



**Universidade Católica Dom Bosco**  
Centro de Ciências Exatas e Tecnológicas  
Curso de Engenharia de Computação

**Estudo Comparativo de Técnicas de Visão  
Computacional para Detecção de Pele Humana**

Roger Sedlacek Fibiger

Prof(a). Orientador(a): Dr. Hemerson Pistori

*Relatório Parcial submetido como um dos requisitos para a obtenção do grau de Engenheiro de Computação.*

UCDB - Campo Grande - MS - Dezembro/2004

# Agradecimentos

Agradeço aos meus pais, Ruy e Leda, por todas as oportunidades que me ofereceram e continuam oferecendo, bem como da paciência, compreensão, carinho, companheirismo e estímulo durante todos estes anos de curso. Sem vocês eu nunca chegaria até aqui.

Ao meu irmão e minha namorada, Rudy e Jany, pelo carinho, companheirismo e também paciência e compreensão.

Ao meu professor, orientador e amigo Hemerson Pistori pela paciência, excelente orientação e pelas conversas a respeito de Software Livre e seus impactos na sociedade, que certamente me influenciarão muito durante a minha vida pós-acadêmica.

A todos os 7 amigos sobreviventes do curso na sala inicial. Conseguimos, galera!

A todos os professores que se tornaram companheiros após longos semestres de convivência: Marco, Amaury, Conceição, Leonardo, Ricardo, Maurício, Milton, Eugene, Gonda, Mauro, Alfredo, Elvézio e Jenner.

A todas as amizades adquiridas durante estes cinco anos de computação que provavelmente continuarão fortes durante um bom tempo de minha vida (graças ao messenger e ao ICQ).

À West Central pelo rock n' roll de boa qualidade e alto astral.

Ao Carlão do Pé-Sujo pelos fiados e pela comida sempre boa e barata.

A todos os meus amigos externos à Universidade que certamente me ajudaram a conduzir o barco com todas as pizzas e cervejas divididas ao longo de minha vida acadêmica.

# Resumo

A cor da pele provou ser uma característica útil e robusta para detecção de mãos e faces, localização e rastreamento. Diversas técnicas para modelagem e reconhecimento da cor da pele foram propostas durante o passar dos anos. Alguns artigos comparando diferentes pontos de vista foram publicados e serão exemplificados neste trabalho. Este trabalho apresenta um estudo comparativo a respeito de algumas técnicas de modelagem e reconhecimento da cor da pele, bem como de alguns aplicativos e seus respectivos módulos existentes no mercado, com o objetivo de facilitar a busca por essas técnicas.

# Abstract

Skin color has proven to be a useful and robust characteristic for face detection, localization and tracking. Numerous techniques for skin color modelling and recognition have been proposed during several past years. Some papers comparing different approaches have been published and it'll be exemplified in this paper. This work presents a comparative study of some of this techniques, as well as the softwares and its respectives plug-ins, with the objective of helping the search for these techniques.

# Conteúdo

<b>1</b>	<b>Introdução</b>	<b>10</b>
<b>2</b>	<b>Revisão Bibliográfica</b>	<b>14</b>
2.1	Trabalhos Correlatos . . . . .	14
2.2	Conceitos Auxiliares . . . . .	16
2.2.1	Teorema de Bayes . . . . .	16
2.2.2	Histogramas . . . . .	17
<b>3</b>	<b>Plataforma SIGUS</b>	<b>18</b>
3.1	Introdução . . . . .	18
3.2	Objetivos e Contexto deste Projeto . . . . .	19
<b>4</b>	<b>Espaços de Cores</b>	<b>21</b>
4.1	Cor . . . . .	21
4.1.1	Cores-luz e cores-pigmento . . . . .	21
4.1.2	Classificação da cor . . . . .	23
4.1.3	Crominância e Luminância . . . . .	24
4.2	Espaços de Cores . . . . .	25
4.2.1	RGB . . . . .	25
4.2.2	RGB Normalizado . . . . .	25
4.2.3	HSV, HSL, HSI - Matiz, Saturação, Valor . . . . .	26
4.2.4	TSL - Matiz, Saturação, Luminância . . . . .	30
4.2.5	YCbCr . . . . .	32
4.2.6	Outros Espaços de Cores . . . . .	33
4.2.7	Espaços de Cores Independentes de Dispositivos . . . . .	35
4.2.8	Comparações entre espaços de cores . . . . .	35
<b>5</b>	<b>Modelos Estatísticos e Algoritmos</b>	<b>37</b>
5.1	Técnica baseada em Tabelas de Conversão ( <i>Look-Up Table</i> ) . . . . .	37
5.2	Método de Bayes . . . . .	39
5.3	Modelos Gaussianos . . . . .	40

---

5.3.1	Distribuição Normal . . . . .	40
5.4	Desempenho de alguns métodos . . . . .	42
<b>6</b>	<b>Ferramentas de Apoio e Experimentos</b>	<b>43</b>
6.1	ImageJ . . . . .	43
6.1.1	Descrição . . . . .	43
6.1.2	Plugins . . . . .	44
6.2	Colorspace . . . . .	58
6.3	RGBCube . . . . .	61
<b>7</b>	<b>Considerações Finais</b>	<b>62</b>
7.1	Conclusão . . . . .	62
7.2	Trabalhos Futuros . . . . .	62
	<b>Referências Bibliográficas</b>	<b>64</b>

