagens que continham pessoas com pele amarela e branca. Em Chai e Ngan (1999) foram utilizados valores fixos na componente CbCr do modelo de cores YCbCr ( $Cb = 77,127$  e  $Cr = 133,173$ ) para detectar pixel-pele com o intuito de segmentar a face humana.

O **histograma** também é outro método usado para representar a distribuição dos tons de pele em um espaço de cor quantizado. Existem diversos trabalhos que utilizam o histograma para segmentação de pele, em que estes procuram realizar esta segmentação dividindo estes espaços em duas classes de pixels (pixels-pele/pixels-não-pele) através de características próprias destes elementos. Cada elemento deste espaço associa a ocorrência da cor deste elemento ao conjunto de dados de treinamento verificando se o pixel é ou não é pele. O uso desse método pode ser encontrado em diversos trabalhos, como: Schwerdt e Crowley (2000), Zarit et al. (1999), Chai e Bouzerdoun (2000), Gomez et al. (2002), Brand e Mason (2000), Gomez e Morales (2002), Fleck et al. (1996), Marcel e Bengio (2002), Srisuk e Kurutach (2002) e Sigal et al. (2004). Este método pode ser usado para construir um classificador de peles simples, relativamente preciso e computacionalmente rápido, mesmo que as imagens sejam coletadas de ambientes sem restrição (como as imagens da web).

Um exemplo que se pode citar é o trabalho desenvolvido por Jones e Rehg (2002) em que foi construído um histograma no modelo de cores RGB com 2 bilhões de pixels coletados em cerca de 18.000 imagens da web. Destes 2 bilhões de pixels, cerca de 10 por cento deles eram pixels-pele, havendo uma forte tendência dos seus valores em estar próximo ao vértice vermelho do cubo de cor, demonstrando que existe uma separação razoável entre as duas classes.

As misturas **Gaussianas** também são largamente usadas para classificar pele humana por causa da vantagem que esses modelos têm em generalizar melhor com menos dados de treinamento, e também pelo fato de utilizar menos requisitos de memória. As cores de pele, de diferentes raças e indivíduos, se agrupam em uma pequena região do espaço de cor em condições controladas de iluminação. E a classificação destas cores pode ser modelada por uma distribuição normal multivariada (Gaussiana) dentro deste espaço de cor normalizado, conforme citado em Yang et al. (1998) e J. Yang e A. Waibel (1996). Apesar dessas cores se agruparem, existem diferentes classificações dentro desse grupo e não podem, portanto, serem definidas por uma única distribuição Gaussiana. Por isso, muitos pesquisadores têm utilizado uma quantidade  $N$  de componentes Gaussiana para classificar estes grupos, com  $N$  variando entre 2 e 16.

Em Greenspan et al. (2001) foram usadas 2 Gaussianas para mostrar como ela pode ser uma representação robusta ao classificar cor de pele normal e cor de pele com maior brilho. Em Caetano et al. (2003) usou-se 2 Gaussianas para classificar pessoas brancas e negras. E em Thu et al. (2002) foram usadas 4 Gaussianas para classificar partes da face iluminada de diferentes maneiras. Em Yang e Ahuja (1999) foram usadas 2 Gaussianas sobre a base de dados de face Michigan para classificar pele humana, enquanto que em Lee e Yoo (2002) foram usadas 6 Gaussianas na base de dados da Compaq, também com o propósito de classificar pele humana. Outros trabalhos, como Oliver et al. (1997), McKenna et al. (1998) e Jones e Rehg (1999), também utilizaram a Gaussiana para classificar pele humana. Em Jones e Rehg (2002) foram utilizadas 16 Gaussianas em cada um dos 2 modelos de treinamento, para classificar pele e não pele.

As **Redes Neurais** (como a *MLP - MultiLayer Perceptron* e a *SOM - Self-Organizing Map*) também são outros métodos bastante utilizados para classificar pele e não pele. A quantidade de neurônios que compõem a rede, as taxas de aprendizado, o número de camadas e outros fatores, é que determina (ou não) o sucesso da classificação desejada. O aprendizado da rede neural é feito com a apresentação de padrões que servirão para o treinamento da rede. Durante este treinamento, os pesos sinápticos são ajustados e, depois de ajustados, eles servirão para ajudar na classificação desejada. Dessa forma, as redes neurais classificam as regiões da

imagem como sendo regiões de pele ou não-pele, conforme o treinamento que tiveram para fazer esta classificação. Em sistemas de reconhecimento de padrões, como uma rede neural, torna-se bastante significativa a coleta de amostras para classificar pele e não-pele, pois o treinamento destas redes é realizado com amostras de pele e amostras de não-pele.

Os trabalhos Chen e Chiang (1997), Karlekar e Desai (1999), Phung et al. (2001), Sahbi e Boujemaa (2002) e Seow et al. (2003) usaram uma Rede Neural do tipo MLP com três camadas de neurônios, mas em modelos de cores diferentes. No trabalho de Chen e Chiang (1997) foi utilizado o modelo de cor CIE-Lab. Em Karlekar e Desai (1999) e Phung et al. (2001) o modelo utilizado foi o YCbCr. Em Sahbi e Boujemaa (2002) e Seow et al. (2003) o modelo utilizado foi o RGB. Em Seow et al. (2003) o treinamento da rede consistia na extração das regiões de pele e da interpolação destes valores em um cubo de cor em 3D, definindo o restante como não sendo pele. Já no trabalho Brown et al. (2001) foram usadas duas redes SOMs. Uma rede foi treinada para aprender o que era pixels cor de pele e outra para aprender o que era pixels cor de não-pele utilizando-se mais de 500 imagens.

## 4.6 Considerações Finais

Com relação aos modelos de cores, pode-se dizer que são muitos os modelos usados para detectar pele humana. Em alguns, como o RGB, por exemplo, percebeu-se que a capacidade de separar pixels em grupos de pele e não-pele é maior que em outros. Já no HSV ou HSI, por exemplo, foi reduzida a intensidade de cor de forma a fornecer parâmetros mais precisos em diversas condições de iluminação.

Em muitos trabalhos, a escolha dos modelos de cores mais apropriados para a classificação de pixel de pele foi uma tarefa bastante difícil, por depender de vários fatores, como por exemplo, o classificador e a iluminação.

Alguns trabalhos, no entanto, que utilizaram uma ou mais Redes Neurais como classificadores de pele mostraram que eram indiferentes à escolha do modelo. Embora o modelo não fosse fundamental, quando utilizada uma Rede Neural como classificador, outros trabalhos mostraram que para cada modelo de cor existe um classificador de pele que pode obter desempenho distinto.

Outros trabalhos que utilizaram Histograma e Gaussiana como classificadores de pele compararam os resultados utilizando diferentes modelos de cor. E estes mostraram que a escolha de um determinado modelo de cor faz muita diferença, e a transformação de um modelo de cor para outro melhora o desempenho de classificação.



Um outro fator importante é que muitos dos trabalhos desenvolvidos e muitas das comparações realizadas foram feitas utilizando um conjunto de dados reduzido. Para conferir os modelos de cor baseados na sua capacidade de classificar pixels de pele e não-pele, os conjuntos de dados a serem utilizados devem possuir uma quantidade significativa de amostras destas classes, incluindo as amostras de pele de um grande número de pessoas de diferentes raças, com poucas restrições nas suas aparências e em diversas condições de iluminação.

Com relação aos classificadores de pixels, pode-se concluir que a capacidade de detectar diferentes tipos de pele humana (branca, amarela, escura, etc.) é que determina se o classificador utilizado é ou não adequado. Ou seja, para que se possa dizer que o classificador é bom, este deve detectar com a máxima precisão a maior variedade possível de pele humana em imagens, inclusive sob diferentes condições de iluminação. Um critério bastante importante para avaliar o desempenho de um classificador de pele é a definição dos grupos de pele e não-pele em um determinado espaço dentro de um modelo de cor.

Alguns dos problemas em avaliar estes classificadores é que muitos deles, utilizados em diferentes técnicas, consideram apenas alguns poucos tipos de pele e em condições restritas de iluminação. Um outro fator importante é que a quantidade de modelos (ou padrões) usados nos treinamentos e nos testes é bastante limitada, reduzindo significativamente seu desempenho.

Para definir qualquer um dos classificadores como sendo este melhor ou pior que outros, é preciso considerar fatores como o tamanho dos conjuntos de treinamento e teste utilizados, visto que o tamanho destes elementos e a quantidade e variedade de amostras que formam este conjunto estão diretamente relacionados ao desempenho do classificador. Um outro fato a ser considerado é que alguns classificadores precisam ajustar seus parâmetros de classificação baseados nos treinamentos realizados para terem um ótimo desempenho, e este tempo de treinamento que o classificador necessita realizar para seu ajuste de classificação costuma ser ignorado como critério de avaliação, embora isto possa ser importante para aplicações em tempo real.

Um outro fator importante a ser mencionado é que os resultados obtidos pelos classificadores são baseados na utilização de diferentes bases de dados. Dessa forma, se os mesmos tivessem utilizado a mesma base de dados para fazer os treinamentos e os testes, poder-se-ia obter, de forma justa, uma avaliação dos resultados alcançados por estes classificadores.

De uma maneira geral, pode-se concluir que muitos métodos de *Thresholding* têm um bom desempenho na detecção de pele, além da vantagem de serem simples

e rápidos, mas também têm muitas limitações e detectam de forma errada (*falso positivo*) o que seria pele humana. Seus valores diferem significativamente em diversos modelos de cores e em diferentes iluminações, o que dificulta o encontro de uma faixa de valores adequados para identificar todos os sujeitos de cor de pele diferentes.

Quando utilizadas as mesmas bases de dados, o desempenho entre as técnicas de Histograma, Gaussiana e Rede Neural não tiveram resultados muito diferentes. Em resumo, as técnicas de Histograma tiveram um resultado um pouco melhor na classificação do que as técnicas empregando Gaussianas ou Redes Neurais (MLP).

Também é importante notar que os Histogramas obtiveram este resultado com relação à Gaussiana no que diz respeito ao desempenho de classificação de pixels-pele, como mostram os trabalhos de Jones e Rehg (2002), Phung et al. (2005) e Jayaram et al. (2004). Entretanto as Gaussianas são mais escolhidas para este propósito, pois, com menos dados de treinamentos, este modelo tem generalizado muito melhor do que os demais classificadores.

As Gaussianas, quando utilizam grandes conjuntos de dados, são computacionalmente mais caras que os demais e também mais lentas quando realizam a classificação, já que todas as probabilidades precisam ser avaliadas em valores restritos de cor. Alguns trabalhos, como Fu et al. (2004), conseguiram reduzir esses altos custos, ocorridos durante o treinamento, realizando algumas atividades, como o uso de histogramas, redução do conjunto de dados e agrupamento dos dados da vizinhança do pixel.

O trabalho de Phung et al. (2005) comparou alguns classificadores, utilizando a mesma base de dados e demonstrou que uma Rede Neural MLP obteve um desempenho melhor que a Gaussiana e o Threshold, por exemplo. Mas quando foi avaliada a questão de armazenamento de dados, a Rede Neural MLP teve um resultado insatisfatório devido ao fato de possuir os requisitos de armazenamento muito menores que os demais.

Portanto a segmentação da pele humana tem sido utilizada para inúmeras finalidades, e a escolha do melhor **espaço** de cores que represente a pele humana dentro de um modelo de cores é uma das maiores dificuldades de pesquisa relativas à segmentação de pele humana. Um exemplo é o estudo realizado por Kovac et al. (2003), no modelo de cores RGB, em que ele especifica que as cores determinantes de pele humana **com iluminação constante da luz do dia** são aquelas que satisfazem a seguinte equação:

$$\begin{aligned} &R > 95 \text{ and } G > 40 \text{ and } B > 20 \text{ and} \\ &(\max(R, G, B) - \min(R, G, B) > 15) \text{ and} \\ &(|R - G| > 15) \text{ and } (R > G) \text{ and } (R > B) \end{aligned} \quad (4.1)$$

ou que as cores determinantes de pele humana **com iluminação artificial, inclusive na luz do dia**, são aquelas que satisfazem a seguinte equação:

$$\begin{aligned} &R > 220 \text{ and } G > 210 \text{ and } B > 170 \text{ and} \\ &|R - G| \leq 15 \text{ and } R > B \text{ and } G > B \end{aligned} \quad (4.2)$$

Estas restrições servirão de base para o desenvolvimento das atividades deste trabalho.

# Capítulo 5

## Metodologia Proposta

### 5.1 Materiais

Para a realização deste trabalho, são utilizadas as seguintes ferramentas:

- **Adobe Photoshop CS2 (Versão 9.0)** e **Corel Draw (Versão 12)**: Softwares para tratamento de imagens, especificamente para a conversão de imagens de um formato a outro.
- **Matlab (Versão 7.0)**: Software de desenvolvimento utilizado para testar os algoritmos de segmentação de imagens;
- **Mysql (Versão 5.0)**: Software de banco de dados utilizado para manipulação e gerenciamento das informações geradas no processamento das imagens.
- **Banco AR**: Banco de imagens que possui cento e trinta pessoas (modelos) e vinte variações das imagens de cada uma dessas pessoas. Totalizando duas mil e seiscentas imagens (MARTINEZ E BENAVENTE, 1998);
- **Câmera Sony DSC-S90**: Máquina fotográfica digital da marca Sony, modelo DSC-S90 com resolução de até 4.1 *megapixel*<sup>1</sup>.

### 5.2 Métodos

- **Escolha do modelo de cores**: Primeiramente é montada uma *imagem padrão (colcha de retalhos)* para definir o modelo de cores a ser utilizada nas próximas atividades;

---

<sup>1</sup>**Megapixel** é um termo utilizado que mostra o quanto uma imagem pode ser ampliada. Por exemplo, uma imagem de 1.3 Megapixels de resolução significa que ela possui aproximadamente 1.300.000 de pixels, ou seja, um tamanho aproximado de 1024 de altura por 1280 de largura.

- **Preparação da base de dados de teste:** Após escolhido o modelo de cores, é criada uma nova *imagem padrão* (ou seja, outra *colcha de retalhos*) ainda maior que a anterior com a finalidade de extrair os valores das médias que auxiliarão na segmentação de pele em imagens diversas;
- **Aquisição dos maiores e menores valores das médias:** São obtidos os maiores e os menores valores das médias de vizinhança dos pixels em subimagens contidas na imagem padrão da colcha de retalhos nas seguintes situações: (1) diversos tamanhos de subimagens (3x3, 5x5, 7x7 e 9x9) mas com a imagem padrão numa única resolução (900x1000); (2) com a imagem padrão em diversas resoluções (900x1000, 640x480, 320x240 e 160x120) em apenas um tamanho de subimagem (3x3); e (3) com a imagem padrão em diversas resoluções (900x1000, 640x480, 320x240 e 160x120) em diversos tamanhos de subimagem (3x3, 5x5, 7x7 e 9x9).
- **Segmentação de pele conforme as médias obtidas:** A segmentação de pele é realizada a partir de algoritmos que identificam, a pele humana em uma determinada imagem. Esta segmentação é realizada utilizando-se: (1) os valores obtidos pelos diversos tamanhos das subimagens; (2) os valores obtidos pelas diferentes resoluções da imagem padrão; e (3) com os valores obtidos pelo relacionamento entre os diversos tamanhos das subimagens e as diferentes resoluções da imagem padrão.
- **Comparação de Resultados:** Os resultados obtidos são comparados com os resultados de trabalhos citados no capítulo 4;

### 5.2.1 Escolha do modelo de cores

Apesar dos Modelos HSV e HSI aparentemente serem os mais indicados por possuírem apenas uma variável que determina a cor do pixel ( $\mathbf{H}$ , *matiz*), a escolha de um sistema de cor mais adequado ao desenvolvimento do trabalho foi realizada pela análise dos resultados obtidos através da comparação entre três modelos: HSV, HSI e RGB.

Procedimento de escolha do modelo:

- Primeiramente foram obtidas, por uma câmera digital, **trinta imagens** do rosto de trinta pessoas (treze homens e dezessete mulheres) de diferentes tonalidades de pele e todas sob as mesmas condições de iluminação (figura 5.1). Assim, montou-se um pequeno banco de imagens para ajudar na escolha do modelo de cores mais adequado.



Figura 5.1: Algumas das imagens obtidas para auxiliar na escolha do modelo de cores.

- Depois disso, foram escolhidas **doze imagens**, das trinta imagens existentes (seis imagens de homens e seis de mulheres), sendo essas imagens convertidas do formato *JPEG*<sup>2</sup>, que é o padrão quando capturadas pela câmera, para o formato *BMP*<sup>3</sup>. Após a conversão, foram extraídas pequenas amostras de partes do rosto de cada uma destas pessoas.
- Essas amostras (ou pequenos pedaços de pele) extraídas formaram uma nova imagem padrão (de tamanho 120x120, totalizando 14.400 pixels), chamada de **colcha de retalhos**, como mostra a figura 5.2.

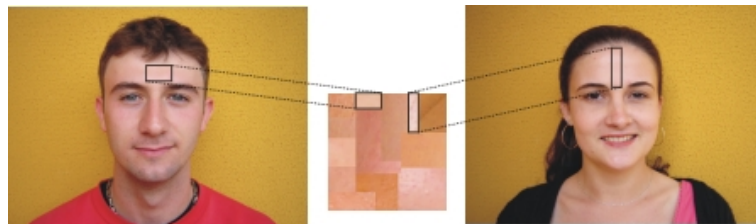


Figura 5.2: **Colcha de retalhos** (ao centro) montada a partir de pedaços de imagens de pele humana.

- Após a montagem desta colcha de retalhos, foi desenvolvido um algoritmo (**ANEXO B - Programa 01**) que **extraísse as características de cada pixel** desta imagem (chamado de **varredura**) nos modelos RGB, HSV e HSI. Depois de extraídas as características destes pixels, esses valores foram migrados de um arquivo texto, para um banco de dados, onde o conjunto de valores de cada pixel (cada pixel composto por três valores) forma um **registro**. (Ver figura 5.3).
- A intenção é organizar essas "amostras de pele" de maneira que nenhum registro se repita. Os 14.400 registros gerados neste banco de dados (originados dos

<sup>2</sup>**JPEG** (*Joint Photographic Experts Group*) é um formato de compressão de imagem que mostra uma determinada imagem com perda de suas características. Ou seja, para otimizar o armazenamento, o número de pixels de uma imagem é reduzido representando a imagem com menor qualidade.

<sup>3</sup>**BMP** (*Bitmap*) é um formato de compressão de imagem que mostra a descrição de cada pixel da imagem. Ou seja, para que uma imagem seja representada com todas as suas características ela precisa de número muito grande de pixels. Assim, o formato BMP representa essa imagem com maior qualidade se comparada ao JPEG.

pixels da imagem da colcha de retalhos) são **organizados**, via comandos SQL<sup>4</sup>, de maneira que todos sejam distintos uns dos outros, conforme mostra a figura 5.3.

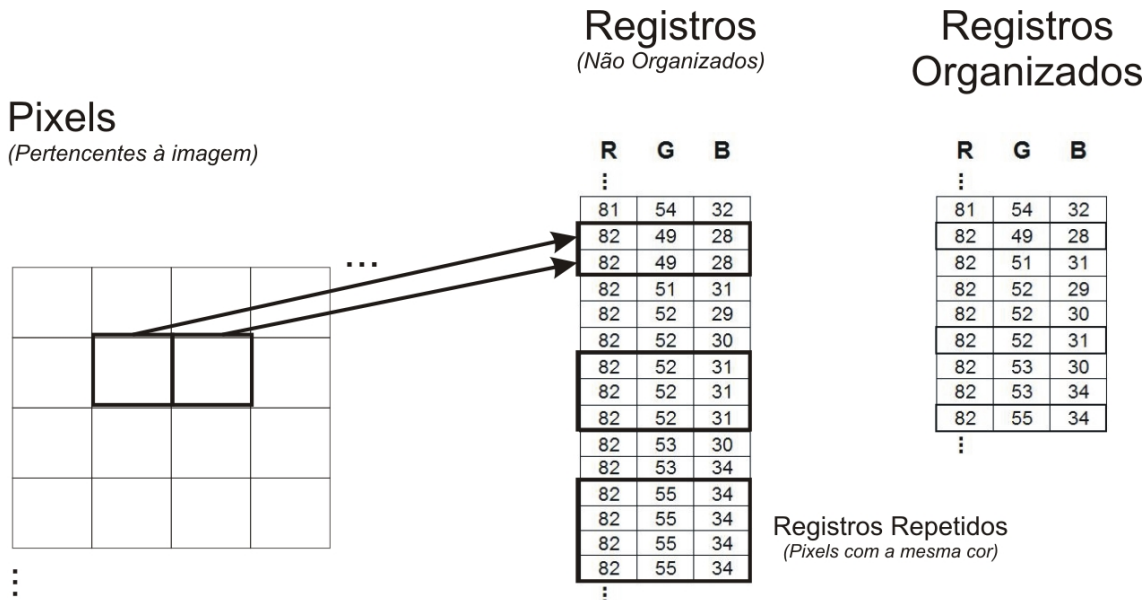


Figura 5.3: Organização dos registros no banco de dados.