# Lista de Figuras

4.1	Decomposição e recomposição da luz . . . . .	22
4.2	Decomposição e recomposição da luz . . . . .	22
4.3	Decomposição e recomposição da luz . . . . .	22
4.4	Soma das três cores primárias . . . . .	22
4.5	Soma das três cores secundárias . . . . .	23
4.6	Pirâmide de Lambert . . . . .	24
4.7	Dois exemplos de espectro de cores . . . . .	24
4.8	Cubo RGB . . . . .	25
4.9	Espaço de cores RGB, gerado no RGBCube . . . . .	26
4.10	Cone HSV . . . . .	28
4.11	Espaço de cores HSI, gerado no RGBCube . . . . .	29
4.12	Espaço de cores HSI, gerado no RGBCube . . . . .	29
4.13	Espaço de cores HSI, gerado no RGBCube . . . . .	30
4.14	Espaço de cores HSI, gerado no RGBCube . . . . .	31
4.15	Espaço de cores HSV, gerado no RGBCube . . . . .	31
4.16	Espaço de cores HSV, gerado no RGBCube . . . . .	32
4.17	Espaço de cores YCbCr, gerado no RGBCube . . . . .	33
4.18	Espaço de cores YCbCr, gerado no RGBCube . . . . .	34
4.19	Espaço de cores YCbCr, gerado no RGBCube . . . . .	34
6.1	<i>Hands.Easy.Back</i> originais de entrada . . . . .	45
6.2	<i>Hands.Easy.Back</i> com $C_b$ e $C_r$ iguais a $A$ . . . . .	46
6.3	<i>Hands.Easy.Back</i> com $C_b$ e $C_r$ iguais a $B$ . . . . .	46
6.4	<i>Hands.Easy.Back</i> com $C_b$ e $C_r$ iguais a $C$ . . . . .	47
6.5	<i>Hands.Complex.Back</i> de entrada . . . . .	47
6.6	<i>Hands.Complex.Back</i> testados com os valores de $A$ . . . . .	47
6.7	<i>Hands.Complex.Back</i> testados com os valores de $B$ . . . . .	48
6.8	<i>Hands.Complex.Back</i> testados com os valores de $C$ . . . . .	48
6.9	Amostra Pele I . . . . .	50
6.10	Pele Histograma I . . . . .	50
6.11	Amostra N-Pele I . . . . .	50

---

6.12	N-Pele Histograma I . . . . .	50
6.13	Amostra Pele II . . . . .	50
6.14	Pele Histograma II . . . . .	50
6.15	Amostra N-Pele II . . . . .	51
6.16	N-Pele Histograma II . . . . .	51
6.17	Amostra Pele III . . . . .	51
6.18	Pele Histograma III . . . . .	51
6.19	Amostra N-Pele III . . . . .	52
6.20	N-Pele Histograma III . . . . .	52
6.21	Amostra Pele IV . . . . .	52
6.22	Pele Histograma IV . . . . .	52
6.23	Amostra N-Pele IV . . . . .	52
6.24	N-Pele Histograma IV . . . . .	52
6.25	Amostra Pele V . . . . .	53
6.26	Pele Histograma V . . . . .	53
6.27	Amostra N-Pele V . . . . .	53
6.28	N-Pele Histograma V . . . . .	53
6.29	Amostra I no 3D Color Inspector . . . . .	55
6.30	Amostra II no 3D Color Inspector . . . . .	55
6.31	Amostra III no 3D Color Inspector . . . . .	56
6.32	RGB . . . . .	56
6.33	YCbCr . . . . .	57
6.34	HSV . . . . .	57
6.35	Pele . . . . .	58
6.36	Imagem com o espaço de cores RGB . . . . .	59
6.37	Imagem com o espaço de cores HSI . . . . .	60
6.38	Imagem com o espaço de cores HSV . . . . .	60
6.39	Imagem com o espaço de cores YCbCr . . . . .	60

# Lista de Tabelas

4.1	Transformações <i>RGB/HSL/HSV</i> . . . . .	28
5.1	Tabela <i>r/g</i> . . . . .	38
5.2	Tabela <i>r/g</i> . . . . .	38
5.3	Resultados para os espaços de cores crominantes . . . . .	41
5.4	Desempenho de detectores de pele . . . . .	42
6.1	Resultados do Histograma . . . . .	51

# Capítulo 1

## Introdução

Visão computacional é uma das muitas áreas que tentam entender e simular a funcionalidade humana com a intenção de complementar o mundo com máquinas inteligentes. Para que exista uma interação homem-máquina compatível com esse processo, é necessário que o computador “enxergue” as pessoas. “Enxergar” as pessoas pode ser possível através da análise e cálculo de parâmetros básicos existentes em seres humanos, como movimentos, pele ou forma do corpo. A detecção da pele humana é uma das fases mais importantes no reconhecimento humano e se mostra essencial (na maioria das vezes) em um processo de interação homem-máquina.

Como citado em [12], a pele é, claramente, a primitiva mais utilizada nas pesquisas de processamento de imagens humanas. As aplicações que envolvem as informações contidas na pele variam desde detecção de faces e localização de pessoas até filtros pornográficos. As técnicas de detecção de pele podem ser simples e exatas, o que favorece a sua utilização em algumas aplicações comerciais (e.g. o localizador e classificador do olho de um motorista [41]<sup>1</sup>).

Como citado em [2], a detecção de pele pode ser definida como o processo de seleção de quais pixels em uma determinada imagem correspondem à pele humana. É importante salientar que um sistema de detecção de pele nunca é perfeito, e diferentes usuários utilizam diferentes critérios para classificação [8]. Por exemplo, James W. Davis <sup>2</sup> utiliza um algoritmo que, através de câmeras termais <sup>3</sup>, consegue detectar movimentos em filmes e/ou seqüências

---

<sup>1</sup>Esse programa, desenvolvido pela Ford ucraniana, tem por objetivo verificar se um motorista está se tornando sonolento, evitando assim que ele durma

<sup>2</sup><http://www.cse.ohio-state.edu/~jwdavis/>

<sup>3</sup>As câmeras termais medem a quantidade de calor em uma cena, onde o brilho de um pixel na imagem corresponde à quantidade de calor detectada naquele local. Essas câmeras funcionam de maneira equivalente em qualquer tipo de iluminação. Sendo assim,

de fotos. As imagens termais geradas por essas câmeras, quando comparadas com imagens geradas em câmeras comuns, fornecem uma assinatura única, completa e mais clara de humanos que se movem independente do fundo. A detecção de movimentos em sequências de fotos é possível armazenando-se informações a respeito das posições dos objetos nas fotos. Enquanto no primeiro quadro (*frame*) do filme a pessoa está em uma posição qualquer  $(x, y)$ , no próximo quadro ela pode estar posicionada em uma posição  $(x + 1, y + 1)$  (caso se mova diagonalmente pela imagem). Ou ainda, pode estar na mesma posição  $(x, y)$ , mas com um braço levantado, o que “mudaria o formato” da pessoa na imagem. Através da diferença entre os dois quadros é possível detectar, caso exista, o movimento realizado pela pessoa na sequência de frames.

As imagens possuem diversas informações importantes no estudo de detecção de pele. Por exemplo, os pixels pertencentes à pele humana não estão espalhados aleatoriamente pela imagem, mas sim agrupados em uma ou mais localidades [28]. Outra informação importante são as cores dos pixels da pele. Apesar de variarem bastante de uma pessoa para outra, a cor dos pixels da pele de uma mesma pessoa serão, na sua maioria, parecidos [28]. Ou seja, se foi detectado que um pixel faz parte da pele de uma pessoa, um outro pixel com um valor aproximado do primeiro (mesmo que esteja localizado em uma região diferente) será, muito provavelmente, também parte da pele desta pessoa. Logicamente, para realizar esse tipo de afirmação são necessários diversos cálculos estatísticos tais como média (centro de massa) e desvio padrão (espalhamento dos pixels). A cor da pele também possibilita um rápido processamento e uma robustez alta em variações geométricas das faces e mãos. Em diferentes casos são utilizados diferentes técnicas de detecção e, em algumas vezes, é necessário o uso de diversas técnicas em conjunto para atender às necessidades da aplicação que estará realizando o reconhecimento da pele humana.

A detecção da pele é útil em, por exemplo, reconhecimento de gestos, interação homem-máquina, multimídia, detecção e reconhecimento de faces e mãos (para segurança e aplicações de vídeo), teleconferência e muitas outras aplicações. Porém, existem algumas dificuldades quando tentamos detectar os pixels-pele. A cor da pele é afetada pela luz do ambiente que é desconhecida na maioria das situações. É preciso definir alguns limites na intensidade e na cor da iluminação variante. Além disso, câmeras diferentes produzem cores diferentes, mesmo partindo da mesma pessoa, sobre as mesmas condições de luz (iluminação invariante). E, finalmente, as cores da pele podem variar de pessoa para pessoa.

---

ela é muito recomendada para sistemas que necessitam de segurança e vigilância 24 horas.

A detecção e rastreamento de faces e mãos tem sido tópico de uma pesquisa extensiva durante muitas décadas passadas. Muitas heurísticas e estratégias baseadas em reconhecimento de padrões foram apresentadas na tentativa de encontrar uma solução robusta e exata. Dentre os métodos existentes (baseados em extração de características) para detecção de humanos, alguns utilizam a cor da pele como uma sugestão para detecção. A cor possibilita um rápido processamento e uma robustez alta em variações geométricas das faces e mãos. Além disso, a cor da pele possui uma cor característica, que é facilmente reconhecida por humanos.

A escolha de um sistema de detecção de pele é uma tarefa custosa, pois é preciso analisar diferentes características para cada caso. Espaços de cores diferenciados proporcionam diferentes desempenhos em diferentes métodos de detecção de pele.

Não é objetivo deste projeto explorar a fundo as características e algoritmos de cada método existente, mas sim comparar e analisar o desempenho de alguns métodos em alguns casos. Dessa maneira, uma busca por um método e/ou espaço de cores será uma tarefa mais trivial para pesquisadores que possuam um objetivo mais específico.

Para tanto, neste trabalho estão sendo analisadas algumas técnicas para detecção de pele. Paralelamente a isso, também está sendo realizada uma avaliação de alguns programas de análise e tratamento de imagens existentes.

Este trabalho faz parte de um módulo na plataforma SIGUS (Sistemas de apoio à Implementação de programas com interface GUIada por Sinais visuais <sup>4</sup>, ver seção 3.1). A plataforma SIGUS tem por objetivo desenvolver um ambiente computacional, utilizando programas-fonte livres para facilitar a implementação de sistemas com interface guiada por sinais visuais, integrando e complementando as bibliotecas de processamento digital de sinais, ImageJ <sup>5</sup>, de aprendizagem computacional, Weka <sup>6</sup> e de dispositivos adaptativos, AdapTools <sup>7</sup>.

Este trabalho foi elaborado na tentativa de facilitar a busca por algumas das diversas técnicas de visão computacional utilizadas para detecção de pele, considerando alguns métodos existentes em conjunto com algumas aplicações específicas da área. Para tanto, a estrutura deste estudo está idealizada na seguinte maneira: o capítulo 2 apresenta a Revisão Bibliográfica, com alguns estudos comparativos e trabalhos já realizados nos estudos de detecção de pele, juntamente com algumas definições importantes sobre Modelos Bayesianos e Histogramas. O 3 apresenta definições mais deta-

---

<sup>4</sup><http://www.ec.ucdb.br/pistori/projetos/sigus/>

<sup>5</sup>Disponível em <http://rsb.info.nih.gov/ij/>

<sup>6</sup>Disponível em <http://www.cs.waikato.ac.nz/~ml/weka/>

<sup>7</sup>Disponível em <http://www.pcs.usp.br/~lta>

lhadas a respeito da Plataforma SIGUS e onde este projeto se encaixa nos módulos existentes na plataforma. O capítulo 4 possui algumas definições básicas a respeito das cores e de suas diferentes representações em alguns espaços de cores. O capítulo 5 explica alguns Algoritmos e Modelos Estatísticos para detecção de pele, bem como o desempenho de diferentes métodos sobre o mesmo banco de imagens. O capítulo 6 apresenta duas ferramentas muito úteis para a construção de segmentadores de pele: ImageJ e Colorspace. Também foram analisados alguns módulos para o ImageJ e uma ferramenta muito interessante para visualização tridimensional de espaços de cores chamada RGBCube. O capítulo 7 apresenta as considerações finais com algumas conclusões e sugestões para trabalhos futuros.