- Depois destes dados organizados, são gerados novos dados resultados de pixels originados de outra varredura em uma nova imagem de pele humana (de tamanho 73x96, totalizando 7.008 pixels), chamada de *imagem teste de pele*, que é uma imagem semelhante à "colcha de retalhos" mas de tamanho menor. Assim, este mesmo programa (**ANEXO B - Programa 01**) varre esta nova imagem teste de pele gerando novos resultados para que estes, depois de organizados, sejam comparados àqueles já organizados pela varredura da colcha de retalhos.
- Com todos os registros organizados são verificados quantos registros coincidem entre as bases (geradas pela "Colcha de Retalhos" e pela "Imagem teste de pele"). Ou seja, quantos registros de cada imagem são iguais, ou **coincidentes**, entre si em cada modelo de cor.

Analisando-se os resultados, verificou-se que o melhor modelo de cores a ser utilizado neste trabalho é o modelo **HSV**, por ter fornecido maior número de registros coincidentes.

<sup>4</sup>SQL (*Structured Query Language*) é uma linguagem de pesquisa para banco de dados relacional.

### 5.2.2 Geração da Colcha de Retalhos para a Base de Testes

Depois de definido o modelo de cores, é construída uma **nova colcha de retalhos** a partir de um novo banco de imagens. Os procedimentos para a criação desta nova colcha de retalhos para a base de testes são:

- Monta-se a nova imagem (de pele humana) seguindo os mesmos procedimentos de montagem da colcha de retalhos descritos para escolher entre os modelos de cores. Esta nova imagem é montada utilizando-se o **Banco AR** com **cento e trinta pessoas** (setenta e seis homens e sessenta mulheres); cada uma dessas pessoas foi fotografada cerca de 20 vezes em diferentes posições, iluminações, vestimentas, etc. (MARTINEZ E BENAVENTE, 1998).
- Para a montagem da colcha, são utilizadas **cem imagens** de pessoas (cinquenta de cada sexo) com quatro imagens de cada uma (figura 5.4). A escolha dessas imagens levou em conta a ausência de variações de iluminação, com variações apenas na posição.



Figura 5.4: Pessoa em diferentes posições (ou expressões) utilizadas para a montagem da nova colcha de retalhos.

- Quando comparada a nova colcha de retalhos (figura 5.5) com a anterior (figura 5.2) as diferenças são: (1) o **tamanho da imagem** (900x1000, totalizando 900.000 pixels); (2) o **número de retalhos** maior do que o modelo anterior ; e (3) as imagens de onde foram extraídas as amostras de pele que compõem esta nova colcha de retalhos (Banco AR).

A montagem desta nova colcha de retalhos é necessário para se obter os valores das médias a serem usadas como parâmetros para a segmentação de pele humana.

### 5.2.3 Cálculo das Médias Variando-se o Tamanho das Vizinhanças das Subimagens

Depois de criada a imagem da colcha de retalhos de pele humana, realizar-se a segmentação por similaridade, obtendo-se a **média aritmética** do pixel *semente* com relação à sua vizinhança dentro de uma **subimagem**.



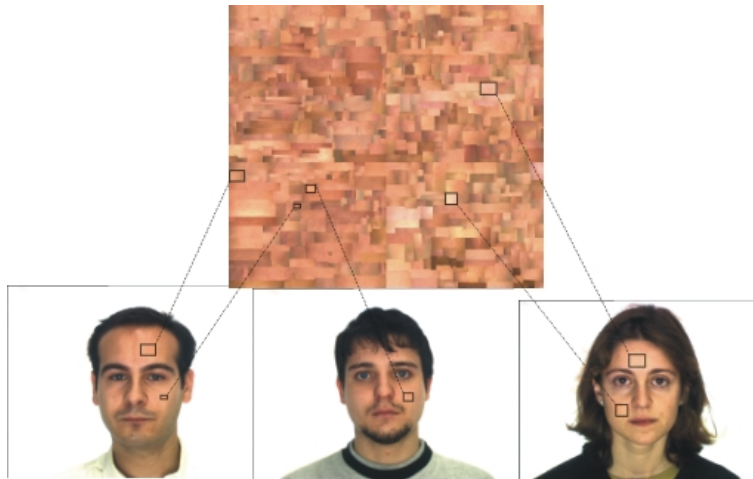


Figura 5.5: Nova **colcha de retalhos** montada da mesma forma que a anterior.

Com os valores das maiores e menores médias do pixel *semente* obtidas nesta nova colcha de retalhos, busca-se classificar em imagens teste os pixels de **pele** e **não pele**.

De maneira mais detalhada, pode-se assim descrever:

- O novo algoritmo (**ANEXO B - Programa 04**) calcula a média do pixel semente (pixel central) com relação à sua vizinhança em uma subimagem (ou máscara) de tamanho 3x3. A média dos pixels pertencentes a esta subimagem na colcha de retalhos, refere-se ao valor de **H** (*matiz*) do pixel, devido ao fato do modelo de cores utilizado ser o HSV.
- Desloca-se esta subimagem de um pixel à direita da colcha de retalhos, realizando uma varredura, como mostra a figura 5.6).
- Após o cálculo de todos os valores da média, são identificados, os **maiores** e os **menores** valores encontradas em cada subimagem 3x3.

O procedimento realizado para encontrar esses valores na subimagem de tamanho 3x3 também é realizado para encontrar os maiores e os menores valores das médias em subimagens de outros tamanhos: 5x5, 7x7 e 9x9 (**ANEXO B - Programas 04**), totalizando, assim, quatro valores de médias inferiores e quatro valores de médias superiores.

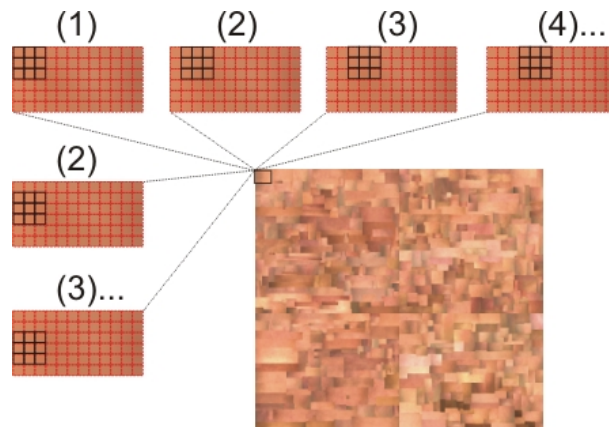


Figura 5.6: Nova colcha de retalhos utilizando uma subimagem para a obtenção das médias.

#### 5.2.4 Cálculo das Médias de Vizinhanças em Imagens com Diversas Resoluções

Depois de encontrados os valores das maiores e menores médias relativas às subimagens de diferentes tamanhos (3x3, 5x5, 7x7 e 9x9) na imagem da colcha com resolução de 900x1000 pixels, são realizados estes mesmos procedimentos **mas com alterações de resolução e subimagem 3x3**. É adotada a seguinte abordagem:

- A imagem da colcha de retalhos é redimensionada (**ANEXO B - Programa 02**) para outras resoluções. Partindo-se da resolução de 900x1000 (900.000 pixels) gera as imagens de:
  - 640x480 pixels de resolução - Totalizando 307.200 pixels
  - 320x240 pixels de resolução - Totalizando 76.800 pixels
  - 160x120 pixels de resolução - Totalizando 19.200 pixels
- A **interpolação**<sup>5</sup>, cada vez mais requisitada para escalonar imagens, é um processo fundamental em processamento de imagens (BOVIK, 2000). O software de desenvolvimento Matlab possui a função `IMRESIZE` para o redimensionamento de imagens por interpolação. Pode-se escolher entre os métodos: `NEAREST NEIGHBOR` (Interpolação pelo vizinho mais próximo), `BICUBIC` (Interpolação Bicúbica) e `BILINEAR` (Interpolação Bilinear). O método escolhido para redimensionar a imagem foi a **interpolação pelo vizinho mais próximo**.

<sup>5</sup>**Interpolação** é a intercalação de pixels de uma imagem quando se modifica sua resolução, ou seja, é qualquer processo de obtenção dos valores de uma função dentro de um intervalo mediante o conhecimento de seu comportamento nos extremos desse intervalo.

A escolha desta interpolação deve-se ao fato de serem atribuídos aos pixels da imagem redimensionada (com resolução inferior à sua original) valores sem nenhuma modificação. Ou seja, as imagens quando alteradas de tamanho nesta interpolação sofrem as seguintes mudanças:

- Quando a imagem é **aumentada**, por exemplo de 6x4 para 12x8, **os valores dos pixels** que irão compor esta nova imagem são **replicados para os pixels que estão à sua volta**, os vizinhos mais próximos, conforme figura 5.7.
- Quando a imagem é **reduzida**, por exemplo de 6x4 para 3x2, **os valores dos pixels** que irão compor esta nova imagem **serão os pixels centrais (os pixels semente), de uma subimagem** da imagem original, proporcional ao tamanho da redução para a nova imagem, conforme figura 5.7.
- Para cada alteração do tamanho da imagem, são realizados os mesmos procedimentos especificados anteriormente, ou seja, para o tamanho de imagem 640x480 pixels é realizada a varredura e o cálculo para encontrar os valores das maiores e menores médias de vizinhanças considerando-se a subimagem de tamanho 3x3.

O procedimento realizado para encontrar esses valores utilizando apenas a subimagem de tamanho 3x3 com a imagem de resolução 900x1000, também é realizado para encontrar os maiores e os menores valores das médias nesta imagem com outras resoluções 640x480, 320x240 e 160x120, totalizando assim, outros quatro valores de médias inferiores e outros quatro valores de médias superiores.

### 5.2.5 Segmentação Considerando-se o Tamanho da Vizinhança de Pixels nas Subimagens

Os valores das médias obtidas são utilizadas para identificar se os pixels contidos nas imagens a serem analisadas **são** ou **não são** pele humana.

O método é:

- Com uma câmera fotográfica, são obtidas **32 imagens** contendo (ou não) pessoas para serem analisadas (**ANEXO A**), das mais variadas formas possíveis: dentro de locais fechados, sob diferentes tipos de iluminações, com diferentes tipos de pele, etc. As resoluções destas imagens variavam entre 640x480 e 2048x1536. Alguns exemplos destas imagens podem ser observados na figura 5.8.

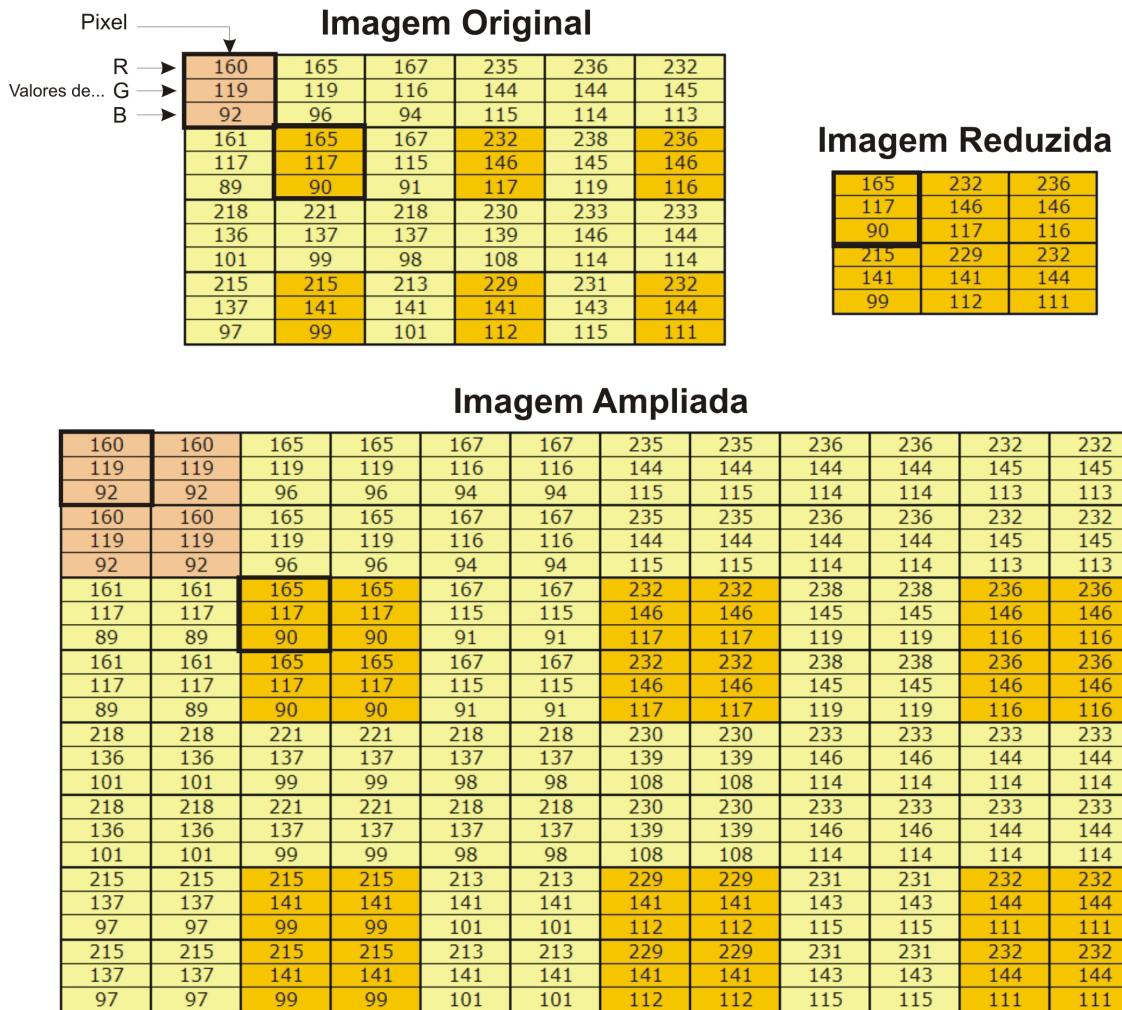


Figura 5.7: Características de cada pixel no formato **RGB** da **imagem original** (resolução 6x4) e o resultado da **ampliação** (resolução 12x8) e **redução** (resolução 3x2) desta imagem.

- As fotos obtidas são convertidas para o formato BMP e aplicado um algoritmo (**ANEXO B - Programa 05**) para varredura completa nesta imagem, para identificar pele humana.
- O processo de identificação adota o mesmo procedimento realizado no cálculo para encontrar os maiores e menores valores das médias de vizinhanças. A diferença é que esse novo algoritmo faz a varredura utilizando os valores das médias encontradas variando-se o tamanho das vizinhanças, na colcha de retalhos, conforme equação 6.1.

$$\bar{H} \min \leq H \text{ pixel} \leq \bar{H} \max \quad (5.1)$$



Figura 5.8: Algumas imagens e suas resoluções para a realização dos testes.

### 5.2.6 Segmentação Considerando a Resolução da Imagem da *Colcha de Retalhos*

Utilizando-se as médias obtidas em apenas um tamanho de subimagem (3x3) nas diferentes resoluções da colcha de retalhos e com as mesmas imagens, são feitas análises para classificação dos pixels em **pele** e **não pele**.

## 5.3 Considerações Finais

A metodologia proposta foi implementada em **Matlab** visando a segmentação de pele humana em imagens gerais. Para isso, foi gerado um modelo de amostras inicial (colcha de retalhos) para escolha do modelo de cores. Após os testes verificou-se que o melhor modelo é o HSV.

Utilizando-se o banco de imagens de faces humanas (*AR*) gerou-se uma nova colcha de retalhos com amostras retiradas do banco. Esta colcha de retalhos forneceu subsídios para a determinação de médias **inferiores** e **superiores** do canal *H*, intervalo este onde se encontram as amostras de pele humana. Com os dados obtidos realizou-se a segmentação de imagens gerais em diversas resoluções espaciais e utilizando-se várias subimagens (3x3, 5x5, 7x7 e 9x9) para a geração das médias.

# Capítulo 6

## Resultados e Conclusões

### 6.1 Resultados

A escolha do modelo de cores a ser utilizado baseou-se nos resultados obtidos num teste que verificou, principalmente, o relacionamento e a integridade das informações resultantes entre: (1) **um padrão** ou modelo de pele (*colcha de retalhos*), e (2) uma **nova imagem de verificação**, que é uma pequena imagem contendo pixels de pele humana.

Depois de extraídas as informações de cada pixel contido na imagem da colcha de retalhos (*o padrão*) e em cada um dos 3 modelos de cores a serem analisados (RGB, HSV e HSI), essas informações foram migradas para um banco de dados com a finalidade de serem organizadas e utilizadas neste trabalho.

Ou seja, cada pixel desta imagem em um modelo de cor específico, é formado por um conjunto de valores que determina suas características, ou a sua cor. Por exemplo, um pixel no modelo RGB com valores R=159, G=109 e B=81 forma um registro em um banco de dados, referente à característica do pixel. Os registros são organizados de maneira que nenhum desses valores se repitam.

Depois que esses registros são organizados, é realizado o mesmo procedimento mas com uma outra imagem (*nova imagem de verificação*) gerando novos dados para serem comparados aos dados da imagem padrão. Os registros organizados, tanto do padrão quanto da imagem de verificação, são apresentados na figura 6.1.

Definido o modelo de cores (HSV) a ser utilizado, uma nova colcha de retalhos (*novo padrão*) é construída, sendo esta com uma resolução (900x1000 pixels) e uma quantidade de amostras de pele ainda maior que a anterior.

O **Programa 03 (ANEXO B)** implementa os algoritmos que verificam sua integridade. Ou seja, se as suas características são apropriadas à classificação de pele,

Imagem	Quantidade Pixels	Resolução	Quantidade Registros
Colcha de Retalhos	14.400	120x120	14.400
Imagem Teste de Pele	7.008	73x96	7.008

Quantidade de registros após a organização				
Modelo Cores	Colcha de Retalhos	Imagem Teste de Pele	Registros Coincidentes	Porcentagem Acerto
RGB	11.179	2.961	437	14.76%
HSV	2.499	115	115	100.0%
HSI	2.531	119	119	100.0%

Figura 6.1: Resultados obtidos para a escolha do modelo de cores, conforme a quantidade de registros gerados em cada uma das bases (Colcha de Retalhos e Imagem Teste de Pele) considerando as coincidências entre elas.

tais como: (1) se os valores dos pixels atendem às especificações de pele mencionadas por Kovac et al. (2003) e (2) se existem espaços (ou pixels) em branco ou não preenchidos que poderiam alterar os resultados (Figura 6.2).

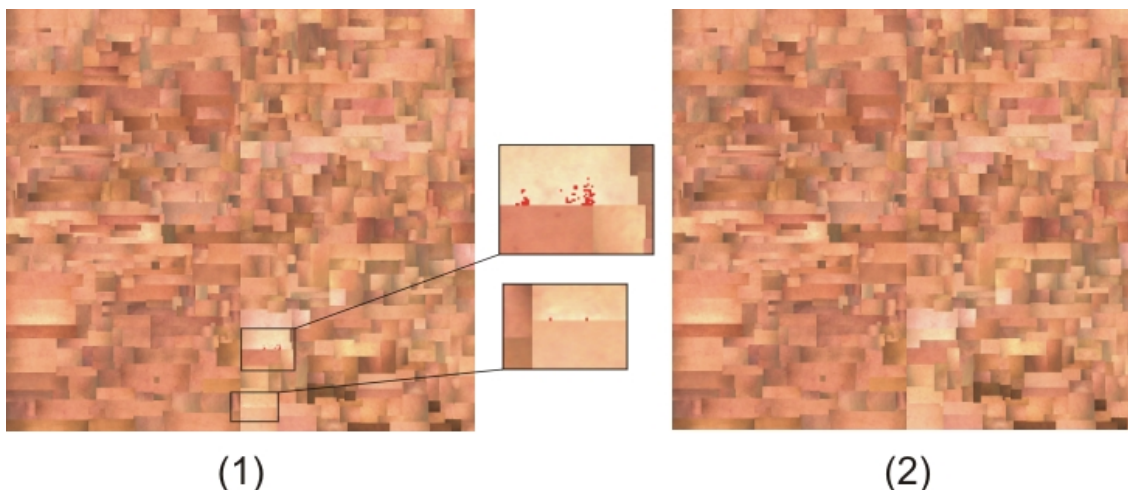


Figura 6.2: Colcha de retalhos em dois momentos: (1) com os pixels que não atendem às especificações de pele devidamente identificados e (2) após a correção dos erros identificados.