# Capítulo 2

## Revisão Bibliográfica

### 2.1 Trabalhos Correlatos

Espaços de cores são modelos que representam as cores de diferentes maneiras. Muitos foram propostos na literatura para aplicações de detecção de pele. Alguns exemplos são *RGB* (*Red*/Vermelho, *Green*/Verde e *Blue*/Azul), *HSV* (*Hue*/Matiz, *Saturation*/Saturação, *Value*/Valor) e *CMYK* (*Cyan*/Ciano, *Magenta*/Magenta, *Yellow*/Amarelo, *Black*/Preto). *YCbCr* é um outro exemplo de espaço de cores e tem sido muito utilizado em aplicações de detecção de pele pois os pixels-pele formam um aglomerado compactado no plano  $Cb - Cr$  (ver seção 6.1.2). Como *YCbCr* também é usado em compactação de vídeo e nenhuma descompactação é necessária, este espaço de cores é muito utilizado em aplicações de detecção de pele onde a seqüência de vídeo é compactada [3],[43]. O espaço de cores *HSV* (*Hue*, *Saturation*, *Value*) foi proposto em [36],[33] porque é mais relacionado com as percepções das cores humanas. Em [44] dois componentes do espaço de cores *RGB* (*R* e *G*) foram propostos para minimizar a dependência da iluminação. Alguns autores [2] afirmam que para todo espaço de cores existe um detector de pele com desempenho ótimo. Na capítulo 3 os espaços de cores são explicados de uma maneira mais clara e detalhada.

Estudos de métodos de detecção de pele para espaços de cores específicos também têm sido realizados. Zarit, Benjamin e Quek em [5] compararam o desempenho de dois métodos de detecção de pele baseados em histogramas: *Lookup Table* e *Bayes Method* (seção 2.2.1). Cinco espaços de cores (*CIELAB*, *FleckHS*, *HSV*, *RGBnormalizado* e *YCrCb*) foram considerados para medir o desempenho destes dois métodos de detecção. Já em [17], nove espaços crominantes <sup>1</sup> foram comparados para avaliar o desem-

---

<sup>1</sup>Espaços de cores onde o brilho é desconsiderado.

penho de um modelo para a detecção de faces baseado nas métricas de Mahalanobis<sup>2,3</sup>. O modelo *TSL* (*Tint-Saturation-Luminance*) foi o mais bem sucedido, seguido pelo modelo *RG* (*Red-Green*)*normalizado* e pelo *CIE-xy*.

Foi mostrado em [39] que a pele apresenta uma grande quantidade da componente *R* do espaço de cores *RGB*. Assim, uma maneira simples (e trivial) de se realizar a detecção dos pixels-pele seria estabelecer o seguinte critério:

- *Um pixel localizado em um espaço de cores RGB será definido como parte da pele quando  $R > G$ ,  $R > B$  ou  $R$  for maior do que ambos ao mesmo tempo.*

Tal critério apresenta uma certa independência em nível de iluminação e pode ser aplicado muito rapidamente em um sistema. Porém, suas limitações são um tanto quanto óbvias. Por exemplo, a maioria dos ambientes (para não dizer todos) onde o sistema fosse aplicado poderia conter a cor vermelha em objetos que não façam parte da pele. Isso classificaria tais objetos como parte da pele, sendo que eles não se incluem nessa categoria.

Em [31] foi proposto um algoritmo para detecção de faces em imagens coloridas na presença de luminosidade variante e fundos complexos. Baseado em uma técnica para compensação de luz e uma transformação de cores não-linear, seu método detecta regiões da pele em toda a imagem e gera os objetos que possivelmente são rostos, baseado no arranjo espacial desses pixels que fazem parte da pele. Para realizar essa detecção, Hsu usou uma transformação no espaço de cores *YCbCr* e modelou a cor da pele usando a distribuição Gaussiana.

Saber e Tekalp usaram a transformação do espaço de cores *YES* para reduzir as variações na crominância causada pela intensidade da luz para tarefas de detecção de faces (ver [1]).

Sigal descreveu em [21] uma nova abordagem para a segmentação da pele em seqüências de vídeo. Para tanto, Sigal utilizou histogramas de cores adaptativos (histogramas que se atualizam dinamicamente, baseados na segmentação atual da pele e nas predições do modelo de Markov). O modelo de Markov é utilizado para prever a evolução da cor da pele no histograma utilizado (*HSV*, no caso) no decorrer da seqüência de vídeo. A evolução na distribuição da cor da pele é parametrizada a cada quadro pela translação, rotação e alteração de escala dentro do espaço de cores. As mudanças conseqüentes na parametrização geométrica da distribuição são propagadas ajustando-se o histograma.

<sup>2</sup><http://biologi.uio.no/fellesavdelinger/finse/spatialstats/Mahalanobis%20distance.ppt>

<sup>3</sup><http://www.ecs.csun.edu/~dsalomon/DC2advertis/DeCorr.pdf>

Nuria Oliver apresentou em [27] um sistema baseado em conjuntos de características bidimensionais de pixels compactos aglomerados espacialmente que pode realizar rastreamento, reconhecimento de formas e classificação de bocas e faces humanas. Esse sistema foi chamado de LAFTER (Lips and Face Real Time Tracker). Oliver afirma que no LAFTER os padrões de comportamento (expressões faciais e movimentos realizados pela cabeça) podem ser classificados em tempo real através de métodos baseados em HMM (*Hidden Markov Model*). De acordo com [27], seu sistema foi testado em centenas de usuários e apresentou uma bom desempenho, além de ter demonstrado confiabilidade. Sua taxa de acertos em reconhecimento, rastreamento e classificação estão próximos de 100%.