Para resolver problemas relativos à estas condições, novas amostras de imagens de pele humana são adicionadas substituindo-se as amostras de imagens irregulares ou cobrindo os espaços (ou pixels) em branco ou não preenchidos.

Após verificadas e corrigidas essas irregularidades, a nova colcha e retalhos de pele humana é utilizada como padrão deste trabalho. A finalidade desta imagem estar adequadamente corrigida é para:

- Encontrar nesta os valores das **médias superiores** e das **médias inferiores** dos pixels *semente* contidos **nas subimagens** de tamanhos: 3x3, 5x5, 7x7 e

9x9 da colcha de retalhos com resolução de 900x1000. Após encontrar esses valores relativos a todas subimagens contidas na imagem padrão, esses tornam-se parâmetros para definir o pixel como pele em outra imagem. Os valores encontrados neste procedimento podem ser vistos na figura 6.3.

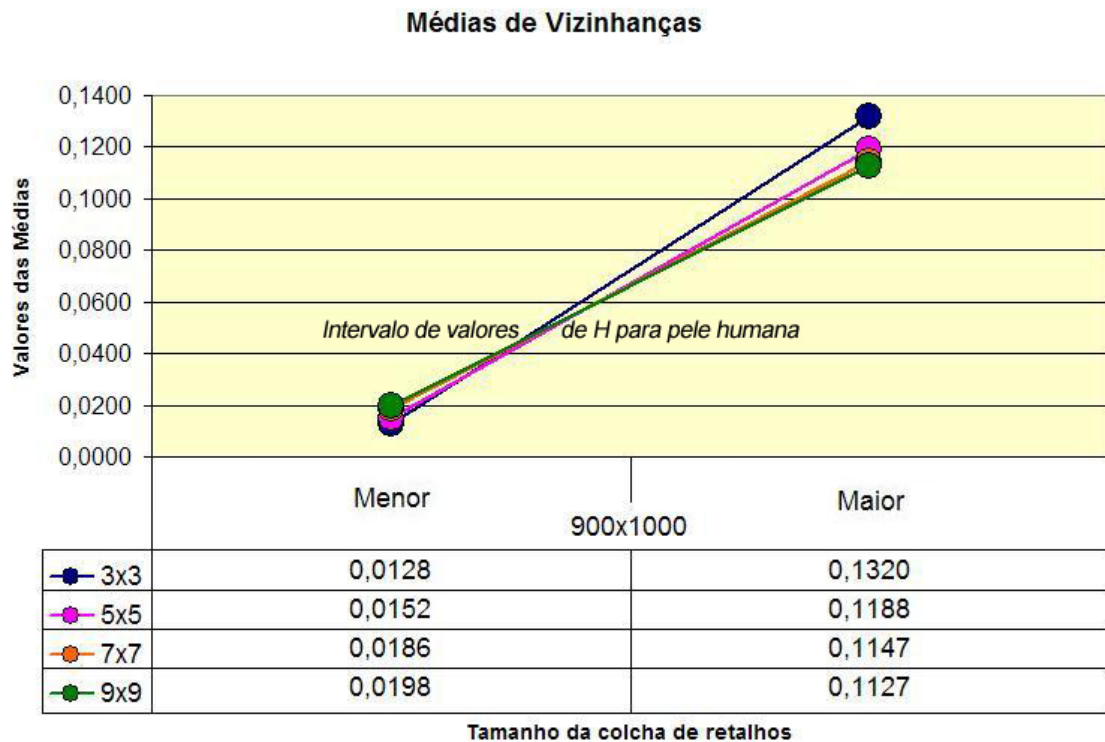


Figura 6.3: Gráfico contendo os valores encontrados nas subimagens da colcha de retalhos com resolução de 900x1000 pixels.

- O algoritmo que identifica pele humana utiliza-se desses valores para fazer a segmentação nas imagens selecionadas. Os resultados deste procedimento podem ser analisados na figura 6.4. Com os valores das maiores e menores médias nas quatro subimagens em apenas uma resolução (900x1000), foi implementado um algoritmo para segmentação utilizado as médias obtidas a partir do dimensionamento da imagem da colcha de retalhos.
- Assim, o processamento realizado com os quatro valores das médias obtidas pela alteração da resolução da colcha de retalhos (900x1000, 640x480, 320x240 e 160x120), apenas com a subimagem de tamanho 3x3, gera resultados diferentes dos anteriores, como pode-se ver na figura 6.5.
- Com a finalidade de se aumentar a eficiência, foi utilizada a imagem da colcha de retalhos em **diferentes resoluções** (900x1000, 640x480, 320x240 e



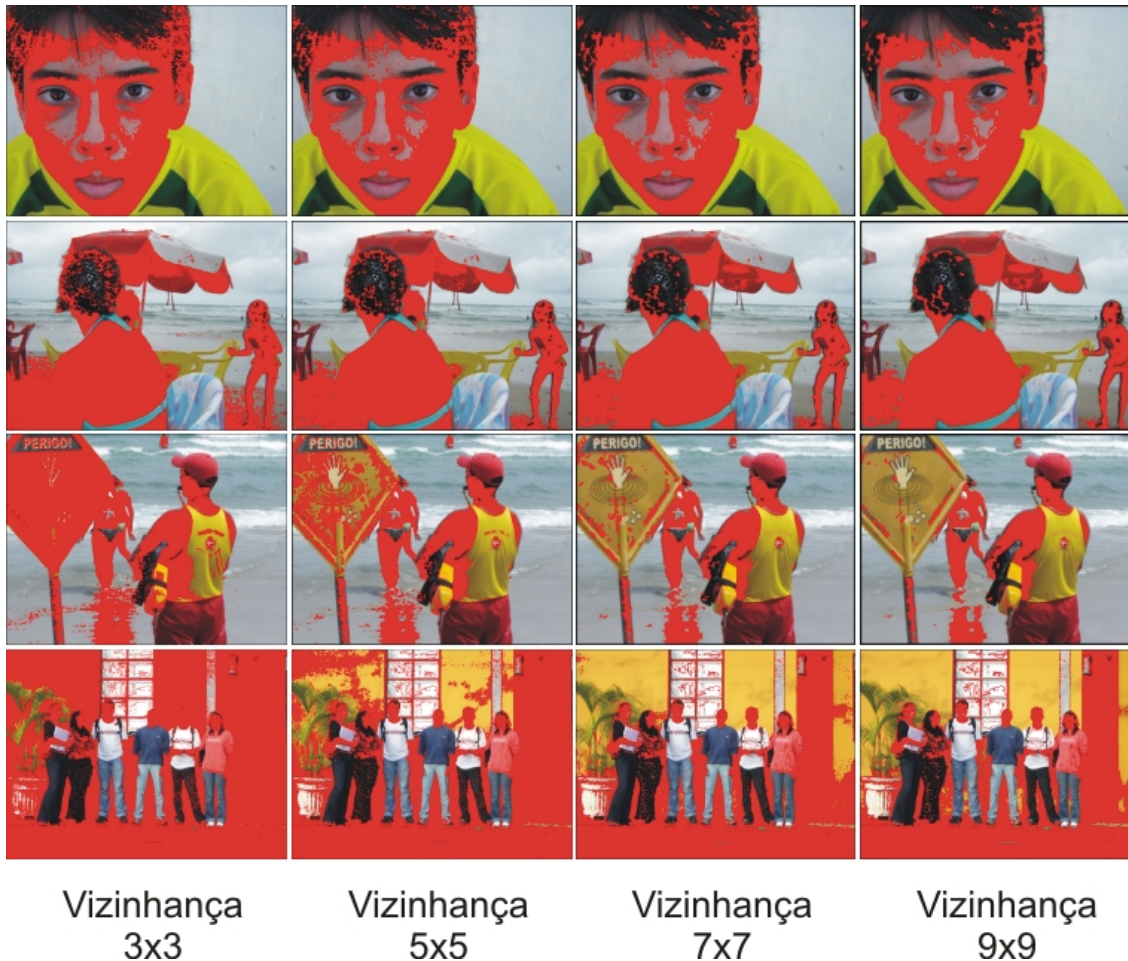


Figura 6.4: Resultados da segmentação conforme os valores das médias obtidas dos diferentes tamanhos de vizinhança: 3x3, 5x5, 7x7 e 9x9 em imagem padrão de resolução 900x1000.

160x120) com os **diferentes tamanhos de subimagens** (3x3, 5x5, 7x7 e 9x9). O resultado, dos maiores e menores valores das médias obtidas, pode ser observado na figura 6.6.

- Os resultados da segmentação realizada com cada um deles (os valores das maiores e menores médias) podem ser vistos nas figuras: 6.7 e 6.8:

## 6.2 Análise dos Resultados

### 6.2.1 Escolha do modelo de cores

Depois de organizadas, comparadas e analisadas as informações resultantes dos registros gerados pela verificação das duas imagens (a colcha de retalhos padrão e a imagem de verificação), pode-se notar que a porcentagem de acerto (ou registros

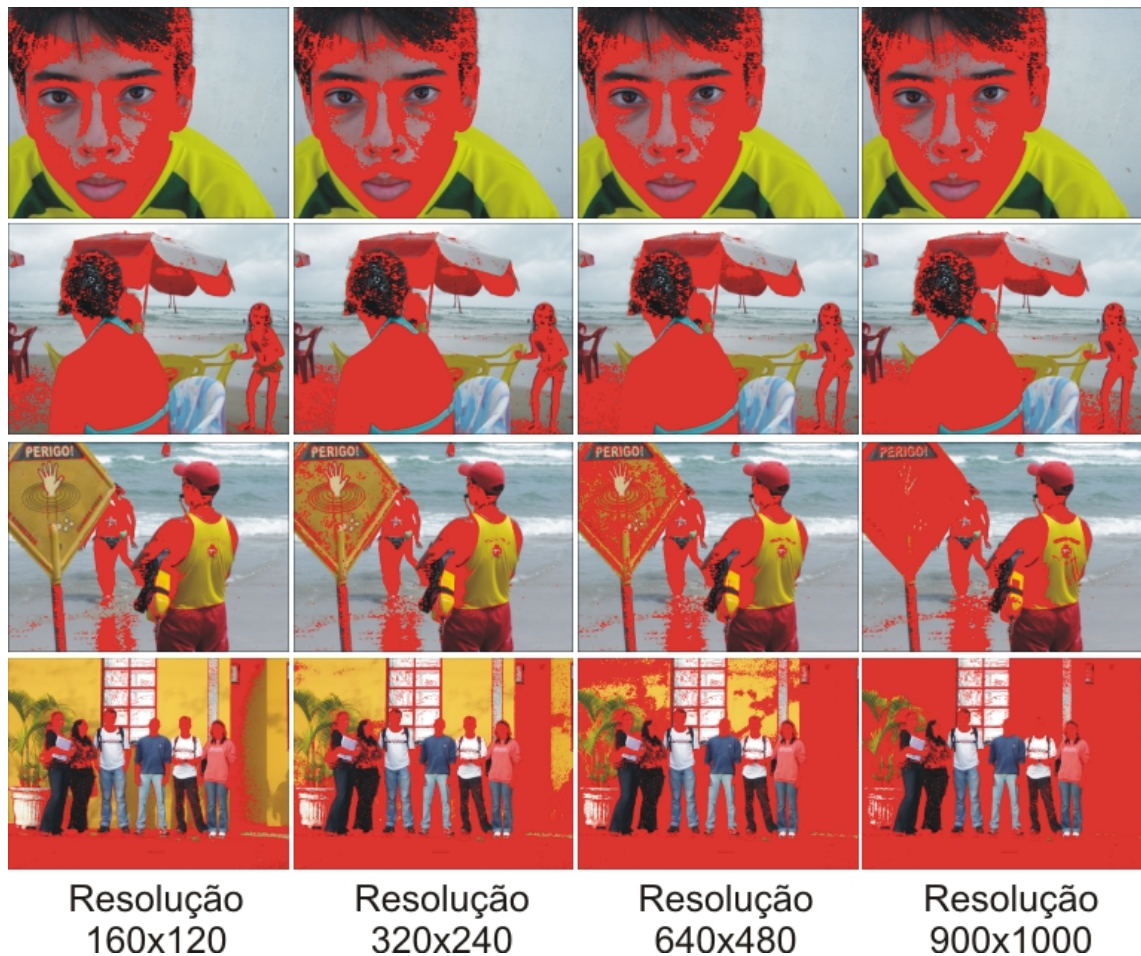


Figura 6.5: Resultados da segmentação conforme a resolução da colcha de retalhos: 160x120, 320x240, 640x480 e 900x1000 considerando uma subimagem de tamanho 3x3.

iguais) entre os modelos analisados teve resultados iguais entre os modelos HSV e HSI e superiores quando comparado com o modelo RGB.

Tanto no modelo HSV quanto no HSI, o resultado de acerto foi de 100 por cento, enquanto que no modelo RGB a porcentagem de acertos foi de 14,76 por cento. O modelo HSV possui uma pequena vantagem em relação ao modelo HSI relativo à quantidade de registros organizados.

Enquanto o modelo de cor RGB é composto por 3 variáveis que representam a cor do pixel (**R**, **G** e **B**), o modelo HSV é representado apenas por uma variável (o **H**). Portanto, a utilização de um modelo composto por mais variáveis faz com que a taxa de acerto seja menor do que um modelo composto por menos variáveis.

Como a quantidade de registros gerados nos modelos também foi outro fator a ser analisado, o modelo de cores HSV foi o escolhido para o desenvolvimento deste trabalho.

Resolução Colcha Retalhos	Médias	Tamanho das subimagens (Vizinhanças)			
		3x3	5x5	7x7	9x9
1000x900	Menor	0,0128	0,0152	0,0186	0,0198
	Maior	0,1320	0,1188	0,1147	0,1127
640x480	Menor	0,0169	0,0198	0,0210	0,0221
	Maior	0,1197	0,1125	0,1099	0,1082
320x240	Menor	0,0197	0,0230	0,0260	0,0274
	Maior	0,1147	0,1084	0,1045	0,0998
160x120	Menor	0,0204	0,0271	0,0312	0,0385
	Maior	0,1045	0,0970	0,0850	0,0787

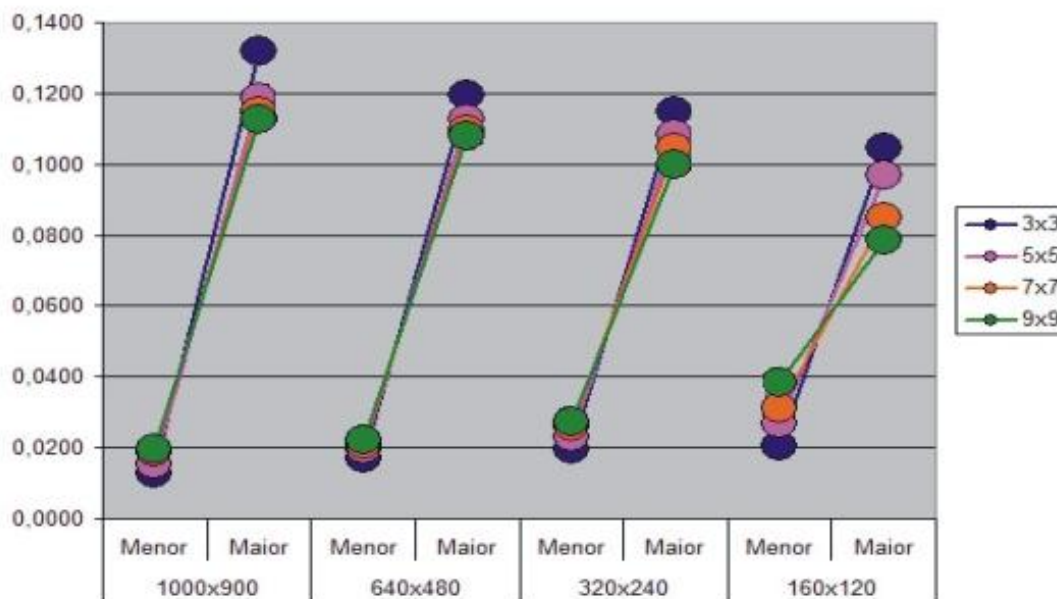


Figura 6.6: Valores das menores e maiores médias obtidas para várias resoluções e várias subimagens.

## 6.2.2 Colcha de Retalhos

Na criação da nova imagem de colcha de retalhos para obtenção das médias, os aspectos inicialmente observados neste procedimento foram: (1) a **quantidade de amostras de pele humana** contida nesta nova imagem, e (2) o **tamanho da imagem** (900x1000 pixels de resolução).

A intenção de construir uma imagem com essas características teve o intuito de se aumentar o número de parâmetros, ou ter mais valores de média, que influenciassem nos resultados do que **é** ou **não é** pele humana nas imagens.

Observou-se que o aumento no número de amostras de pele dentro da colcha de retalhos tem pequena influência na variação das médias mínimas e máximas.

### 6.2.3 Tamanho da Vizinhança

Os valores das médias inferiores e superiores variaram em cada processamento relativo aos diferentes tamanhos das subimagens (3x3, 5x5, 7x7 e 9x9) na resolução de 900x1000 pixels. Ou seja, **conforme o tamanho da subimagem é aumentado (indo do 3x3 para o 9x9), o valor das médias inferiores aumenta e o valor das médias superiores diminui**, conforme gráfico da figura 6.3.

Conforme figura 6.6 os valores das médias inferiores aumentam com a redução da resolução da colcha de retalhos e os valores das médias superiores diminuem.

Dessa forma, quando se segmenta imagens com valores obtidos da subimagem de tamanho 3x3, encontra-se pixels que não são pele humana mais do que quando se segmenta a mesma imagem com valores obtidos de uma subimagem de tamanho maior. Em contrapartida, para melhor classificar, ou melhor segmentar uma imagem com o que é pele humana, a subimagem de tamanho 9x9 é melhor conforme mostra a figura 6.4.

### 6.2.4 Resolução da Colcha de Retalhos

Nesta abordagem foi realizada a segmentação com valores encontrados no processamento relativo a apenas um tamanho de subimagem (3x3) e em diferentes resoluções da colcha de retalhos (900x1000, 640x480, 320x240 e 160x120). Os resultados obtidos foram praticamente os mesmos obtidos pelo processamento anterior.

Ou seja, quando se segmenta imagens com valores obtidos da colcha de retalhos com uma resolução maior (tamanho 900x1000, por exemplo), encontra-se pixels que não são pele humana mais do que quando se segmenta a mesma imagem com valores obtidos da colcha de retalhos com uma resolução menor (160x120 por exemplo), conforme pode ser visto na figura 6.5.

### 6.2.5 Verificação completa

As figuras 6.7 e 6.8 mostram uma verificação completa, ou seja, a combinação de todas as variações possíveis entre os tamanhos das subimagens (3x3, 5x5, 7x7 e 9x9) e as resoluções da colcha de retalhos (160x120, 320x240, 640x480 e 900x1000).

Pode-se observar que a influência do aumento do tamanho da subimagem, na segmentação de pele humana, é maior do que a influência no aumento da resolução da colcha de retalhos. Ao contrário, o aumento da resolução da colcha de retalhos é inverso, relativamente à segmentação correta de pixel de pele e não pele, ou seja, dentro de um tamanho de subimagem, o resultado é melhor em menores resoluções.