Em [26] foi apresentado um método de detecção que localiza pessoas nuas (conteúdos adultos) em imagens usando um filtro de pele. Para tanto foi utilizado um espaço de cores RGB transformado logaritmicamente (cf e.g.[13]). O filtro de pele é criado baseando-se em duas informações básicas: (1) a representação de cores baseada no componente logarítmico e (2) a intensidade do componente verde no espaço RGB (o componente verde é usado para representar intensidade porque, dependendo do tipo de sensor das câmeras, os componentes vermelho e azul podem ter uma baixa resolução espacial). O algoritmo de Fleck primeiramente localiza as imagens que contém a cor da pele em regiões extensas. Realizada essa detecção, o algoritmo procura por regiões alongadas e grupos delas em possíveis membros humanos. O algoritmo também procura, paralelamente às regiões alongadas, por membros conectados, usando dados que possuem informações sobre a estrutura dos objetos que podem ser comparados à conteúdos adultos.

## 2.2 Conceitos Auxiliares

Nesta seção algumas definições auxiliares para a compreensão do projeto serão apresentadas. A subseção 2.2.1 explica rapidamente o Teorema de Bayes e sua origem. A subseção 2.2.2 apresenta algumas características importantes a respeito dos histogramas, que serão utilizados posteriormente na seção 6.1.2 do capítulo 6.

### 2.2.1 Teorema de Bayes

Thomas Bayes nasceu em 1702 em Londres, Inglaterra. Foi o autor do Teorema de Bayes através do artigo *Essay towards solving a problem in the doctrine of chances*, publicado postumamente. Recentemente foram publicadas novas teorias de Bayes escritas em 1755, no artigo [4].

O Teorema de Bayes é utilizado na inferência estatística para atualizar estimativas da probabilidade de que diferentes hipóteses sejam verdadeiras, baseado nas observações e no conhecimento de como essas observações se relacionam com as hipóteses. A probabilidade direta de uma hipótese  $H$  condicionada à um corpo de dados  $E$  ( $P(H|E)$ ) está relacionada com o inverso da probabilidade dos dados  $E$  condicionados à hipótese  $H$  ( $P(E|H)$ ). A equação 2.1 explica matematicamente o que foi introduzido acima.

$$P(H|E) = P(E|H) \frac{P(H)}{P(E)} \quad (2.1)$$

Explicações mais detalhadas a respeito do Teorema de Bayes podem ser encontrada em [24] e [10].

### 2.2.2 Histogramas

O histograma de uma imagem é uma ferramenta valiosa usada para visualizar o perfil da intensidade de uma imagem. Ele provê informações sobre o contraste e a distribuição global da intensidade de uma imagem. O histograma de uma imagem é simplesmente um gráfico de barras da intensidade dos pixels. As intensidades dos pixels são plotadas ao longo do eixo  $x$  e o número de ocorrências para cada intensidade representa o eixo  $y$ .

As imagens escuras possuem histogramas com distribuições de pixels mais aglomeradas na esquerda. Imagens mais brilhantes possuem distribuições de pixels mais ao lado direito do histograma. Em uma imagem ideal, existe uma distribuição uniforme dos pixels sobre o histograma. Na seção 6.1.2 é apresentado um módulo do ImageJ que mostra o histograma de uma imagem RGB.

# Capítulo 3

## Plataforma SIGUS

Neste capítulo será apresentada uma breve introdução a respeito da plataforma SIGUS e seus principais objetivos, bem como onde esse projeto se insere na plataforma.

### 3.1 Introdução

SIGUS <sup>1</sup> é um ambiente de apoio ao desenvolvimento de aplicações com interfaces guiadas total ou parcialmente por sinais visuais. A idéia do projeto SIGUS se iniciou no projeto “Detecção da Direção do Olhar utilizando Visão Computacional e Tecnologias Adaptativas” <sup>2</sup>, de Hemerson Pistori e Eduardo Rocha Costa. Sua fundamentação se baseia no baixo custo dos dispositivos de captura de imagens, juntamente com o aumento significativo na capacidade de processamento dos computadores pessoais. Esses fatores abrem espaço para o desenvolvimento de novos tipos de interfaces homem-máquina. A substituição dos tradicionais teclados e mouses por sinais visuais, a serem apresentados ao computador através de uma filmadora digital, oferece uma variada gama de aplicações que podem viabilizar e facilitar a utilização de computadores por segmentos da sociedade que apresentam determinadas deficiências físicas. Editores de texto e navegadores para Internet acionados através de movimentos do globo ocular, por exemplo, poderiam ser utilizados por pessoas tetraplégicas. Surdos também poderiam se beneficiar deste tipo de sistema, com a introdução de programas de computador capazes de interpretar línguas de sinais. Outras aplicações interessantes nesta área incluem jogos e controle de aparelhos domésticos utilizando sinais manuais.

---

<sup>1</sup><http://www.ec.ucdb.br/~pistori/projetos/sigus/>

<sup>2</sup><http://www.pcs.usp.br/~lta/vc/>

Uma boa parte dos experimentos desenvolvidos nesta área integram técnicas de processamento de imagens digitais e de aprendizagem de máquina. Embora existam algumas ferramentas livres de apoio ao desenvolvimento de diferentes módulos de um sistema guiado por sinais visuais, não temos conhecimento da existência de uma ferramenta integrada, especializada, e de fácil utilização, para auxílio ao desenvolvimento deste tipo de sistema.

## 3.2 Objetivos e Contexto deste Projeto

O principal objetivo da Plataforma SIGUS é desenvolver um ambiente computacional, com programas-fonte livres, para facilitar a implementação de sistemas com interface guiada por sinais visuais, integrando e complementando as bibliotecas de processamento digital de sinais, ImageJ <sup>3</sup>, de aprendizagem computacional, Weka <sup>4</sup> e de dispositivos adaptativos, AdapTools <sup>5</sup>.

Para cumprir com esse objetivo o projeto SIGUS propõe os seguintes objetivos específicos:

1. Complementar e integrar as bibliotecas ImageJ e Weka, criando uma interface gráfica que permita fácil acesso aos seus recursos.
2. Desenvolver um módulo estatístico, baseado nas implementações disponíveis no pacote Weka, que permita a comparação do desempenho de diferentes estratégias de aprendizagem automática, incluindo mecanismos para análise estatística de desempenho em ambientes de tempo real com aprendizagem incremental.
3. Construir filtros especializados para detecção de sinais visuais.
4. Facilitar a utilização de dispositivos de captura de imagens de baixo custo no desenvolvimento de aplicativos Java.
5. Integrar a Plataforma SIGUS com a ferramenta de apoio à aplicação de tecnologia adaptativa, AdapTools.
6. Implementar um protótipo de um editor de textos para a Língua Brasileira de Sinais (LIBRAS)
7. Criar um banco de imagens de sinais LIBRAS para ser utilizado em experimentos com algoritmos de aprendizagem e reconhecimento de padrões.

---

<sup>3</sup>Disponível em <http://rsb.info.nih.gov/ij/>

<sup>4</sup>Disponível em <http://www.cs.waikato.ac.nz/~ml/weka/>

<sup>5</sup>Disponível em <http://www.pcs.usp.br/~lta>

8. Implementar um protótipo de uma interface para um controlador de cadeira de rodas guiado pela direção do olhar.
9. Produzir material didático e de divulgação para sistemas guiados por sinais visuais e Plataforma SIGUS.

Cada etapa apresentada acima está sendo elaborada por um ou mais desenvolvedores. A função deste projeto na Plataforma SIGUS é realizar uma contribuição na construção de filtros especializados para detecção de sinais visuais. Essa contribuição consiste no estudo de tecnologias e teorias disponíveis para segmentação de partes do corpo humano para que possam ser determinados quais filtros devem ser implementados na Plataforma SIGUS.

# Capítulo 4

## Espaços de Cores

Este capítulo apresenta alguns estudos sobre algumas características das cores e suas diferentes formas de representação através dos espaços de cores. A seção 4.1 apresenta alguns conceitos sobre cores em geral, espectro de cores e classificação das cores. A seção 4.2 apresenta alguns espaços de cores importantes para a detecção de pele, bem como as transformações entre espaços de cores e suas representações gráficas.

### 4.1 Cor

#### 4.1.1 Cores-luz e cores-pigmento

Cor é uma sensação produzida no nosso cérebro pela luz que chega aos nossos olhos. Existem duas linhas de pensamentos distintas ao falarmos de cores em geral: as cores-luz e as cores-pigmento. As cores-pigmento são as cores que encontramos na natureza. As cores-luz são aquelas geradas pelo fenômeno da decomposição da luz branca (através de um prisma). A decomposição da luz é mostrada na figura 4.1 <sup>1</sup>.

As figuras 4.2 e 4.3 apresentam a decomposição da luz seguida da recomposição através de dois e quatro prismas, respectivamente.

Os dois extremos da classificação das cores-luz são o branco (ausência total de cor, luz pura) e o preto (ausência total de luz, nenhuma cor é refletida). Porém, não é necessário somar todas as cores-luz para se obter a cor branca. A figura 4.4 mostra como a cor branca pode ser obtida através da soma das três cores primárias (vermelho + verde + azul).

Subtrair cores consiste em eliminar uma ou mais componentes da luz. Por exemplo, misturar tintas equivale a subtrair cores. É sabido que tinta

---

<sup>1</sup>As figuras 4.1, 4.2 e 4.3 foram retiradas de <http://www.fisica.ufc.br/coresluz1.htm>

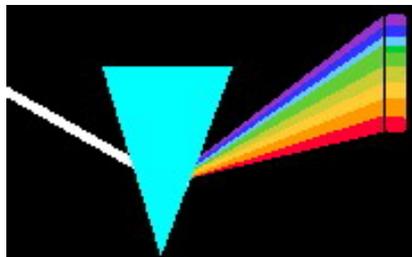


Figura 4.1: Decomposição e recomposição da luz



Figura 4.2: Decomposição e recomposição da luz



Figura 4.3: Decomposição e recomposição da luz

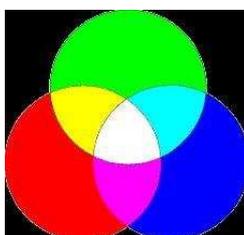


Figura 4.4: Soma das três cores primárias

azul misturada com tinta amarela nos dá tinta verde. Isso ocorre porque os pigmentos da tinta azul absorvem as componentes do lado vermelho e os pigmentos da tinta amarela absorvem as componentes do lado azul. Sobram as componentes intermediárias, isto é, o verde.

A figura 4.5 mostra como a cor preta é obtida através da subtração das três cores secundárias (azul-turquesa - magenta - amarelo).

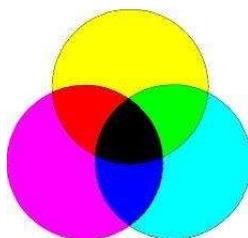


Figura 4.5: Soma das três cores secundárias

Dessa maneira, podemos dizer que a presença de todas as cores-luz nos dá a cor branca e a presença de todas as cores-pigmento nos dá a cor preta. Da mesma maneira, a ausência de todas as cores-luz nos dá a cor preta e a ausência de todas as cores-pigmento nos dá a cor branca.

### 4.1.2 Classificação da cor

Os teóricos classificam as cores com três dimensões: tom, valor e saturação. Para representar essa classificação elaboraram-se vários esquemas como a pirâmide de Lambert (figura 4.6 <sup>2</sup>), o duplo cone de Ostwald e a ordenação criada por Munsell <sup>3</sup>. Estes sistemas baseiam-se no mesmo princípio. O eixo vertical representa as escalas de valores acromáticos, que vão desde o branco na parte superior até o negro na inferior. O círculo correspondente ao equador contém a escala dos tons que possuem, nesta altura, um valor de claridade médio. Cada uma das seções horizontais do sólido apresenta os valores cromáticos num dado nível de claridade. Quanto mais afastado do eixo central, mais saturada será a cor. Quanto mais próximo, mais misturada com um cinza de mesmo valor.