Assim, os testes utilizando-se as médias (superiores e inferiores) **fixando-se a resolução e variando-se o tamanho das subimagens** (Subimagens: 3x3, 5x5, 7x7 e 9x9 - Resolução: 900x1000) e **fixando-se o tamanho da subimagem e variando-se a resolução** (Subimagens: 3x3 - Resolução: 900x1000, 640x480, 320x240 e 160x120) **obtiveram praticamente os mesmos resultados com pouca ou nenhuma variação.**

## 6.3 Comparação dos Resultados

São muitos os trabalhos desenvolvidos para segmentação de pele em imagens coloridas e muitas as suas aplicações. Dentre os trabalhos citados no capítulo 4, o trabalho desenvolvido por Kovac et al. (2003) foi a motivação para o desenvolvimento deste. Suas equações para segmentação de pele humana serão usadas como base para comparação de resultados (**ANEXO B - Programas 06**).

A figura 6.9 mostra o resultado da segmentação obtida através da metodologia desenvolvida por Kovac et al. (2003) quando aplicado sobre as imagens utilizadas em nosso trabalho.

### 6.3.1 Resultados comparados de forma visual (Qualitativa)

Conforme pode ser visto na figura 6.10, os resultados obtidos, **observados visualmente**, pelas restrições de Kovac et al. (2003) não atingiram, ou melhor, não garantem a segmentação correta em 100 por cento dos casos.

Como pode ser observado, somente na primeira imagem mais à esquerda, os resultados da metodologia proposta neste trabalho foi ligeiramente inferior ao de Kovac et al. (2003). Em todas as outras imagens, não somente o acerto de pixels foi maior, mas também a classificação de pixels de não pele.

Além disso, as restrições de Kovac et al. (2003) são fixas, o que seguramente dará resultados inferiores, relativamente à nossa abordagem, quando o conjunto de imagens utilizados tiver características restritivas com relação à iluminação, foco, distância do objeto, etc.

Com a metodologia proposta, as médias a serem obtidas podem ser diferentes de acordo com o banco de imagens a ser processado, tornando a segmentação mais flexível.

### 6.3.2 Resultados comparados de forma numérica (Quantitativa)

Em todas as 32 imagens foi realizada também uma segmentação manual com o intuito de compará-la com os resultados obtidos pelo algoritmo proposto e pelo método de Kovac et al. (2003). O ANEXO A mostra as imagens segmentadas manualmente.

Nesta análise, foram especificados os parâmetros de **ERRO** e **ACERTO** da seguinte forma:

- **ACERTO PELE**: O algoritmo segmenta o que **É PELE**, na imagem segmentada manualmente, de forma correta. Ou seja, na imagem o pixel é pele e o programa segmentou corretamente.
- **ACERTO NÃO PELE**: O algoritmo segmenta o que **NÃO É PELE**, na imagem segmentada manualmente, de forma correta. Ou seja, na imagem o pixel não é pele e o programa segmentou corretamente.
- **ERRO PELE**: O algoritmo segmenta o que **É PELE**, na imagem segmentada manualmente, de forma errada. Ou seja, na imagem o pixel não é pele e o programa segmenta como sendo pixel pele.
- **ERRO NÃO PELE**: O algoritmo segmenta o que **NÃO É PELE**, na imagem segmentada manualmente, de forma errada. Ou seja, na imagem o pixel é pele e o programa segmenta como sendo pixel não pele.

A quantidade de pixel contido nas imagens e os resultados numéricos, onde as maiores quantidades de pixels estão destacados, podem ser vistos na tabela 6.1. (O gráfico que melhor demonstra esses resultados estão no ANEXO C).

## 6.4 Conclusões

Enquanto que, no trabalho realizado por Kovac et al. (2003), os resultados mostram que a quantidade de pixels pele é segmentada corretamente em número maior que este trabalho (ACERTO PELE), os resultados também mostram que este trabalho segmenta o que não é pele, de forma mais correta que o de Kovac et al. (2003) (ACERTO NÃO PELE).

O que ocorre nos ACERTOS, ocorre também nos ERROS só que de maneira contrária, ou seja, no trabalho de Kovac et al. (2003) a quantidade de pixel que não

IMAGENS	SEGMENT.MANUAL		SEGMENTAÇÃO PROGRAMA							
	QUANTIDADE PIXELS ENCONTRADOS		QUANTIDADE PIXEL - ESTE TRABALHO				QUANTIDADE PIXEL - TRABALHO KOVAC			
	PELE	NÃO PELE	ERRO		ACERTO		ERRO		ACERTO	
			PELE	NÃO PELE	PELE	NÃO PELE	PELE	NÃO PELE	PELE	NÃO PELE
Imagem 01	101528	205672	66940	4136	34588	201536	4017	8273	97511	197399
Imagem 02	7955	299245	7710	1748	245	297497	2146	5367	5809	293878
Imagem 03	9975	297225	3867	25983	6108	271242	7890	14814	2085	282411
Imagem 04	6503	300697	4742	6632	1761	294065	4214	559	2289	300138
Imagem 05	21927	285273	12280	6137	9647	279136	9782	3374	12145	281899
Imagem 06	537	306663	491	5457	46	301206	217	84	320	306579
Imagem 07	14352	292848	12726	677	1626	292171	3506	5490	10846	287358
Imagem 08	21278	285922	19878	7079	1400	278843	1548	35017	19730	250905
Imagem 09	24998	282202	8716	549	16282	281653	2125	12668	22873	269534
Imagem 10	26340	280860	18476	13529	7864	267331	20186	5473	6154	275387
Imagem 11	23981	283219	20691	2298	3290	280921	6580	9248	17401	273971
Imagem 12	5302	301898	3388	1038	1914	300860	1269	255	4033	301643
Imagem 13	64290	242910	7982	7010	56308	235900	9764	19957	54526	222953
Imagem 14	7087	300113	2377	12648	4710	287465	2005	13527	5082	286586
Imagem 15	39324	267876	20517	5164	18807	262712	8392	12671	30932	255205
Imagem 16	25246	281954	11147	2187	14099	279767	7340	56961	17906	224993
Imagem 17	75449	231751	39353	4733	36096	227018	8773	25444	66676	206307
Imagem 18	12017	295183	6557	30	5460	295153	4853	911	7164	294272
Imagem 19	73008	234192	49774	238	23234	233954	23188	10737	49820	223455
Imagem 20	11252	295948	8877	12222	2375	283726	3766	36929	7486	259019
Imagem EX1	90593	3055135	19921	177013	70672	2878122	2949	1662331	87644	1392804
Imagem EX2	358928	869872	6618	624502	352310	245370	1535	630049	357393	239823
Imagem EX3	32897	274303	22240	157426	10657	116877	17703	237404	15194	36899
Imagem EX4	18752	288448	2132	212004	16620	76444	176	232935	18576	55513
Imagem EX5	43220	263980	17324	46434	25896	217546	8038	54347	35182	209633
Imagem EX6	100302	1128498	77802	843	22500	1127655	57667	1625	42635	1126873
Imagem EX7	14475	1214325	11779	156	2696	1214169	11946	1526	2529	1212799
Imagem EX8	22017	1206783	8526	67817	13491	1138966	9038	87204	12979	1119579
Imagem EX9	0	1228800	0	26410	0	1202390	0	31346	0	1197454
Imagem EX10	0	1228800	0	467540	0	761260	0	397138	0	831662
Imagem EX11	0	1228800	0	55901	0	1172899	0	222441	0	1006359
Imagem EX12	0	1228800	0	10999	0	1217801	0	153351	0	1075449

Tabela 6.1: Comparação de resultados quantitativamente.

é pele e que é segmentada como sendo (ERRO PELE) é menor que os resultados deste trabalho, mas a quantidade de pixel que é pele e o programa segmenta como não sendo pele (ERRO NÃO PELE) é maior que os resultados deste trabalho, como mostra a tabela 6.2.

De uma forma geral, este trabalho obteve resultados satisfatórios pelo fato de demonstrar que a segmentação de pele humana pode ser melhor quando não são fixados valores, parâmetros, tamanhos ou quaisquer variáveis que sirvam para o auxílio na segmentação.

## 6.5 Trabalhos Futuros

Os resultados deste trabalho podem servir de base para a realização de muitos outros em que se pretende segmentar pele humana em imagens coloridas. Assim, a segmentação pode ser melhorada utilizando outros modelos de cores em relação a outros métodos de classificação, conforme mencionado no capítulo 4.

Como exemplo podemos citar a utilização e comparação de duas ou mais arquiteturas de redes neurais artificiais, neste mesmo modelo de cor, onde o conjunto

IMAGENS	SEGMENT.MANUAL		SEGMENTAÇÃO PROGRAMA							
	QUANTIDADE PIXELS ENCONTRADOS		VALORES EM PORCENTAGEM - ESTE TRABALHO				VALORES EM PORCENTAGEM - TRABALHO KOVAC			
	PELE	NÃO PELE	ERRO		ACERTO		ERRO		ACERTO	
			PELE	NÃO PELE	PELE	NÃO PELE	PELE	NÃO PELE	PELE	NÃO PELE
Imagem 01	101528	205672	<b>65,93</b>	2,01	34,07	<b>97,99</b>	3,96	<b>4,02</b>	<b>96,04</b>	95,98
Imagem 02	7955	299245	<b>96,92</b>	0,58	3,08	<b>99,42</b>	26,98	<b>1,79</b>	<b>73,02</b>	98,21
Imagem 03	9975	297225	38,77	<b>8,74</b>	<b>61,23</b>	91,26	<b>79,10</b>	4,98	20,90	<b>95,02</b>
Imagem 04	6503	300697	<b>72,92</b>	<b>2,21</b>	27,08	97,79	64,80	0,19	<b>35,20</b>	<b>99,81</b>
Imagem 05	21927	285273	<b>56,00</b>	<b>2,15</b>	44,00	97,85	44,61	1,18	<b>55,39</b>	<b>98,82</b>
Imagem 06	537	306663	<b>91,43</b>	<b>1,78</b>	8,57	98,22	40,41	0,03	<b>59,59</b>	<b>99,97</b>
Imagem 07	14352	292848	<b>88,67</b>	0,23	11,33	99,77	24,43	<b>1,87</b>	<b>75,57</b>	<b>98,13</b>
Imagem 08	21278	285922	<b>93,42</b>	2,48	6,58	<b>97,52</b>	7,28	<b>12,25</b>	<b>92,72</b>	87,75
Imagem 09	24998	282202	<b>34,87</b>	0,19	65,13	<b>99,81</b>	8,50	<b>4,49</b>	<b>91,50</b>	95,51
Imagem 10	26340	280860	70,14	<b>4,82</b>	<b>29,86</b>	95,18	<b>76,64</b>	1,95	23,36	<b>98,05</b>
Imagem 11	23981	283219	<b>86,28</b>	0,81	13,72	<b>99,19</b>	27,44	<b>3,27</b>	<b>72,56</b>	96,73
Imagem 12	5302	301898	<b>63,90</b>	<b>0,34</b>	36,10	99,66	23,93	0,08	<b>76,07</b>	<b>99,92</b>
Imagem 13	64290	242910	12,42	2,89	<b>87,58</b>	<b>97,11</b>	<b>15,19</b>	<b>8,22</b>	84,81	91,78
Imagem 14	7087	300113	<b>33,54</b>	4,21	66,46	<b>95,79</b>	28,29	<b>4,51</b>	<b>71,71</b>	95,49
Imagem 15	39324	267876	<b>52,17</b>	1,93	47,83	<b>98,07</b>	21,34	<b>4,73</b>	<b>78,66</b>	95,27
Imagem 16	25246	281954	<b>44,15</b>	0,78	55,85	<b>99,22</b>	29,07	<b>20,20</b>	<b>70,93</b>	79,80
Imagem 17	75449	231751	<b>52,16</b>	2,04	47,84	<b>97,96</b>	11,63	<b>10,98</b>	<b>88,37</b>	89,02
Imagem 18	12017	295183	<b>54,56</b>	0,01	45,44	<b>99,99</b>	40,38	<b>0,31</b>	<b>59,62</b>	99,69
Imagem 19	73008	234192	<b>68,18</b>	0,10	31,82	<b>99,90</b>	31,76	<b>4,58</b>	<b>68,24</b>	95,42
Imagem 20	11252	295948	<b>78,89</b>	4,13	21,11	<b>95,87</b>	33,47	<b>12,48</b>	<b>66,53</b>	87,52
Imagem EX1	90593	3055135	<b>21,99</b>	5,79	78,01	<b>94,21</b>	3,26	<b>54,41</b>	<b>96,74</b>	45,59
Imagem EX2	358928	869872	<b>1,84</b>	71,79	98,16	<b>28,21</b>	0,43	<b>72,43</b>	<b>99,57</b>	27,57
Imagem EX3	32897	274303	<b>67,60</b>	57,39	32,40	<b>42,61</b>	53,81	<b>86,55</b>	<b>46,19</b>	13,45
Imagem EX4	18752	288448	<b>11,37</b>	73,50	88,63	<b>26,50</b>	0,94	<b>80,75</b>	<b>99,06</b>	19,25
Imagem EX5	43220	263980	<b>40,08</b>	17,59	59,92	<b>82,41</b>	18,60	<b>20,59</b>	<b>81,40</b>	79,41
Imagem EX6	100302	1128498	<b>77,57</b>	0,07	22,43	<b>99,93</b>	57,49	<b>0,14</b>	<b>42,51</b>	99,86
Imagem EX7	14475	1214325	81,37	0,01	<b>18,63</b>	<b>99,99</b>	<b>82,53</b>	<b>0,13</b>	17,47	99,87
Imagem EX8	22017	1206783	38,72	5,62	<b>61,28</b>	<b>94,38</b>	<b>41,05</b>	<b>7,23</b>	58,95	92,77
Imagem EX9	0	1228800	0,00	2,15	0,00	<b>97,85</b>	0,00	<b>2,55</b>	0,00	97,45
Imagem EX10	0	1228800	0,00	<b>38,05</b>	0,00	61,95	0,00	32,32	0,00	<b>67,68</b>
Imagem EX11	0	1228800	0,00	4,55	0,00	<b>95,45</b>	0,00	<b>18,10</b>	0,00	81,90
Imagem EX12	0	1228800	0,00	0,90	0,00	<b>99,10</b>	0,00	<b>12,48</b>	0,00	87,52

Tabela 6.2: Comparação de resultados de maneira percentual.

de dados para o treinamento destas poderiam ser, inclusive, o resultado obtido pelo crescimento da região em torno de um pixel (ou de um conjunto destes) conforme os resultados deste procedimento.

A quantidade de imagens geradas por este trabalho em conjunto com os resultados obtidos pelo crescimento da região em torno do pixel e a utilização de pelo menos duas arquiteturas de redes neurais, seria uma possibilidade de melhora dos resultados alcançados até o momento.



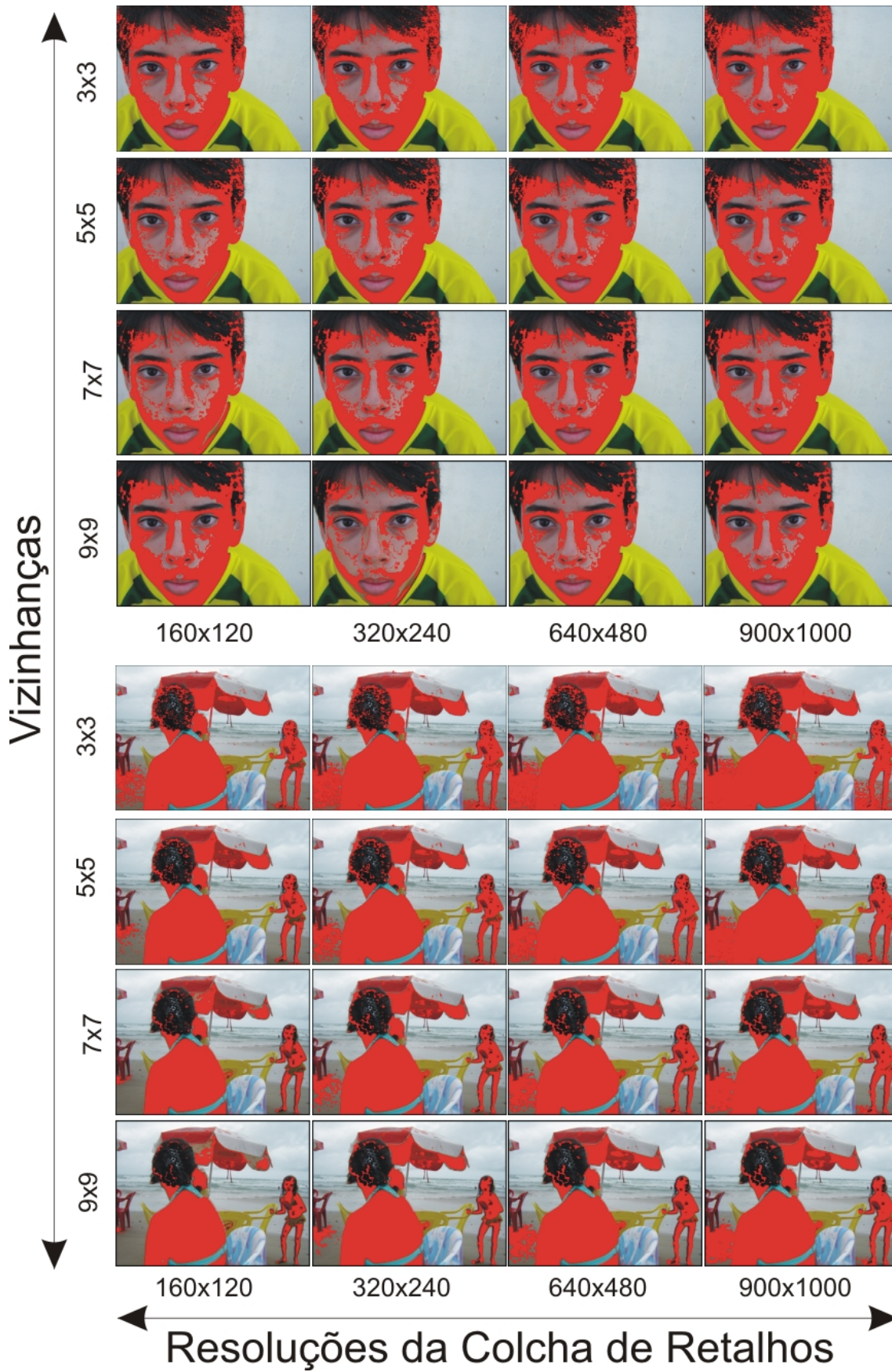


Figura 6.7:

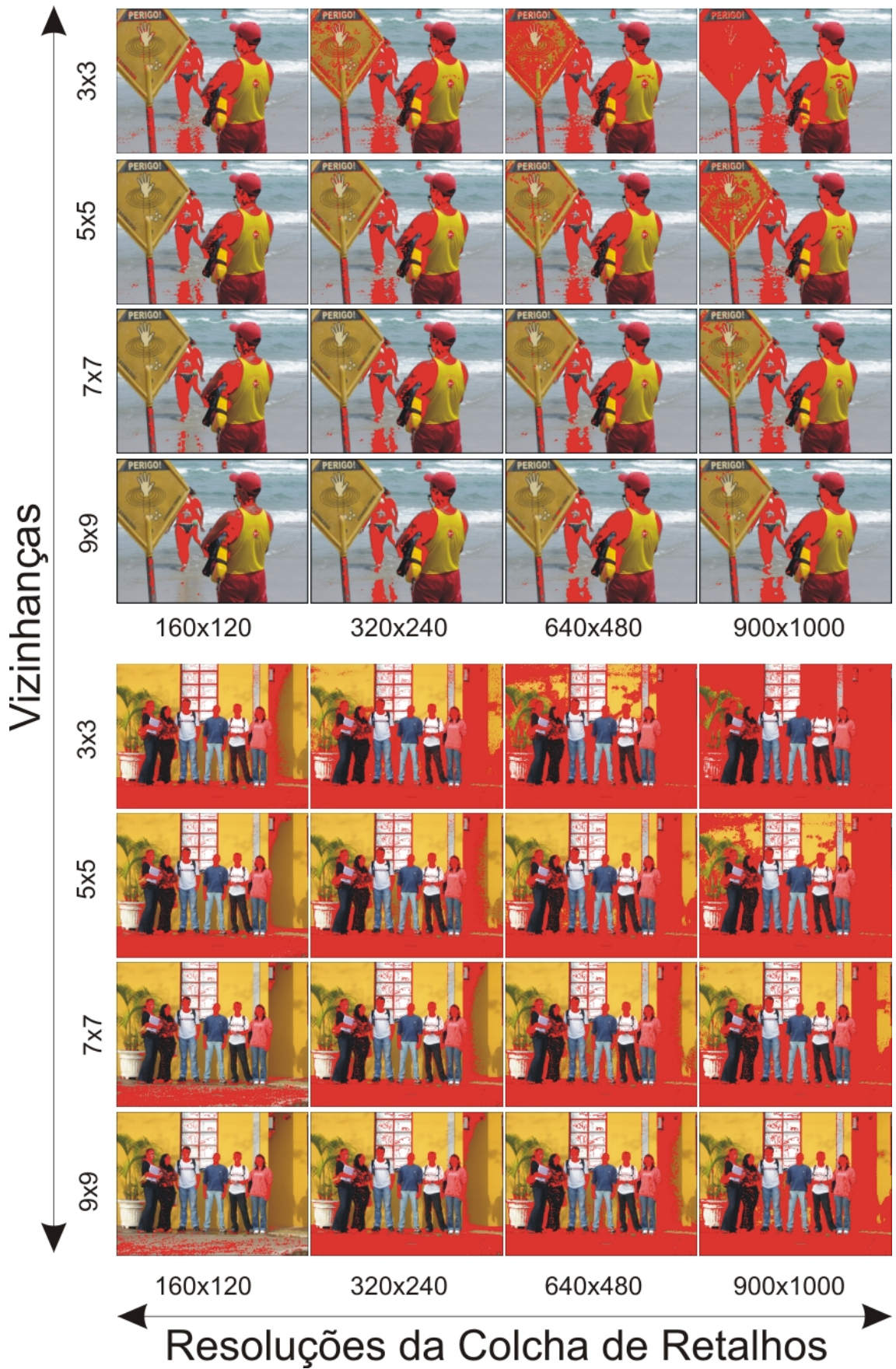


Figura 6.8:



## Resolução das Imagens Adquiridas

Figura 6.9: Resultado das imagens segmentadas conforme Kovac et al. (2003)

Segmentação conforme equações de Jure Kovac:



Segmentação proposta por este trabalho:



Subimagem  
9x9  
Resolução  
160x120

Subimagem  
9x9  
Resolução  
160x120

Subimagem  
9x9  
Resolução  
160x120

Subimagem  
9x9  
Resolução  
160x120

Figura 6.10: Comparação de resultados de forma visual.