Explicando-se as três dimensões, temos que o tom pode ser definido como a qualidade que distingüe uma cor de outra (por exemplo, um amarelo de um vermelho). Corresponde ao comprimento de onda dominante. A figura 4.7 <sup>4</sup> mostra um espectro de cores e seus respectivos valores de comprimento

<sup>2</sup>Figura retirada de <http://www.colorsystm.com/projekte/engl/11lame.htm>

<sup>3</sup><http://www.georgehernandez.com/xComputers/Color/Systems.htm>

<sup>4</sup>Figura retirada de <http://www.fisica.ufc.br/coresluz4.htm>

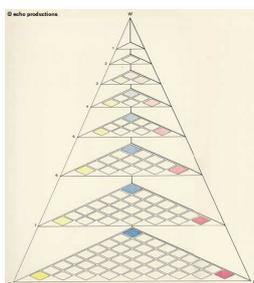


Figura 4.6: Pirâmide de Lambert

de onda. Cada cor pode ter diferentes valores, de acordo com o seu grau de claridade ou obscuridade refletida (por exemplo, um vermelho claro tem valor mais alto do que um vermelho escuro). Desta maneira, valor significa a quantidade de luz que uma superfície tem a capacidade de refletir. A saturação corresponde ao grau de intensidade ou croma, e relaciona-se com a pureza ou a opacidade da cor.

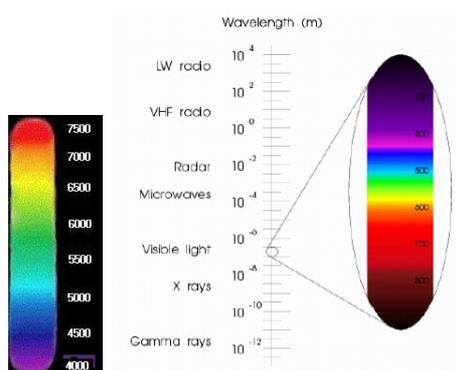


Figura 4.7: Dois exemplos de espectro de cores

### 4.1.3 Crominância e Luminância

A luminância é a componente da imagem que só contém as informações de brilho (tons de cinza) de uma imagem, ou seja, os valores entre o preto e o branco. Podemos dizer que a luminância é a principal componente de uma imagem, que lhe distingue por sua nitidez e qualidade.

A crominância é a componente que agrega as cores a uma imagem. A crominância ou croma de uma imagem confere o colorido, sem acrescentar entretanto maior riqueza nos detalhes que já devem estar representados pela luminância. O sinal de croma possui menos largura de faixa do que o sinal

de luminância.

Muitas técnicas de visão computacional utilizam espaços de cores crominantes (ver seção 4.2) para a detecção de pele por sua separação bem definida entre crominância e luminância.

## 4.2 Espaços de Cores

Esta seção explica os conceitos a respeito de alguns espaços de cores utilizados nas técnicas para segmentação de pele.

### 4.2.1 RGB

RGB é um espaço de cores formado por três cores primárias (vermelho, verde e azul). É um dos espaços de cores mais utilizado para processamento e armazenamento de dados digitais de imagens[7],[18]. Entretanto, a alta relação entre os canais (muitos objetos do mundo real possuem essas três cores misturadas), o percentual significativo de não-uniformidade e a mistura dos dados de crominância e luminância fazem do RGB uma escolha não muito favorável para análise de cores e algoritmos baseados em reconhecimento de cores.

As figuras 4.9 (Gerada no RGBCube <sup>5</sup>) e 4.8 <sup>6</sup> apresentam o cubo tridimensional de cores RGB. Também é possível observar as cores secundárias azul-turquesa (*Cyan*), magenta (*Magenta*) e amarelo (*Yellow*).

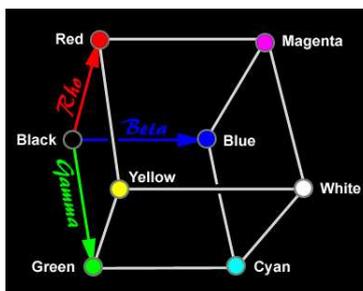


Figura 4.8: Cubo RGB

### 4.2.2 RGB Normalizado

O RGB Normalizado é uma representação facilmente obtida através dos valores RGB após um procedimento de normalização que apresentamos abaixo

<sup>5</sup><http://www.couleur.org/>

<sup>6</sup>Extraído de <http://www.photo.net/photo/edscott/vis00020.htm>

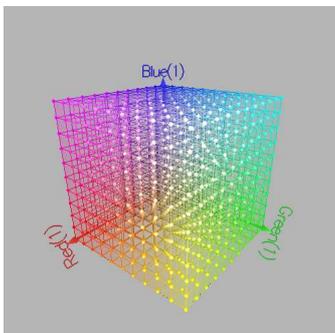


Figura 4.9: Espaço de cores RGB, gerado no RGBCube

na equação 4.1:

$$\begin{aligned} r &= \frac{R}{R + G + B} \\ g &= \frac{G}{R + G + B} \\ b &= \frac{B}{R + G + B} \end{aligned} \quad (4.1)$$

Como a soma dos três componentes normalizados é conhecida ( $R + G + B = 1$ ), o terceiro componente ( $b$ ) não mantém nenhuma informação significativa, e pode ser omitido com o intuito de reduzir a dimensão espacial. Os componentes restantes são chamados de “cores puras”, pois a normalização diminui a dependência dos componentes  $r$  e  $g$  ao brilho do espaço de cores  $RGB$ . Uma propriedade notável deste espaço de cores é que, quando ignoramos a luz do ambiente em superfícies opacas, o RGB normalizado é invariante às mudanças de orientação das fontes de luz na superfície [35]. Isto, em conjunto com a simplicidade na transformação ajuda este espaço de cores a ganhar popularidade entre os pesquisadores.