# Referências Bibliográficas

- BERGASA, L.; MAZO, M.; GARDEL, A.; SOTELO, M. ; BOQUETE, L. (2000). Unsupervised and adaptive gaussian skin-color model. *Image Vision Comput* 987–1003. 22
- BOVIK, A. (2000). *Handbook of Image and Video Processing*. Academic Press. 35
- BRAND, J. ; MASON, J. (2000). A comparative assessment of three approaches to pixel level human skin-detection. *ICPR01* **1**, 1056–1059. 23
- BROWN, D.; CRAW, I. ; LEWTHWAITE, J. (2001). A som based approach to skin detection with application in real time systems. *BMVC01* . 22, 25
- CAETANO, T. ; BARONE, D. (2001). A probabilistic model for the human skin-color. *ICIAP01* 279–283. 22
- CAETANO, T.; OLABARRIAGA, S. ; BARONE, D. (2003). Do mixture models in chromaticity space improve skin detection? *Pattern Recognition* **36**, 3019–3021. 24
- CAI, J. ; GOSHTASBY, A. (1999). Detecting human faces in color images. *Image Vision Comput.* **18**, 63–75. 22
- CARELLI, G. (2006). Questões de pele. *Revista Exame* 97–102. 1
- CHAI, D. ; BOUZERDOUM, A. (2000). A bayesian approach to skin color classification in ycbcr color space. *IEEE TENCON00* **2**, 421–424. 22, 23
- CHAI, D. ; NGAN, K. (1999). Face segmentation using skin-color map in videophone applications. *IEEE Trans. Circuits Syst. Video Technol.* **9**. 23
- CHEN, C. ; CHIANG, S. (1997). Detection of human faces in colour images. *IEE Proc. Vision Image Signal Process* **144**, 384–388. 25
- CORTEZ, P.; DA SILVA, W. ; DE OLIVEIRA, T. (2006). Segmentação da pele humana em imagens de vídeo utilizando wavelet e redes neurais . 1
- D. CHAI, K. N. (1998). Locating facial region of a head-and-shoulders color image. *ICFGR98* . 22
- DAI, Y. ; NAKANO, Y. (1996). Face-texture model based on sgld and its application in face detection in a color scene. *Pattern Recognition* **29**, 1007–1017. 23
- FLECK, M.; FORSYTH, D. ; BREGLER, C. (1996). Finding naked people. *Proceedings of European Conference on Computer Vision* **2**, 592–602. 23

- FRASER, B.; MURPHY, C. ; BUNTING, F. (2005). *Real World Color Management*. Peachpit Press, 2th edition ed. 16
- FU, Z.; YANG, J.; HU, W. ; TAN, T. (2004). Mixture clustering using multidimensional histograms for skin detection. *ICPR04* 549–552. 27
- GARCIA, C. ; TZIRITAS, G. (1999). Face detection using quantized skin color regions merging and wavelet packet analysis. *IEEE Trans. Multimedia* **1**, 264–277. 22
- GOMEZ, G. ; MORALES, E. (2002). Automatic feature construction and a simple rule induction algorithm for skin detection. *Proceedings of Workshop on Machine Learning in Computer Vision* 31–38. 23
- GOMEZ, G.; SANCHEZ, M. ; SUCAR, L. (2002). On selecting an appropriate colour space for skin detection. *Springer-Verlag: Lecture Notes in Artificial Intelligence* **2313**, 70–79. 23
- GONZALES, R. C. ; WOODS, R. E. (1992). *Processamento de Imagens Digitais*. Edgar Blücher. ix, 5, 12, 13, 14, 19, 20
- GONZALES, R. C. ; WOODS, R. E. (2001). *Digital Image Processing*. Prentice Hall. 16
- GREENSPAN, H.; GOLDBERGER, J. ; ESHET, I. (2001). Mixture model for facecolor modeling and segmentation. *Pattern Recognition Lett.* **22**, 1525–1536. 24
- HSU, R.; ABDEL-MOTTALEB, M. ; JAIN, A. (2002). Face detection in color images. *IEEE Trans. Pattern Anal. Machine Intell.* **24**, 696–706. 22
- JAHNE, B. (2002). *Digital Image Processing*. Springer, 5th revised and extended edition ed. 4
- JAYARAM, S.; SCHMUGGE, S.; SHIN, M. ; TSAP, L. (2004). Effect of color space transformation, the illuminance component, and color modeling on skin detection. *CVPR04* 813–818. 27
- JONES, M. ; REHG, J. (1999). Statistical color models with application to skin detection. *CVPR99* . 24
- JONES, M. ; REHG, J. (2002). Statistical color models with application to skin detection. *J. Comput. Vision* **46**, 81–96. 24, 27
- J. YANG ; A. WAIBEL (1996). A real-time face tracker. *Proceedings of the third IEEE Workshop on Applications of Computer Vision - WACV96* . 24
- KAKUMANU, P.; MAKROGIANNIS, S. ; BOURBAKIS, N. (2006). A survey of skin-color modeling and detection methods. *PATTERN RECOGNITION* . 22
- KARLEKAR, J. ; DESAI, U. (1999). Finding faces in color images using wavelet transform. *International Conference on Image Analysis and Processing* . 25

- KAWATO, S. ; OHYA, J. (2000). Automatic skin-color distribution extraction for face detection and tracking. *Fifth International Conference on Signal Processing* **2**, 1415–1418. 22
- KIM, S.; KIM, N.; AHN, S. ; KIM, H. (1998). Object oriented face detection using range and color information. *AFGR98* . 22
- KOVAC, J.; PEER, P. ; SOLINA, F. (2003). Human skin colour clustering for face detection. *EUROCON* . v, x, 27, 40, 46, 47, 52
- K.W.WONG; LAM, K. ; SIU, W. (2003). A robust scheme for live detection of human faces in color images. *Signal Process. Image Commun.* **18**, 103–114. 22
- LEÃO, A.; ARAÚJO, A. ; SOUZA, L. (2005). Implementação de sistemas de gerenciamento de cores para imagens digitais. . . ix, 18
- LEE, J. ; YOO, S. (2002). An elliptical boundary model for skin color detection. *Proceedings of the International Conference on Imaging Science, Systems and Technology* . 24
- MARCEL, S. ; BENGIO, S. (2002). Improving face verification using skin color information. *ICPR02* . 23
- MARTINEZ, A. ; BENAVENTE, R. (1998). The ar face database. , CVC Technical Report. 29, 33
- MCKENNA, S.; GONG, S. ; RAJA, Y. (1998). Modeling facial colour and identity with gaussian mixtures. *Pattern Recognition* **31**, 1883–1892. 22, 24
- OLIVER, N.; PENTLAND, A. ; BERARD, F. (1997). Lafter - lips and face real time tracker. *CVPR97* . 22, 24
- PHUNG, S.; BOUZERDOUM, A. ; CHAI, D. (2005). Skin segmentation using color pixel classification: analysis and comparison. *IEEE Trans. Pattern Anal. Mach. Intell.* **27**. 27
- PHUNG, S.; CHAI, D. ; BOUZERDOUM, A. (2001). A universal and robust human skin color model using neural networks. *IJCNN01* . 25
- SAHBI, H. ; BOUJEMAA, N. (2002). Coarse to fine face detection based on skin color adaptation. *Workshop on Biometric Authentication Lecture Notes in Computer Science* **2359**, 112–120. 25
- SAXE, D. ; FOULDS, R. (1996). Toward robust skin identification in video images. *AFGR96* . 22
- SCHWERDT, K. ; CROWELY, J. (2000). Robust face tracking using color. *AFGR00* . 22, 23
- SEBE, N.; COHEN, T.; HUANG, T. ; GEVERS, T. (2004). Skin detection - a bayesian network approach. *ICPR04* . 22

- SEOW, M.; VALAPARLA, D. ; ASARI, V. (2003). Neural network-based skin color model for face detection. *Proceedings of the 32nd Workshop on Applied Imagery Pattern Recognition* . 25
- SHARMA, G. (2003). *Imaging Handbook*. CRC Press. 3, 18
- SIGAL, L.; SCLAROFF, S. ; ATHITSOS, V. (2004). Skin color-based video segmentation under time-varying illumination. *IEEE Trans. Pattern Anal. Mach. Intell.* **26**. 23
- SOBOTKA, K. ; PITAS, I. (1996). Extraction of facial regions and features using color and shape information. *ICPR96* . 23
- SOBOTKA, K. ; PITAS, I. (1998). A novel method for automatic face segmentation, facial feature extraction and tracking. *Signal Process. Image Commun.* **12**, 263–281. 22, 23
- SORIANO, M.; MARTINKAUPPI, J.; HUOVINEN, S. ; LÄÄKSONEN, M. (2003). Adaptive skin color modeling using the skin locus for selecting training pixels. *Pattern Recognition* **36**, 681–690. 22
- SRISUK, S. ; KURUTACH, W. (2002). New robust face detection in color images. *AFGR02* 291–296. 23
- STÖRRING, M.; KOÈKA, T.; ANDERSON, H. ; GRANUM, E. (2003). Tracking regions of human skin through illumination changes. *Pattern Recognition Lett* **24**. 22
- THU, Q.; MEGURO, M. ; KANEKO, M. (2002). Skin-color extraction in images with complex background and varying illumination. *Sixth IEEE Workshop on Applications of Computer Vision* . 22, 24
- WANG, J. ; SUNG, E. (1999). Frontal-view face detection and facial feature extraction using color and morphological operations. *Pattern Recognition Lett* **20**, 1053–1068. 22
- WANG, Y. ; YUAN, B. (2001). A novel approach for human face detection from color images under complex background. *Pattern Recognition* **34**, 1983–1992. 22, 23
- YANG, J.; LU, W. ; WAIBEL, A. (1998). Skin-color modeling and adaptation. *ACCV98* . 22, 24
- YANG, M. ; AHUJA, N. (1999). Gaussian mixture model for human skin color and its application in image and video databases. *Proceedings of SPIE - Conference on Storage and Retrieval for Image and Video Databases* **3656**, 458–466. 22, 23, 24
- ZARIT, B.; SUPER, J. ; QUEK, F. (1999). Comparison of five color models in skin pixel classification. *ICCV99* . 23
- ZHU, Q.; CHENG, K.-T.; WU, C.-T. ; WU, Y.-L. (2004). Adaptive learning of an accurate skin-color model. *AFGR04* . 22

# Anexo A



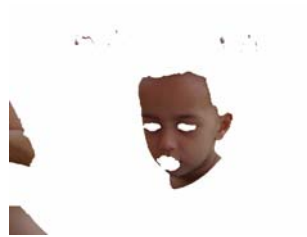
Imagem Original

Imag. Segm. Manual

Imag. Segmentação deste trabalho

Imag. Segmentação de J.Kovac

15



16



17



18



19



20



EX1



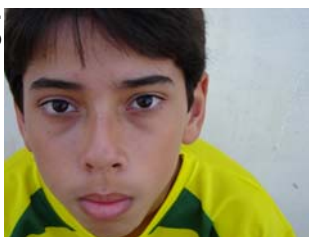
Imagem Original

Imag. Segm.Manual

Imag. Segmentação deste trabalho

Imag. Segmentação de J.Kovac

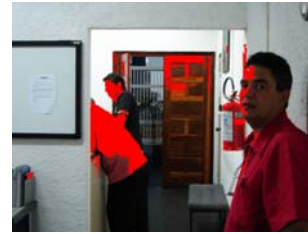
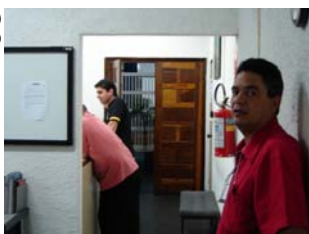
01



02



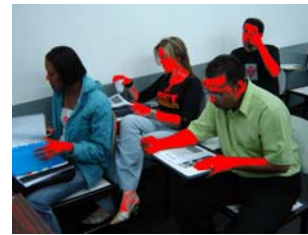
03



04



05



06



07



Imagem Original

Imag. Segm.Manual

Imag. Segmentação deste trabalho

Imag. Segmentação de J.Kovac

EX8



EX9



EX10



EX11



EX12



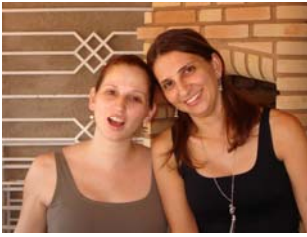
Imagem Original

Imag. Segm.Manual

Imag. Segmentação  
deste trabalho

Imag. Segmentação  
de J.Kovac

EX2



EX3



EX4



EX5



EX6



EX7

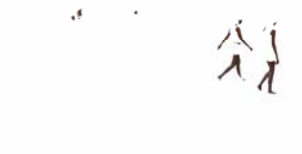


Imagem Original

Imag. Segm.Manual

Imag. Segmentação deste trabalho

Imag. Segmentação de J.Kovac

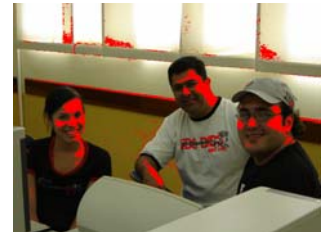
08



09



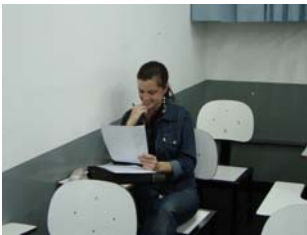
10



11



12



13



14



# Anexo B

```
% -----  
% PROGRAMA 05 - VARRE IMAGEM 7x7 - 1000x900  
% Segmenta Pixels PELE/NAO PELE conforme Media encontrada  
% -----  
  
clc;  
clear all;  
close all;  
  
[x,u]=uigetfile('*.bmp','Selecione a imagem a ser varrida...');  
tempo_inicio = clock;  
arq=[u,x];  
[I]=imread(arq,'bmp');  
  
IH=rgb2hsv(I);  
[L,C,Q]=size(IH);  
R=imresize(IH,[L,C]);  
Z=double(R);  
S=squeeze(Z);  
WSC = S;  
  
INovoH = zeros(L,C,Q);  
INovoR = zeros(L,C,Q);  
  
QLF = L-6;  
QCF = C-6;  
CB = 0;  
PLC = 0;  
PCC = 0;  
WSC = 0;  
SCN = 0;  
WLC = 0;  
WCC = 0;  
  
for LI = 1:QLF  
    for CI = 1:QCF  
  
        ContPIX = 0;  
        SomaELC = 0;  
        MMatELC = 0;  
        MedSomaELC = 0;  
        MedMMatELC = 0;  
        DifMediELC = 0;  
        DesvPadELC = 0;  
  
        for LC = LI:LI+6  
            PLC = PLC + 1;  
  
            for CC = CI:CI+6  
                PCC = PCC + 1;  
  
                for PX = 1:3  
  
                    if PX==1  
                        SCN = (S(LC,CC,PX));  
                        SomaELC = (SomaELC + SCN);
```

```
        MMatELC = MMatELC + (SCN * SCN);
    end

end

if (PLC == 4 & PCC == 4)
    WLC = LC;
    WCC = CC;
    WSC = S(LC,CC,:);
    INovoH(WLC,WCC,:) = WSC;
end

ContPIX = ContPIX + 1;

end
PCC = 0;
end
PLC = 0;

MedSomaELC = (SomaELC / ContPIX);
MedMMatELC = (MMatELC / ContPIX);
DifMediELC = (MedMMatELC - MedSomaELC.^2);
DesvPadELC = (DifMediELC^0.5);

if (MedSomaELC >= 0.0186 & MedSomaELC <= 0.1147)
    INovoH(WLC,WCC,1) = 0.0000;
    INovoH(WLC,WCC,2) = 1;
    INovoH(WLC,WCC,3) = 1;
end

end
end

INovoR=hsv2rgb(INovoH);
cabe2 = sprintf('ImagemXX - Vizinhanca 7x7.bmp');
imshow(INovoR);title(cabe2);
imwrite(INovoR,'ImagemXX - Vizinhanca 7x7.bmp');
```



```
% -----  
% PROGRAMA 05 - VARRE IMAGEM 5x5 - 1000x900  
% Segmenta Pixels PELE/NAO PELE conforme Media encontrada  
% -----  
  
clc;  
clear all;  
close all;  
  
[x,u]=uigetfile('*.bmp','Selecione a imagem a ser varrida...');  
tempo_inicio = clock;  
arq=[u,x];  
[I]=imread(arq,'bmp');  
  
IH=rgb2hsv(I);  
[L,C,Q]=size(IH);  
R=imresize(IH,[L,C]);  
Z=double(R);  
S=squeeze(Z);  
WSC = S;  
  
INovoH = zeros(L,C,Q);  
INovoR = zeros(L,C,Q);  
  
QLF = L-4;  
QCF = C-4;  
CB = 0;  
PLC = 0;  
PCC = 0;  
WSC = 0;  
SCN = 0;  
WLC = 0;  
WCC = 0;  
  
for LI = 1:QLF  
    for CI = 1:QCF  
  
        ContPIX = 0;  
        SomaELC = 0;  
        MMatELC = 0;  
        MedSomaELC = 0;  
        MedMMatELC = 0;  
        DifMediELC = 0;  
        DesvPadELC = 0;  
  
        for LC = LI:LI+4  
            PLC = PLC + 1;  
  
            for CC = CI:CI+4  
                PCC = PCC + 1;  
  
                for PX = 1:3  
  
                    if PX==1  
                        SCN = (S(LC,CC,PX));  
                        SomaELC = (SomaELC + SCN);
```

```
        MMatELC = MMatELC + (SCN * SCN);
    end

end

if (PLC == 3 & PCC == 3)
    WLC = LC;
    WCC = CC;
    WSC = S(LC,CC,:);
    INovoH(WLC,WCC,:) = WSC;
end

    ContPIX = ContPIX + 1;

end
PCC = 0;
end
PLC = 0;

MedSomaELC = (SomaELC / ContPIX);
MedMMatELC = (MMatELC / ContPIX);
DifMediELC = (MedMMatELC - MedSomaELC.^2);
DesvPadELC = (DifMediELC^0.5);

if (MedSomaELC >= 0.0152 & MedSomaELC <= 0.1188)
    INovoH(WLC,WCC,1) = 0.0000;
    INovoH(WLC,WCC,2) = 1;
    INovoH(WLC,WCC,3) = 1;
end

end

end

INovoR=hsv2rgb(INovoH);
cabe2 = sprintf('ImagemXX - Vizinhanca 5x5.bmp' );
imshow(INovoR);title(cabe2);
imwrite(INovoR,'ImagemXX - Vizinhanca 5x5.bmp');
```

```
% -----  
% PROGRAMA 05 - VARRE IMAGEM 3X3-1000X900  
% Segmenta Pixels PELE/NAO PELE conforme Media encontrada  
% -----  
  
clc;  
clear all;  
close all;  
  
[x,u]=uigetfile('*.bmp','Selecione a imagem a ser varrida...');  
tempo_inicio = clock;  
arq=[u,x];  
[I]=imread(arq,'bmp');  
  
IH=rgb2hsv(I);  
[L,C,Q]=size(IH);  
R=imresize(IH,[L,C]);  
Z=double(R);  
S=squeeze(Z);  
WSC = S;  
  
INovoH = zeros(L,C,Q);  
INovoR = zeros(L,C,Q);  
  
QLF = L-2;  
QCF = C-2;  
CB = 0;  
PLC = 0;  
PCC = 0;  
WSC = 0;  
SCN = 0;  
WLC = 0;  
WCC = 0;  
  
for LI = 1:QLF  
    for CI = 1:QCF  
  
        ContPIX = 0;  
        SomaELC = 0;  
        MMatELC = 0;  
        MedSomaELC = 0;  
        MedMMatELC = 0;  
        DifMediELC = 0;  
        DesvPadELC = 0;  
  
        for LC = LI:LI+2  
            PLC = PLC + 1;  
  
            for CC = CI:CI+2  
                PCC = PCC + 1;  
  
                for PX = 1:3  
  
                    if PX==1  
                        SCN = (S(LC,CC,PX));  
                        SomaELC = (SomaELC + SCN);
```

```
        MMatELC = MMatELC + (SCN * SCN);
    end

end

if (PLC == 2 & PCC == 2)
    WLC = LC;
    WCC = CC;
    WSC = S(LC,CC,:);
    INovoH(WLC,WCC,:) = WSC;
end

    ContPIX = ContPIX + 1;

end
PCC = 0;
end
PLC = 0;

MedSomaELC = (SomaELC / ContPIX);
MedMMatELC = (MMatELC / ContPIX);
DifMediELC = (MedMMatELC - MedSomaELC.^2);
DesvPadELC = (DifMediELC^0.5);

if (MedSomaELC >= 0.0128 & MedSomaELC <= 0.1320)
    INovoH(WLC,WCC,1) = 0.0000;
    INovoH(WLC,WCC,2) = 1;
    INovoH(WLC,WCC,3) = 1;
end

end
end

INovoR=hsv2rgb(INovoH);
cabe2 = sprintf('Medias Utilizadas 1000x900 - Imagem Origem 0B-21M %dx%d .bmp',C,L);
imshow(INovoR);title(cabe2);
imwrite(INovoR,'Medias Utilizadas 1000x900 - Imagem Origem 0B-21M.bmp');
```

```
% -----  
% PROGRAMA 05 - VARRE IMAGEM 3X3-640x480  
% Segmenta Pixels PELE/NAO PELE conforme Media encontrada  
% -----  
  
clc;  
clear all;  
close all;  
  
[x,u]=uigetfile('*.bmp','Selecione a imagem a ser varrida...');  
tempo_inicio = clock;  
arq=[u,x];  
[I]=imread(arq,'bmp');  
  
IH=rgb2hsv(I);  
[L,C,Q]=size(IH);  
R=imresize(IH,[L,C]);  
Z=double(R);  
S=squeeze(Z);  
WSC = S;  
  
INovoH = zeros(L,C,Q);  
INovoR = zeros(L,C,Q);  
  
QLF = L-2;  
QCF = C-2;  
CB = 0;  
PLC = 0;  
PCC = 0;  
WSC = 0;  
SCN = 0;  
WLC = 0;  
WCC = 0;  
  
for LI = 1:QLF  
    for CI = 1:QCF  
  
        ContPIX = 0;  
        SomaELC = 0;  
        MMatELC = 0;  
        MedSomaELC = 0;  
        MedMMatELC = 0;  
        DifMediELC = 0;  
        DesvPadELC = 0;  
  
        for LC = LI:LI+2  
            PLC = PLC + 1;  
  
            for CC = CI:CI+2  
                PCC = PCC + 1;  
  
                for PX = 1:3  
  
                    if PX==1  
                        SCN = (S(LC,CC,PX));  
                        SomaELC = (SomaELC + SCN);
```

```
        MMatELC = MMatELC + (SCN * SCN);
    end

end

if (PLC == 2 & PCC == 2)
    WLC = LC;
    WCC = CC;
    WSC = S(LC,CC,:);
    INovoH(WLC,WCC,:) = WSC;
end

    ContPIX = ContPIX + 1;

end
PCC = 0;
end
PLC = 0;

MedSomaELC = (SomaELC / ContPIX);
MedMMatELC = (MMatELC / ContPIX);
DifMediELC = (MedMMatELC - MedSomaELC.^2);
DesvPadELC = (DifMediELC^0.5);

if (MedSomaELC >= 0.0169 & MedSomaELC <= 0.1197)
    INovoH(WLC,WCC,1) = 0.0000;
    INovoH(WLC,WCC,2) = 1;
    INovoH(WLC,WCC,3) = 1;
end

end
end

INovoR=hsv2rgb(INovoH);
cabe2 = sprintf('Medias Utilizadas 640X480 - Imagem Origem 0A-1A %dx%d .bmp',C,L);
imshow(INovoR);title(cabe2);
imwrite(INovoR,'Medias Utilizadas 640X480 - Imagem Origem 0A-1A.bmp');
```

```
% -----  
% PROGRAMA 05 - VARRE IMAGEM 3X3-320x240  
% Segmenta Pixels PELE/NAO PELE conforme Media encontrada  
% -----  
  
clc;  
clear all;  
close all;  
  
[x,u]=uigetfile('*.bmp','Selecione a imagem a ser varrida...');  
tempo_inicio = clock;  
arq=[u,x];  
[I]=imread(arq,'bmp');  
  
IH=rgb2hsv(I);  
[L,C,Q]=size(IH);  
R=imresize(IH,[L,C]);  
Z=double(R);  
S=squeeze(Z);  
WSC = S;  
  
INovoH = zeros(L,C,Q);  
INovoR = zeros(L,C,Q);  
  
QLF = L-2;  
QCF = C-2;  
CB = 0;  
PLC = 0;  
PCC = 0;  
WSC = 0;  
SCN = 0;  
WLC = 0;  
WCC = 0;  
  
for LI = 1:QLF  
    for CI = 1:QCF  
  
        ContPIX = 0;  
        SomaELC = 0;  
        MMatELC = 0;  
        MedSomaELC = 0;  
        MedMMatELC = 0;  
        DifMediELC = 0;  
        DesvPadELC = 0;  
  
        for LC = LI:LI+2  
            PLC = PLC + 1;  
  
            for CC = CI:CI+2  
                PCC = PCC + 1;  
  
                for PX = 1:3  
  
                    if PX==1  
                        SCN = (S(LC,CC,PX));  
                        SomaELC = (SomaELC + SCN);
```

```
        MMatELC = MMatELC + (SCN * SCN);
    end

end

if (PLC == 2 & PCC == 2)
    WLC = LC;
    WCC = CC;
    WSC = S(LC,CC,:);
    INovoH(WLC,WCC,:) = WSC;
end

    ContPIX = ContPIX + 1;

end
PCC = 0;
end
PLC = 0;

MedSomaELC = (SomaELC / ContPIX);
MedMMatELC = (MMatELC / ContPIX);
DifMediELC = (MedMMatELC - MedSomaELC.^2);
DesvPadELC = (DifMediELC^0.5);

if (MedSomaELC >= 0.0197 & MedSomaELC <= 0.1147)
    INovoH(WLC,WCC,1) = 0.0000;
    INovoH(WLC,WCC,2) = 1;
    INovoH(WLC,WCC,3) = 1;
end

end
end

INovoR=hsv2rgb(INovoH);
cabe2 = sprintf('Medias Utilizadas 320x240 - Imagem Origem 0B-21M %dx%d .bmp',C,L);
imshow(INovoR);title(cabe2);
imwrite(INovoR,'Medias Utilizadas 320x240 - Imagem Origem 0B-21M.bmp');
```



```
% -----  
% PROGRAMA 05 - VARRE IMAGEM 3X3-160x120  
% Segmenta Pixels PELE/NAO PELE conforme Media encontrada  
% -----  
  
clc;  
clear all;  
close all;  
  
[x,u]=uigetfile('*.bmp','Selecione a imagem a ser varrida...');  
tempo_inicio = clock;  
arq=[u,x];  
[I]=imread(arq,'bmp');  
  
IH=rgb2hsv(I);  
[L,C,Q]=size(IH);  
R=imresize(IH,[L,C]);  
Z=double(R);  
S=squeeze(Z);  
WSC = S;  
  
INovoH = zeros(L,C,Q);  
INovoR = zeros(L,C,Q);  
  
QLF = L-2;  
QCF = C-2;  
CB = 0;  
PLC = 0;  
PCC = 0;  
WSC = 0;  
SCN = 0;  
WLC = 0;  
WCC = 0;  
  
for LI = 1:QLF  
    for CI = 1:QCF  
  
        ContPIX = 0;  
        SomaELC = 0;  
        MMatELC = 0;  
        MedSomaELC = 0;  
        MedMMatELC = 0;  
        DifMediELC = 0;  
        DesvPadELC = 0;  
  
        for LC = LI:LI+2  
            PLC = PLC + 1;  
  
            for CC = CI:CI+2  
                PCC = PCC + 1;  
  
                for PX = 1:3  
  
                    if PX==1  
                        SCN = (S(LC,CC,PX));  
                        SomaELC = (SomaELC + SCN);
```

```
        MMatELC = MMatELC + (SCN * SCN);
    end

end

if (PLC == 2 & PCC == 2)
    WLC = LC;
    WCC = CC;
    WSC = S(LC,CC,:);
    INovoH(WLC,WCC,:) = WSC;
end

    ContPIX = ContPIX + 1;

end
PCC = 0;
end
PLC = 0;

MedSomaELC = (SomaELC / ContPIX);
MedMMatELC = (MMatELC / ContPIX);
DifMediELC = (MedMMatELC - MedSomaELC.^2);
DesvPadELC = (DifMediELC^0.5);

if (MedSomaELC >= 0.0204 & MedSomaELC <= 0.1045)
    INovoH(WLC,WCC,1) = 0.0000;
    INovoH(WLC,WCC,2) = 1;
    INovoH(WLC,WCC,3) = 1;
end

end
end

INovoR=hsv2rgb(INovoH);
cabe2 = sprintf('Medias Utilizadas 160X120 - Imagem Origem 0A-11A %dx%d .bmp',C,L);
imshow(INovoR);title(cabe2);
imwrite(INovoR,'Medias Utilizadas 160X120 - Imagem Origem 0A-11A.bmp');
```

```

% -----
% PROGRAMA 04 - CALCULO 9x9
% Encontra os valores das Medias em SubImagem tamanho 9x9
% -----

clc;
clear all;
close all;

[x,u]=uigetfile('*.bmp','Selecione a imagem...');
tempo_inicio = clock;
arq=[u,x];
[I]=imread(arq,'bmp');

IH=rgb2hsv(I);
[L,C,Q]=size(IH);
R=imresize(IH,[L,C]);
Z=double(R);
S=squeeze(Z);

fid1 = fopen('MediaDesvPadr HSV Vizinh9x9 HSV1 ResCalc.txt','w');
fid2 = fopen('MediaDesvPadr HSV Vizinh9x9 HSV2 ResCalcConf.txt','w');
fid3 = fopen('MediaDesvPadr HSV Vizinh9x9 HSV3 ResCalcConfResFinal.txt','w');

% -----
% Define a quantidade de iteracoes a ser realizada na imagem
QLF = L-8;
QCF = C-8;
CB = 0;
menormed = 9.9999;
maiormed = 0.0000;
menordesv = 9.9999;
maiordesv = 0.0000;

for LI = 1:QLF

    for CI = 1:QCF

        CB = CB+1;
        fprintf(fid1,'Info Cubo Numero --> %d \n',CB);
        fprintf(fid1,'Inic Cubo(LinxCol) --> %dX%d \n',LI,CI);

        fprintf(fid2,'Info Cubo Numero --> %d \n',CB);
        fprintf(fid2,'Inic Cubo(LinxCol) --> %dX%d \n',LI,CI);

        SomaELC = 0;
        MMatELC = 0;
        MedSomaELC = 0;
        MedMMatELC = 0;
        DifMediELC = 0;
        DesvPadELC = 0;

        ContPIX = 0;

        for LC = LI:LI+8

```

```
for CC = CI:CI+8

    fprintf(fid1,'%dx%d \t',LC,CC);

    for PX = 1:3

        SC=num2str(S(LC,CC,PX));
        fprintf(fid1,'%s',SC);

        if PX==1
            SCN = str2num(SC);
            SomaELC = (SomaELC + SCN);
            MMatELC = MMatELC + (SCN * SCN);
        end

        if PX<3
            fprintf(fid1,'\t');
        end

    end

    fprintf(fid1,'\n');
    ContPIX = ContPIX + 1;
end
end

fprintf(fid1,'\n\n');

MedSomaELC = (SomaELC / ContPIX);
fprintf(fid2,'Resultado da Media Cubo --> %7.4f \n',MedSomaELC);
MedMMatELC = (MMatELC / ContPIX);
DifMediELC = (MedMMatELC - MedSomaELC.^2);
DesvPadELC = (DifMediELC^0.5);
fprintf(fid2,'Resultado Desvio Padrao --> %7.4f \n\n',DesvPadELC);

fprintf(fid3,'%7.4f',MedSomaELC);
fprintf(fid3,'\t');
fprintf(fid3,'%7.4f',DesvPadELC);
fprintf(fid3,'\n');

if MedSomaELC < menormed
    menormed = MedSomaELC;
end

if MedSomaELC > maiormed
    maiormed = MedSomaELC;
end

if DesvPadELC < menordesv
    menordesv = DesvPadELC;
end

if DesvPadELC > maiordesv
    maiordesv = DesvPadELC;
end
```

```
    end
end

fclose(fid1);
fclose(fid2);
fclose(fid3);

subplot(2,1,1), imshow(I);
subplot(2,1,2), imshow(IH);

tempo_fim = clock;

fprintf('\n RESULTADO do PROCESSAMENTO...\n')
fprintf(' -----\n')
fprintf('\n Tempo de Processamento: Em segundos... %7.2f \n', etime(tempo_fim, tempo_inicio));
fprintf('\n Nro.Pixels contidos na Imagem Seleccionada: LIN x COL = QTDE    --> %d x %d = %d',L,C,
(L*C));
fprintf('\n Quantidade de Cubos (9x9) de Pixels contidos/varridos na Imagem --> %d\n\n',CB);
fprintf('\n *** Verifique arquivos "MEDIAREGIAO PIXEL HSVn....TXT" gerado no Processamento...↵
\n\n\n\n',L*C);
fprintf('\n *** Resultado Menor Media ... -> %7.4f',menormed);
fprintf('\n *** Resultado Maior Media ... -> %7.4f \n',maiormed);
fprintf('\n *** Resultado Menor Desvio... -> %7.4f',menordesv);
fprintf('\n *** Resultado Maior Desvio ... -> %7.4f \n\n\n',maiordesv);

msgbox('Concluido com Exito... Verifique as Mensagens do Processamento...', 'Aviso...');
```

```

% -----
% PROGRAMA 04 - CALCULO 7x7
% Encontra os valores das Medias em SubImagem tamanho 7x7
% -----

clc;
clear all;
close all;

[x,u]=uigetfile('*.bmp','Selecione a imagem...');
tempo_inicio = clock;
arq=[u,x];
[I]=imread(arq,'bmp');

IH=rgb2hsv(I);
[L,C,Q]=size(IH);
R=imresize(IH,[L,C]);
Z=double(R);
S=squeeze(Z);

fid1 = fopen('MediaDesvPadr HSV Vizinh7x7 HSV1 ResCalc.txt','w');
fid2 = fopen('MediaDesvPadr HSV Vizinh7x7 HSV2 ResCalcConf.txt','w');
fid3 = fopen('MediaDesvPadr HSV Vizinh7x7 HSV3 ResCalcConfResFinal.txt','w');

% -----
% Define a quantidade de iteracoes a ser realizada na imagem
QLF = L-6;
QCF = C-6;
CB = 0;
menormed = 9.9999;
maiormed = 0.0000;
menordesv = 9.9999;
maiordesv = 0.0000;

for LI = 1:QLF

    for CI = 1:QCF

        CB = CB+1;
        fprintf(fid1,'Info Cubo Numero --> %d \n',CB);
        fprintf(fid1,'Inic Cubo(LinxCol) --> %dX%d \n',LI,CI);

        fprintf(fid2,'Info Cubo Numero --> %d \n',CB);
        fprintf(fid2,'Inic Cubo(LinxCol) --> %dX%d \n',LI,CI);

        SomaELC = 0;
        MMatELC = 0;
        MedSomaELC = 0;
        MedMMatELC = 0;
        DifMediELC = 0;
        DesvPadELC = 0;

        ContPIX = 0;

        for LC = LI:LI+6

```

```
for CC = CI:CI+6

    fprintf(fid1,'%dx%d \t',LC,CC);

    for PX = 1:3

        SC=num2str(S(LC,CC,PX));
        fprintf(fid1,'%s',SC);

        if PX==1
            SCN = str2num(SC);
            SomaELC = (SomaELC + SCN);
            MMatELC = MMatELC + (SCN * SCN);
        end

        if PX<3
            fprintf(fid1,'\t');
        end

    end

    fprintf(fid1,'\n');
    ContPIX = ContPIX + 1;
end
end

fprintf(fid1,'\n\n');

MedSomaELC = (SomaELC / ContPIX);
fprintf(fid2,'Resultado da Media Cubo --> %7.4f \n',MedSomaELC);
MedMMatELC = (MMatELC / ContPIX);
DifMediELC = (MedMMatELC - MedSomaELC.^2);
DesvPadELC = (DifMediELC^0.5);
fprintf(fid2,'Resultado Desvio Padrao --> %7.4f \n\n',DesvPadELC);

fprintf(fid3,'%7.4f',MedSomaELC);
fprintf(fid3,'\t');
fprintf(fid3,'%7.4f',DesvPadELC);
fprintf(fid3,'\n');

if MedSomaELC < menormed
    menormed = MedSomaELC;
end

if MedSomaELC > maiormed
    maiormed = MedSomaELC;
end

if DesvPadELC < menordesv
    menordesv = DesvPadELC;
end

if DesvPadELC > maiordesv
    maiordesv = DesvPadELC;
end
```

```
end
end

fclose(fid1);
fclose(fid2);
fclose(fid3);

subplot(2,1,1), imshow(I);
subplot(2,1,2), imshow(IH);

tempo_fim = clock;

fprintf('\n RESULTADO do PROCESSAMENTO...\n')
fprintf(' -----\n')
fprintf('\n Tempo de Processamento: Em segundos... %7.2f \n', etime(tempo_fim, tempo_inicio));
fprintf('\n Nro.Pixels contidos na Imagem Seleccionada: LIN x COL = QTDE --> %d x %d = %d',L,C,
(L*C));
fprintf('\n Quantidade de Cubos (7x7) de Pixels contidos/varridos na Imagem --> %d\n\n',CB);
fprintf('\n *** Verifique arquivos "MEDIAREGIAO PIXEL HSVn....TXT" gerado no Processamento...↵
\n\n\n\n',L*C);
fprintf('\n *** Resultado Menor Media ... -> %7.4f',menormed);
fprintf('\n *** Resultado Maior Media ... -> %7.4f \n',maiormed);
fprintf('\n *** Resultado Menor Desvio... -> %7.4f',menordesv);
fprintf('\n *** Resultado Maior Desvio ... -> %7.4f \n\n\n',maiordesv);

msgbox('Concluido com Exito... Verifique as Mensagens do Processamento...', 'Aviso...');
```



```

% -----
% PROGRAMA 04 - CALCULO 5x5
% Encontra os valores das Medias em SubImagem tamanho 5x5
% -----

clc;
clear all;
close all;

[x,u]=uigetfile('*.bmp','Selecione a imagem...');
tempo_inicio = clock;
arq=[u,x];
[I]=imread(arq,'bmp');

IH=rgb2hsv(I);
[L,C,Q]=size(IH);
R=imresize(IH,[L,C]);
Z=double(R);
S=squeeze(Z);

fid1 = fopen('MediaDesvPadr HSV Vizinh5x5 HSV1 ResCalc.txt','w');
fid2 = fopen('MediaDesvPadr HSV Vizinh5x5 HSV2 ResCalcConf.txt','w');
fid3 = fopen('MediaDesvPadr HSV Vizinh5x5 HSV3 ResCalcConfResFinal.txt','w');

% -----
% Define a quantidade de iteracoes a ser realizada na imagem
QLF = L-4;
QCF = C-4;
CB = 0;
menormed = 9.9999;
maiormed = 0.0000;
menordesv = 9.9999;
maiordesv = 0.0000;

for LI = 1:QLF

    for CI = 1:QCF

        CB = CB+1;
        fprintf(fid1,'Info Cubo Numero --> %d \n',CB);
        fprintf(fid1,'Inic Cubo(LinxCol) --> %dX%d \n',LI,CI);

        fprintf(fid2,'Info Cubo Numero --> %d \n',CB);
        fprintf(fid2,'Inic Cubo(LinxCol) --> %dX%d \n',LI,CI);

        SomaELC = 0;
        MMatELC = 0;
        MedSomaELC = 0;
        MedMMatELC = 0;
        DifMediELC = 0;
        DesvPadELC = 0;

        ContPIX = 0;

        for LC = LI:LI+4

```

```

for CC = CI:CI+4

    fprintf(fid1,'%dx%d \t',LC,CC);

    for PX = 1:3

        SC=num2str(S(LC,CC,PX));
        fprintf(fid1,'%s',SC);

        if PX==1
            SCN = str2num(SC);
            SomaELC = (SomaELC + SCN);
            MMatELC = MMatELC + (SCN * SCN);
        end

        if PX<3
            fprintf(fid1,'\t');
        end

    end

    fprintf(fid1,'\n');
    ContPIX = ContPIX + 1;
end
end

fprintf(fid1,'\n\n');

MedSomaELC = (SomaELC / ContPIX);
fprintf(fid2,'Resultado da Media Cubo --> %7.4f \n',MedSomaELC);
MedMMatELC = (MMatELC / ContPIX);
DifMediELC = (MedMMatELC - MedSomaELC.^2);
DesvPadELC = (DifMediELC^0.5);
fprintf(fid2,'Resultado Desvio Padrao --> %7.4f \n\n',DesvPadELC);

fprintf(fid3,'%7.4f',MedSomaELC);
fprintf(fid3,'\t');
fprintf(fid3,'%7.4f',DesvPadELC);
fprintf(fid3,'\n');

if MedSomaELC < menormed
    menormed = MedSomaELC;
end

if MedSomaELC > maiormed
    maiormed = MedSomaELC;
end

if DesvPadELC < menordesv
    menordesv = DesvPadELC;
end

if DesvPadELC > maiordesv
    maiordesv = DesvPadELC;
end

```

```
    end
end

fclose(fid1);
fclose(fid2);
fclose(fid3);

subplot(2,1,1), imshow(I);
subplot(2,1,2), imshow(IH);

tempo_fim = clock;

fprintf('\n RESULTADO do PROCESSAMENTO...\n')
fprintf(' -----\n')
fprintf('\n Tempo de Processamento: Em segundos... %7.2f \n', etime(tempo_fim, tempo_inicio));
fprintf('\n Nro.Pixels contidos na Imagem Seleccionada: LIN x COL = QTDE    --> %d x %d = %d',L,C,
(L*C));
fprintf('\n Quantidade de Cubos (5x5) de Pixels contidos/varridos na Imagem --> %d\n\n',CB);
fprintf('\n *** Verifique arquivos "MEDIAREGIAO PIXEL HSVn....TXT" gerado no Processamento...↵
\n\n\n\n',L*C);
fprintf('\n *** Resultado Menor Media ... -> %7.4f',menormed);
fprintf('\n *** Resultado Maior Media ... -> %7.4f \n',maiormed);
fprintf('\n *** Resultado Menor Desvio... -> %7.4f',menordesv);
fprintf('\n *** Resultado Maior Desvio ... -> %7.4f \n\n\n',maiordesv);

msgbox('Concluido com Exito... Verifique as Mensagens do Processamento...', 'Aviso...');
```

```
% -----  
% PROGRAMA 04 - CALCULO 3X3  
% Encontra os valores das Medias em SubImagem tamanho 3x3  
% -----  
  
clc;  
clear all;  
close all;  
  
[x,u]=uigetfile('*.bmp','Selecione a imagem...');  
tempo_inicio = clock;  
arq=[u,x];  
[I]=imread(arq,'bmp');  
  
IH=rgb2hsv(I);  
[L,C,Q]=size(IH);  
R=imresize(IH,[L,C]);  
Z=double(R);  
S=squeeze(Z);  
  
fid1 = fopen('MediaDesvPadr HSV Vizinh3x3 HSV1 ResCalc.txt','w');  
fid2 = fopen('MediaDesvPadr HSV Vizinh3x3 HSV2 ResCalcConf.txt','w');  
fid3 = fopen('MediaDesvPadr HSV Vizinh3x3 HSV3 ResCalcConfResFinal.txt','w');  
  
% -----  
% Define a quantidade de iteracoes a ser realizada na imagem  
QLF = L-2;  
QCF = C-2;  
CB = 0;  
menormed = 9.9999;  
maiormed = 0.0000;  
menordesv = 9.9999;  
maiordesv = 0.0000;  
  
for LI = 1:QLF  
  
    for CI = 1:QCF  
  
        CB = CB+1;  
        fprintf(fid1,'Info Cubo Numero --> %d \n',CB);  
        fprintf(fid1,'Inic Cubo(LinxCol) --> %dX%d \n',LI,CI);  
  
        fprintf(fid2,'Info Cubo Numero --> %d \n',CB);  
        fprintf(fid2,'Inic Cubo(LinxCol) --> %dX%d \n',LI,CI);  
  
        SomaELC = 0;  
        MMatELC = 0;  
        MedSomaELC = 0;  
        MedMMatELC = 0;  
        DifMediELC = 0;  
        DesvPadELC = 0;  
  
        ContPIX = 0;  
  
        for LC = LI:LI+2
```

```

for CC = CI:CI+2

    fprintf(fid1,'%dx%d \t',LC,CC);

    for PX = 1:3

        SC=num2str(S(LC,CC,PX));
        fprintf(fid1,'%s',SC);

        if PX==1
            SCN = str2num(SC);
            SomaELC = (SomaELC + SCN);
            MMatELC = MMatELC + (SCN * SCN);
        end

        if PX<3
            fprintf(fid1,'\t');
        end

    end

    fprintf(fid1,'\n');
    ContPIX = ContPIX + 1;
end
end

fprintf(fid1,'\n\n');

MedSomaELC = (SomaELC / ContPIX);
fprintf(fid2,'Resultado da Media Cubo --> %7.4f \n',MedSomaELC);
MedMMatELC = (MMatELC / ContPIX);
DifMediELC = (MedMMatELC - MedSomaELC.^2);
DesvPadELC = (DifMediELC^0.5);
fprintf(fid2,'Resultado Desvio Padrao --> %7.4f \n\n',DesvPadELC);

fprintf(fid3,'%7.4f',MedSomaELC);
fprintf(fid3,'\t');
fprintf(fid3,'%7.4f',DesvPadELC);
fprintf(fid3,'\n');

if MedSomaELC < menormed
    menormed = MedSomaELC;
end

if MedSomaELC > maiormed
    maiormed = MedSomaELC;
end

if DesvPadELC < menordesv
    menordesv = DesvPadELC;
end

if DesvPadELC > maiordesv
    maiordesv = DesvPadELC;
end

```

```
    end
end

fclose(fid1);
fclose(fid2);
fclose(fid3);

subplot(2,1,1), imshow(I);
subplot(2,1,2), imshow(IH);

tempo_fim = clock;

fprintf('\n RESULTADO do PROCESSAMENTO...\n')
fprintf(' -----\n')
fprintf('\n Tempo de Processamento: Em segundos... %7.2f \n', etime(tempo_fim, tempo_inicio));
fprintf('\n Nro.Pixels contidos na Imagem Seleccionada: LIN x COL = QTDE    --> %d x %d = %d',L,C,
(L*C));
fprintf('\n Quantidade de Cubos (3x3) de Pixels contidos/varridos na Imagem --> %d\n\n',CB);
fprintf('\n *** Verifique arquivos "MEDIAREGIAO PIXEL HSVn....TXT" gerado no Processamento...↵
\n\n\n\n',L*C);
fprintf('\n *** Resultado Menor Media ... -> %7.4f',menormed);
fprintf('\n *** Resultado Maior Media ... -> %7.4f \n',maiormed);
fprintf('\n *** Resultado Menor Desvio... -> %7.4f',menordesv);
fprintf('\n *** Resultado Maior Desvio ... -> %7.4f \n\n\n',maiordesv);

msgbox('Concluido com Exito... Verifique as Mensagens do Processamento...', 'Aviso...');
```

```

% -----
% PROGRAMA 03
% Procura Pixel errado na Colcha de Retalhos para que possam ser
% alterados conforme especificações de J.Kovac
% -----

clc;
clear all;
close all;

[x,u]=uigetfile('*.bmp','Selecione a imagem a ser varrida...');

arq=[u,x];
[I]=imread(arq,'bmp');

[L,C,Q]=size(I);
R = 0;
G = 0;
B = 0;

pixel = 0;

LI = 0;
CI = 0;

for LI = 1:L
    for CI = 1:C

        INovoR(LI,CI,1) = I(LI,CI,1);
        INovoR(LI,CI,2) = I(LI,CI,2);
        INovoR(LI,CI,3) = I(LI,CI,3);

        if ((I(LI,CI,1) == 255) & (I(LI,CI,2) == 241) & (I(LI,CI,3) == 190 )) ...
| ((I(LI,CI,1) == 255) & (I(LI,CI,2) == 241) & (I(LI,CI,3) == 192 )) ...
| ((I(LI,CI,1) == 255) & (I(LI,CI,2) == 241) & (I(LI,CI,3) == 194 )) ...
| ((I(LI,CI,1) == 255) & (I(LI,CI,2) == 241) & (I(LI,CI,3) == 195 )) ...
| ((I(LI,CI,1) == 255) & (I(LI,CI,2) == 241) & (I(LI,CI,3) == 196 )) ...
| ((I(LI,CI,1) == 255) & (I(LI,CI,2) == 241) & (I(LI,CI,3) == 199 )) ...
| ((I(LI,CI,1) == 255) & (I(LI,CI,2) == 241) & (I(LI,CI,3) == 200 )) ...
| ((I(LI,CI,1) == 255) & (I(LI,CI,2) == 241) & (I(LI,CI,3) == 201 )) ...
| ((I(LI,CI,1) == 255) & (I(LI,CI,2) == 241) & (I(LI,CI,3) == 203 )) ...
| ((I(LI,CI,1) == 255) & (I(LI,CI,2) == 241) & (I(LI,CI,3) == 205 )) ...
| ((I(LI,CI,1) == 255) & (I(LI,CI,2) == 241) & (I(LI,CI,3) == 209 )) ...
| ((I(LI,CI,1) == 255) & (I(LI,CI,2) == 242) & (I(LI,CI,3) == 190 )) ...
| ((I(LI,CI,1) == 255) & (I(LI,CI,2) == 242) & (I(LI,CI,3) == 193 )) ...
| ((I(LI,CI,1) == 255) & (I(LI,CI,2) == 242) & (I(LI,CI,3) == 195 )) ...
| ((I(LI,CI,1) == 255) & (I(LI,CI,2) == 242) & (I(LI,CI,3) == 197 )) ...
| ((I(LI,CI,1) == 255) & (I(LI,CI,2) == 242) & (I(LI,CI,3) == 198 )) ...
| ((I(LI,CI,1) == 255) & (I(LI,CI,2) == 242) & (I(LI,CI,3) == 200 )) ...
| ((I(LI,CI,1) == 255) & (I(LI,CI,2) == 242) & (I(LI,CI,3) == 201 )) ...
| ((I(LI,CI,1) == 255) & (I(LI,CI,2) == 242) & (I(LI,CI,3) == 206 )) ...
| ((I(LI,CI,1) == 255) & (I(LI,CI,2) == 242) & (I(LI,CI,3) == 207 )) ...
| ((I(LI,CI,1) == 255) & (I(LI,CI,2) == 243) & (I(LI,CI,3) == 194 )) ...
| ((I(LI,CI,1) == 255) & (I(LI,CI,2) == 243) & (I(LI,CI,3) == 198 )) ...
| ((I(LI,CI,1) == 255) & (I(LI,CI,2) == 243) & (I(LI,CI,3) == 199 )) ...
| ((I(LI,CI,1) == 255) & (I(LI,CI,2) == 243) & (I(LI,CI,3) == 200 )) ...

```

```

| ((I(LI,CI,1) == 255) & (I(LI,CI,2) == 243) & (I(LI,CI,3) == 201 )) ...
| ((I(LI,CI,1) == 255) & (I(LI,CI,2) == 243) & (I(LI,CI,3) == 202 )) ...
| ((I(LI,CI,1) == 255) & (I(LI,CI,2) == 243) & (I(LI,CI,3) == 203 )) ...
| ((I(LI,CI,1) == 255) & (I(LI,CI,2) == 243) & (I(LI,CI,3) == 205 )) ...
| ((I(LI,CI,1) == 255) & (I(LI,CI,2) == 243) & (I(LI,CI,3) == 206 )) ...
| ((I(LI,CI,1) == 255) & (I(LI,CI,2) == 243) & (I(LI,CI,3) == 207 )) ...
| ((I(LI,CI,1) == 255) & (I(LI,CI,2) == 244) & (I(LI,CI,3) == 200 )) ...
| ((I(LI,CI,1) == 255) & (I(LI,CI,2) == 244) & (I(LI,CI,3) == 201 )) ...
| ((I(LI,CI,1) == 255) & (I(LI,CI,2) == 244) & (I(LI,CI,3) == 202 )) ...
| ((I(LI,CI,1) == 255) & (I(LI,CI,2) == 244) & (I(LI,CI,3) == 203 )) ...
| ((I(LI,CI,1) == 255) & (I(LI,CI,2) == 244) & (I(LI,CI,3) == 205 )) ...
| ((I(LI,CI,1) == 255) & (I(LI,CI,2) == 244) & (I(LI,CI,3) == 206 )) ...
| ((I(LI,CI,1) == 255) & (I(LI,CI,2) == 246) & (I(LI,CI,3) == 198 )) ...
| ((I(LI,CI,1) == 255) & (I(LI,CI,2) == 246) & (I(LI,CI,3) == 203 )) ...
| ((I(LI,CI,1) == 255) & (I(LI,CI,2) == 246) & (I(LI,CI,3) == 206 )) ...
| ((I(LI,CI,1) == 255) & (I(LI,CI,2) == 247) & (I(LI,CI,3) == 203 ))

```

```

    INovoR(LI,CI,1) = 255;
    INovoR(LI,CI,2) = 0;
    INovoR(LI,CI,3) = 0;
    pixel = pixel + 1;
end

```

```

end
end

```

```

% -----
imwrite(INovoR,'Imagem0 - PeleGeral-1000x900 REVISADO.bmp');

fprintf('\n RESULTADO do PROCESSAMENTO...\n')
fprintf(' -----\n')
fprintf('\n Quantidade de Pixel Alterado... ---> %d ', pixel);

subplot(1,2,1), imshow(I);title('Imagem Original RGB - 900 x 1000');
subplot(1,2,2), imshow(INovoR);title('Pixels com Erros de RGB');

msgbox('Verifique as Imagens e o Resultado do Processamento...','Concluido...');

```



```
% -----  
% PROGRAMA 02  
% Redimensiona Imagem da Colcha de Retalhos para Calculo das Médias  
% -----  
  
clc;  
clear all;  
close all;  
  
[x,u]=uigetfile('*.bmp','Selecione a imagem...');  
arq=[u,x];  
[I]=imread(arq,'bmp');  
[L,C,Q]=size(I);  
  
% -----  
% Dimensiona a imagem para um tamanho especifico  
%CD = C / 2;  
%LD = L / 2;  
CD = 160;  
LD = 120;  
  
R=imresize(I,[LD,CD]);  
  
cabe2 = sprintf('Imagem %dx%d.bmp',CD,LD);  
imshow(R);title(cabe2);  
  
imwrite(R,'Imagem tamanho 160x120.bmp');
```

```
% -----  
% PROGRAMA 01  
% Identifica Pixel PELE/NÃO PELE conforme Média definida  
% -----  
  
clc;  
clear all;  
close all;  
  
[x,u]=uigetfile('*.bmp','Selecione a imagem a ser varrida...');  
tempo_inicio = clock;  
arq=[u,x];  
[I]=imread(arq,'bmp');  
  
H=rgb2hsv(I);  
[L,C,Q]=size(H);  
R=imresize(H,[L,C]);  
Z=double(R);  
S=squeeze(Z);  
WSC = S;  
  
INovoH = zeros(L,C,Q);  
INovoR = zeros(L,C,Q);  
  
QLF = L-2;  
QCF = C-2;  
CB = 0;  
PLC = 0;  
PCC = 0;  
WSC = 0;  
SCN = 0;  
WLC = 0;  
WCC = 0;  
  
figure(1);  
imshow(I);  
  
for LI = 1:QLF  
    for CI = 1:QCF  
  
        % -----  
        % %Conjunto de Variaveis utilizadas para o calculo da Media  
        ContPIX = 0;  
        SomaELC = 0;  
        MMatELC = 0;  
        MedSomaELC = 0;  
        MedMMatELC = 0;  
        DifMediELC = 0;  
        DesvPadELC = 0;  
  
        for LC = LI:LI+2  
            PLC = PLC + 1;  
  
            for CC = CI:CI+2  
                PCC = PCC + 1;
```

```

% -----
% Define os valores do pixel no formato especificado
for PX = 1:3

    if PX==1
        SCN = (S(LC,CC,PX));
        SomaELC = (SomaELC + SCN);
        MMatELC = MMatELC + (SCN * SCN);
    end

end

if (PLC == 2 & PCC == 2)
    WLC = LC;
    WCC = CC;
    WSC = S(LC,CC,:);
    INovoH(WLC,WCC,:) = WSC;
end

ContPIX = ContPIX + 1;

end
PCC = 0;
end
PLC = 0;

MedSomaELC = (SomaELC / ContPIX);
MedMMatELC = (MMatELC / ContPIX);
DifMediELC = (MedMMatELC - MedSomaELC.^2);
DesvPadELC = (DifMediELC^0.5);

% -----
% Procedimento para PINTAR/IDENTIFICAR na nova Imagem o Ponto(Pixel) que e Pele
if (MedSomaELC >= 0.0088 & MedSomaELC <= 0.1360);
    INovoH(WLC,WCC,1) = 0.3333;
    INovoH(WLC,WCC,2) = 1;
    INovoH(WLC,WCC,3) = 1;
end

end

end

% -----
% Exibe a Figura com a cor do pixel alterada...
INovoR=hsv2rgb(INovoH);

figure(2);
imshow(INovoR);

msgbox('Verifique as Imagens do Processamento...','Concluido...');

```

```
% -----  
% PROGRAMA 06 - VARRE IMAGEM baseado em J.KOVAC - Formula B  
% Segmenta Pixels PELE/NAO PELE conforme Media encontrada  
% -----
```

```
clc;  
clear all;  
close all;
```

```
[x,u]=uigetfile('*.bmp','Selecione a imagem a ser varrida...');  
arq=[u,x];  
[I]=imread(arq,'bmp');  
[L,C,Q]=size(I);
```

```
R=imresize(I,[L,C]);  
Z=double(R);  
S=squeeze(Z);  
WSC = S;
```

```
IAtual = I;  
INovo = I;
```

```
for LI = 1:L  
    for CI = 1:C
```

```
        R = I(LI,CI,1);  
        G = I(LI,CI,2);  
        B = I(LI,CI,3);
```

```
        if ((R>220) & (G>210) & (B>170))
```

```
            DiferenRG = R - G;  
            if (DiferenRG <= 15) & (R>B) & (G>B)
```

```
                MaiorV = 0;  
                MenorV = 0;  
                MaiorV = max(I(LI,CI,:));  
                MenorV = min(I(LI,CI,:));  
                DiferenMaiMen = MaiorV - MenorV;
```

```
                INovo(LI,CI,1) = 255;  
                INovo(LI,CI,2) = 0;  
                INovo(LI,CI,3) = 0;
```

```
            end  
        end  
    end  
end
```

```
arqgerado = ['2Resultado_Kovac_',x];  
imshow(INovo);title(arqgerado);  
imwrite(INovo,arqgerado);
```

```
% -----  
% PROGRAMA 06 - VARRE IMAGEM baseado em J.KOVAC - Formula A  
% Segmenta Pixels PELE/NAO PELE conforme Media encontrada  
% -----
```

```
clc;  
clear all;  
close all;
```

```
[x,u]=uigetfile('*.bmp','Selecione a imagem a ser varrida...');  
arq=[u,x];  
[I]=imread(arq,'bmp');  
[L,C,Q]=size(I);
```

```
R=imresize(I,[L,C]);  
Z=double(R);  
S=squeeze(Z);  
WSC = S;
```

```
IAtual = I;  
INovo = I;
```

```
for LI = 1:L  
    for CI = 1:C
```

```
        R = I(LI,CI,1);  
        G = I(LI,CI,2);  
        B = I(LI,CI,3);
```

```
        if ((R>95) & (G>40) & (B>20))
```

```
            DiferenRG = R - G;  
            if (DiferenRG > 15) & (R>G) & (R>B)
```

```
                MaiorV = 0;  
                MenorV = 0;  
                MaiorV = max(I(LI,CI,:));  
                MenorV = min(I(LI,CI,:));  
                DiferenMaiMen = MaiorV - MenorV;
```

```
                if (DiferenMaiMen > 15)
```

```
                    INovo(LI,CI,1) = 255;  
                    INovo(LI,CI,2) = 0;  
                    INovo(LI,CI,3) = 0;
```

```
                end
```

```
            end
```

```
        end
```

```
    end
```

```
end
```

```
arqgerado = ['Resultado_Kovac_',x];  
imshow(INovo);title(arqgerado);  
imwrite(INovo,arqgerado);
```

```
% -----  
% PROGRAMA 05 - VARRE IMAGEM 9x9 - 1000x900  
% Segmenta Pixels PELE/NAO PELE conforme Media encontrada  
% -----  
  
clc;  
clear all;  
close all;  
  
[x,u]=uigetfile('*.bmp','Selecione a imagem a ser varrida...');  
tempo_inicio = clock;  
arq=[u,x];  
[I]=imread(arq,'bmp');  
  
IH=rgb2hsv(I);  
[L,C,Q]=size(IH);  
R=imresize(IH,[L,C]);  
Z=double(R);  
S=squeeze(Z);  
WSC = S;  
  
INovoH = zeros(L,C,Q);  
INovoR = zeros(L,C,Q);  
  
QLF = L-8;  
QCF = C-8;  
CB = 0;  
PLC = 0;  
PCC = 0;  
WSC = 0;  
SCN = 0;  
WLC = 0;  
WCC = 0;  
  
for LI = 1:QLF  
    for CI = 1:QCF  
  
        ContPIX = 0;  
        SomaELC = 0;  
        MMatELC = 0;  
        MedSomaELC = 0;  
        MedMMatELC = 0;  
        DifMediELC = 0;  
        DesvPadELC = 0;  
  
        for LC = LI:LI+8  
            PLC = PLC + 1;  
  
            for CC = CI:CI+8  
                PCC = PCC + 1;  
  
                for PX = 1:3  
  
                    if PX==1  
                        SCN = (S(LC,CC,PX));  
                        SomaELC = (SomaELC + SCN);
```

```
        MMatELC = MMatELC + (SCN * SCN);
    end

end

if (PLC == 5 & PCC == 5)
    WLC = LC;
    WCC = CC;
    WSC = S(LC,CC,:);
    INovoH(WLC,WCC,:) = WSC;
end

ContPIX = ContPIX + 1;

end
PCC = 0;
end
PLC = 0;

MedSomaELC = (SomaELC / ContPIX);
MedMMatELC = (MMatELC / ContPIX);
DifMediELC = (MedMMatELC - MedSomaELC.^2);
DesvPadELC = (DifMediELC^0.5);

if (MedSomaELC >= 0.0198 & MedSomaELC <= 0.1127)
    INovoH(WLC,WCC,1) = 0.0000;
    INovoH(WLC,WCC,2) = 1;
    INovoH(WLC,WCC,3) = 1;
end

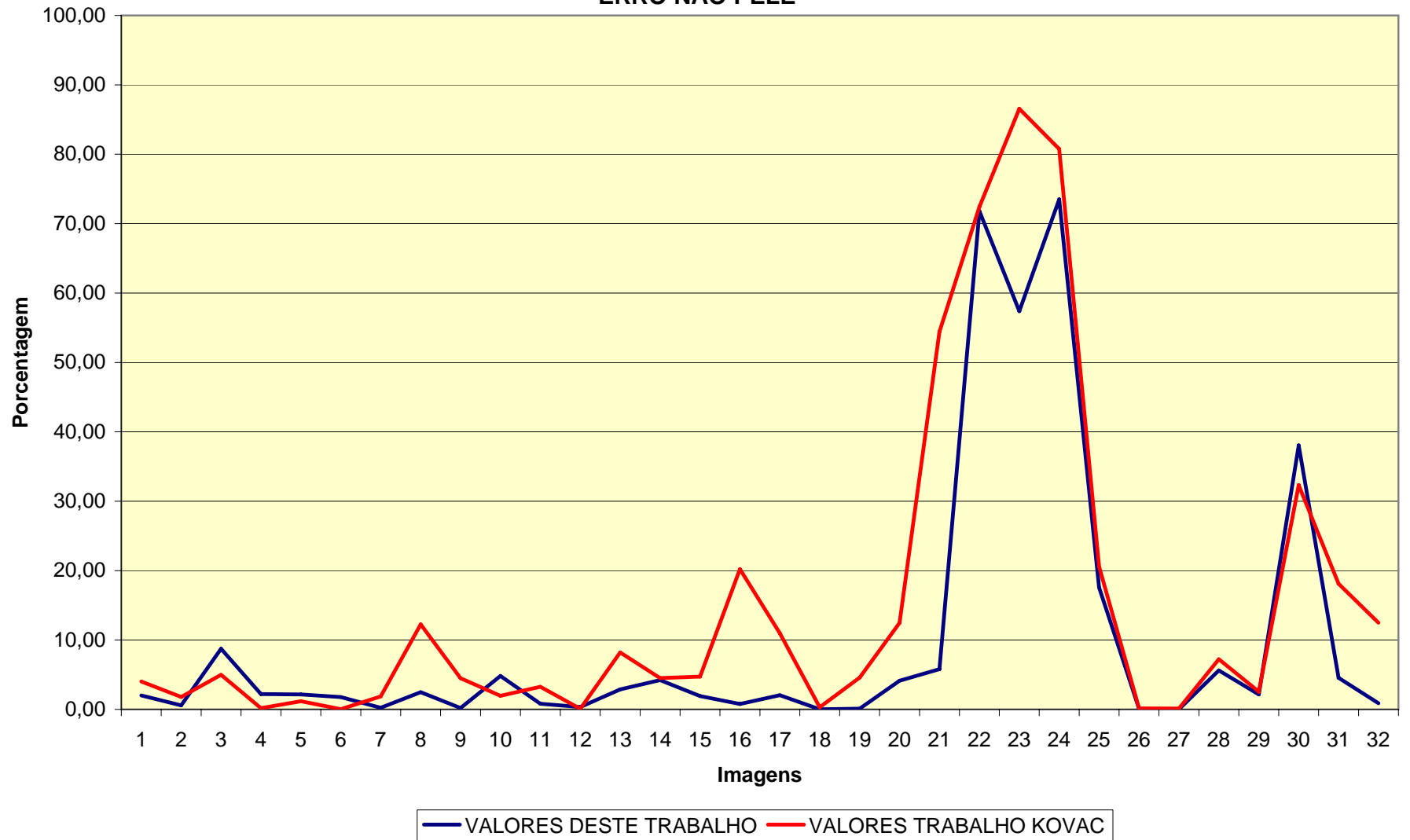
end
end

INovoR=hsv2rgb(INovoH);
cabe2 = sprintf('ImagemXX - Vizinhanca 9x9.bmp' );
imshow(INovoR);title(cabe2);
imwrite(INovoR,'ImagemXX - Vizinhanca 9x9.bmp');
```

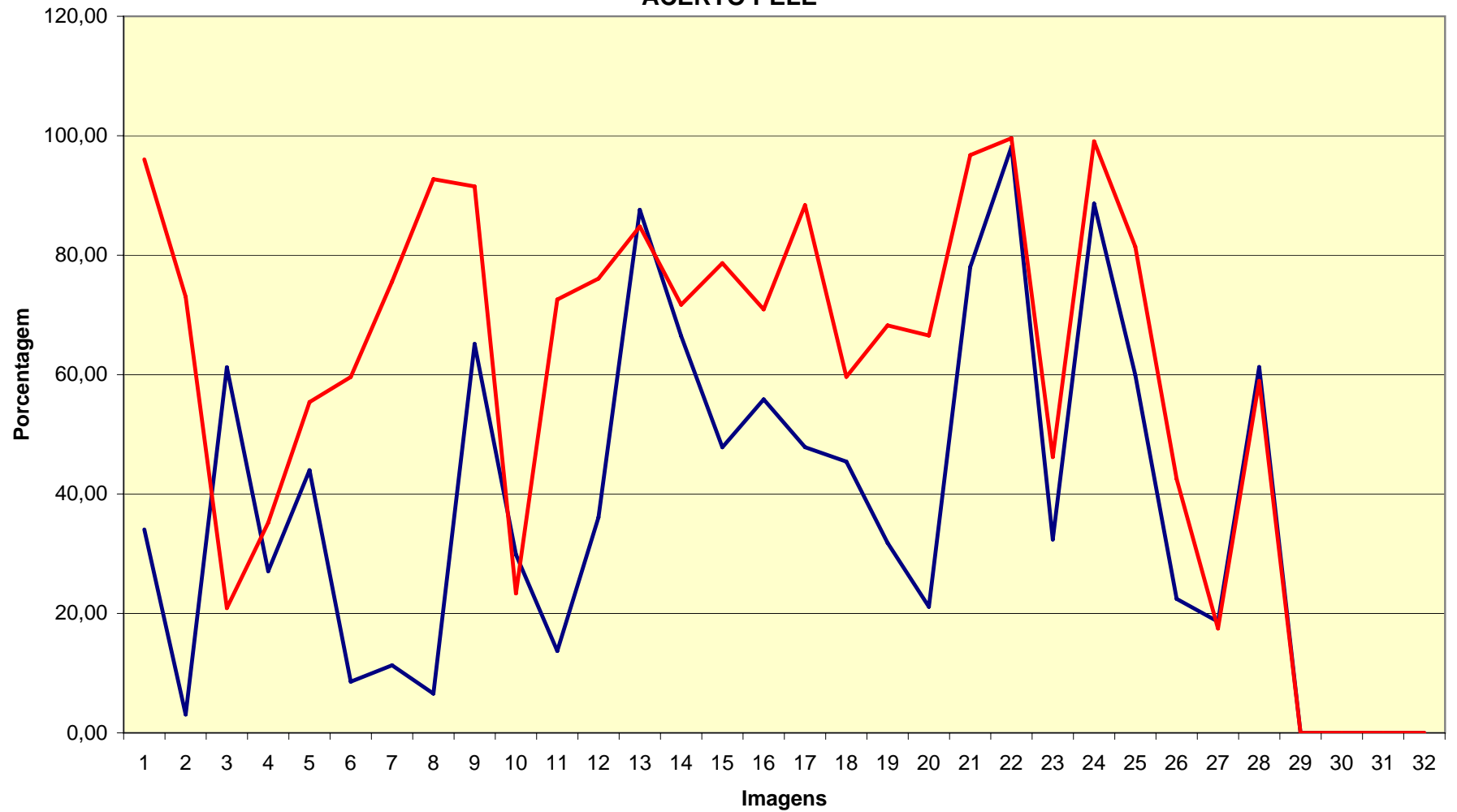
# Anexo C



### ERRO NÃO PELE

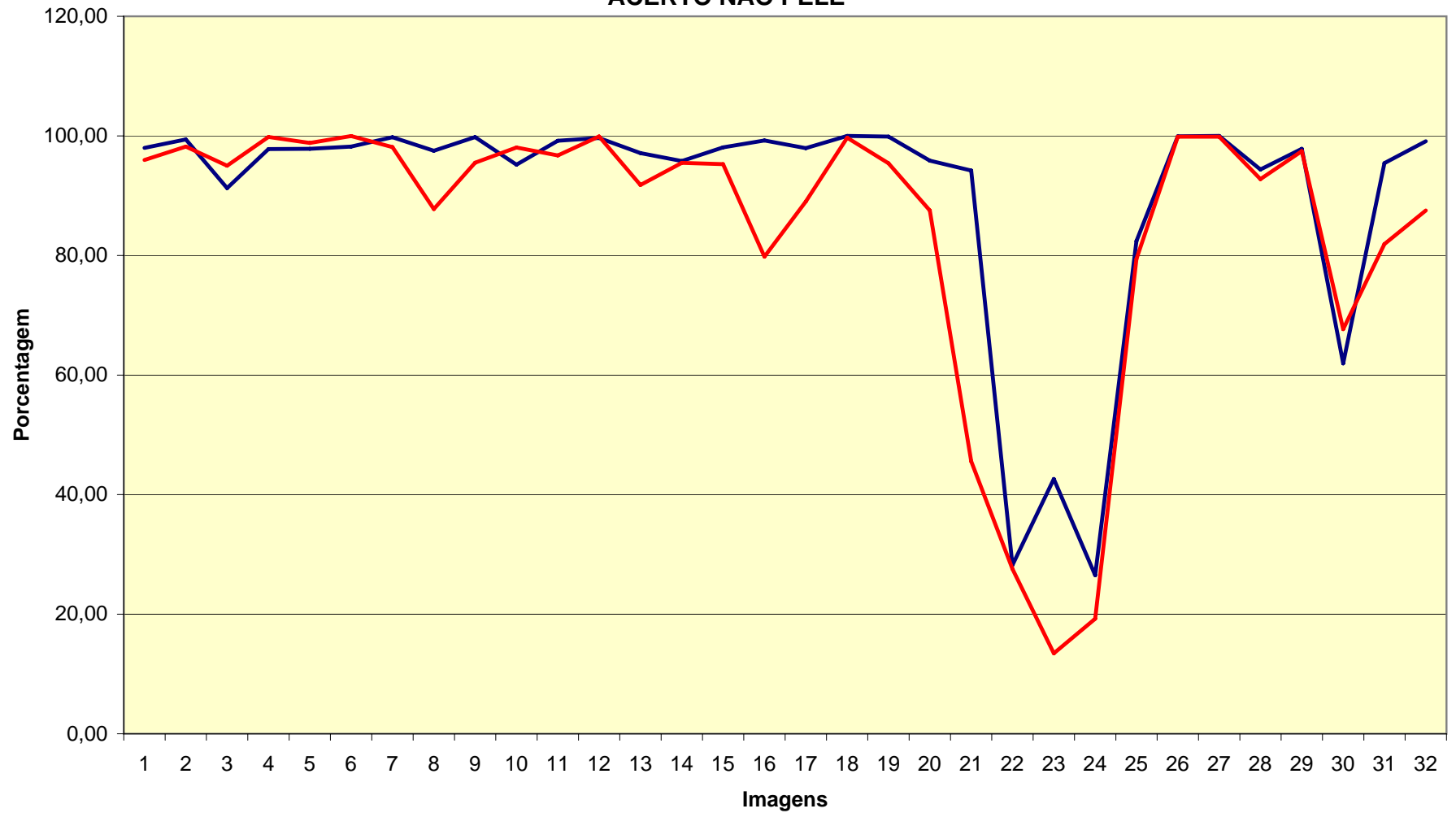


### ACERTO PELE



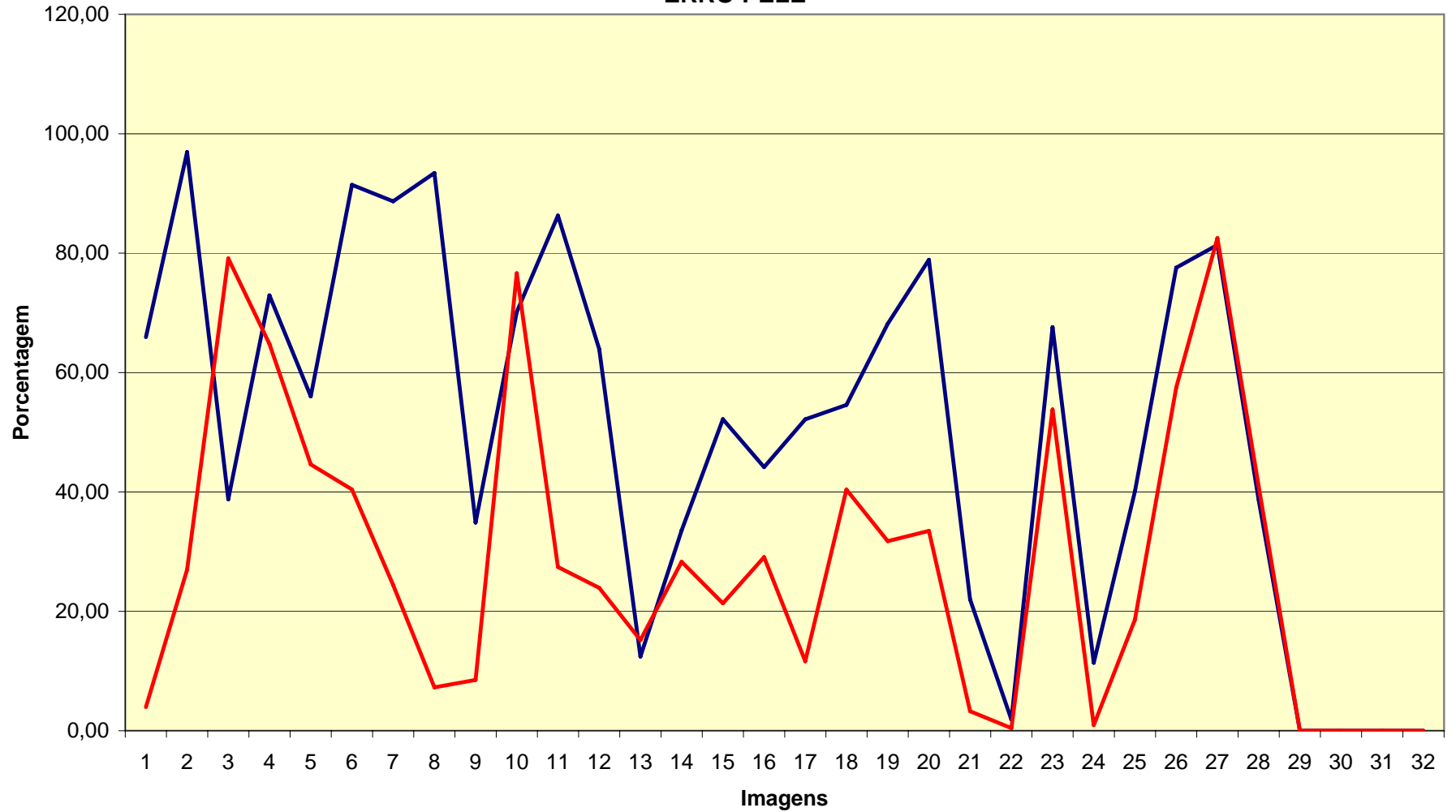
— VALORES DESTE TRABALHO — VALORES TRABALHO KOVAC

### ACERTO NÃO PELE



— VALORES DESTE TRABALHO — VALORES TRABALHO KOVAC

### ERRO PELE



— VALORES DESTE TRABALHO — VALORES TRABALHO KOVAC

# 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](#)