### 4.2.3 HSV, HSL, HSI - Matiz, Saturação, Valor

Espaços baseados em matiz e saturação foram apresentados quando havia uma necessidade de se especificar as propriedades das cores numericamente. Esses espaços mostram as cores com valores intuitivos, baseados na idéia de matiz, saturação e valor. A matiz é relacionada à cor em si e define a cor dominante de uma área. A matiz diferencia o azul do vermelho, por exemplo. A saturação mede a pureza da cor da área. A grosso modo, a saturação é a característica que diferencia a cor rosa da cor vermelha. Enquanto a cor

vermelha é uma cor pura, a cor rosa é um vermelho com alguma quantidade da cor branca. A intensidade, brilho ou valor é relacionada à luminância da cor. Diferencia o claro do escuro em cada cor da matiz.

O fato de os componentes serem intuitivos e de existir uma discriminação entre as propriedades da luminância e da crominância faz esse espaço de cores ser popular nos trabalhos de segmentação da cor da pele [5],[25],[21]. Foram apresentadas muitas propriedades interessantes da matiz em [35]. Porém, existem fortes características não desejáveis destes espaços de cores, incluindo discontinuidade da matiz e o custo computacional no tratamento do “brilho” (luminância, valor ou intensidade) <sup>7</sup>. Essas características são um fator agravante quando tentamos realizar as técnicas de visão computacional para detecção de pele.

A transformação do espaço  $RGB$  para o espaço de cores  $HSV$  é efetuada pela fórmula 4.2 abaixo:

$$\begin{aligned} H &= \arccos \frac{\frac{1}{2}((R - G) + (R - B))}{\sqrt{((R - G)^2 + (R - B)(G - B))}} \\ S &= 1 - 3 \frac{\min(R, G, B)}{R + G + B} \\ V &= \frac{1}{3}(R + G + B) \end{aligned} \quad (4.2)$$

A figura 4.10 <sup>8</sup> apresenta o cone HSV. Nesta representação, a matiz é descrita como uma formação tridimensional da roda de cores. A saturação é representada pela distância do centro até a “borda” do cone e o valor é a distância até o “pico” do cone.

O espaço  $HSL$  é uma das possíveis variantes do  $HSV$ , em que as componentes de saturação e luminância são calculadas de maneira um pouco diferente. As fórmulas abaixo descrevem as diferenças entre os espaços de cores  $HSL$  e  $HSV$ :

- Para o espaço de cores  $HSL$ :
  - saturação =  $\max(R, G, B) - \min(R, G, B)$
  - pureza =  $(\max(R, G, B) + \min(R, G, B))/2$
- Para o espaço de cores  $HSV$ :
  - saturação =  $(\max(R, G, B) - \min(R, G, B))/\max(R, G, B)$

<sup>7</sup>Ver <ftp://www.inforamp.net/pub/users/poynton/doc/colour/ColorFAQ.ps.gz>

<sup>8</sup>Extraído de [http://en.wikipedia.org/wiki/Image:HSV\\_cone.jpg](http://en.wikipedia.org/wiki/Image:HSV_cone.jpg)

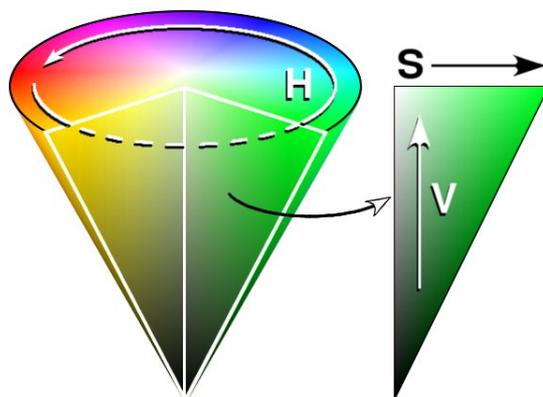


Figura 4.10: Cone HSV

– valor =  $\max(R, G, B)$

A tabela 4.1 apresenta três exemplos de transformações  $RGB$  para  $HSL$  e  $RGB$  para  $HSV$ . A coluna “Espaço de Cores” apresenta os espaços de cores onde estão sendo realizadas as transformações. A coluna “Componente” apresenta qual o componente correspondente à linha. A coluna “Exemplo I”, “Exemplo II” e “Exemplo III” apresentam os valores de 3 pixels nos espaços de cores respectivos às linhas.

Espaço de Cores	Componente	Exemplo I	Exemplo II	Exemplo III
$RGB$	$R$	1	0.5	0
	$G$	0	1	0.5
	$B$	0	0.5	0
$HSL$	$H$	0	120	240
	$S$	1	0.5	0.5
	$L$	0.5	0.75	0.5
$HSV$	$H$	0	120	240
	$S$	1	1	1
	$V$	1	1	0.5

Tabela 4.1: Transformações  $RGB/HSL/HSV$ 

Uma outra variante do espaço  $HSV$  é o espaço  $HSI$ . As figuras 4.11, 4.12, 4.15 e 4.16 ilustram esses dois espaços de cores.

Na figura 4.11 é possível observar a variação da componente  $H$  representando a matiz, que vai desde a região mais à esquerda (mais rosa e avermelhada) até a região mais à direita (mais azulada). A componente varia

em diversas cores. Sai do vermelho, passa para o laranja, depois segue no amarelo, verde e por fim azul. Também é possível visualizar a variação da matiz na figura 4.12.

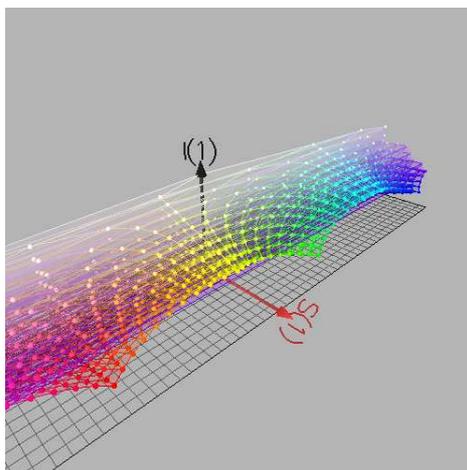


Figura 4.11: Espaço de cores HSI, gerado no RGBCube

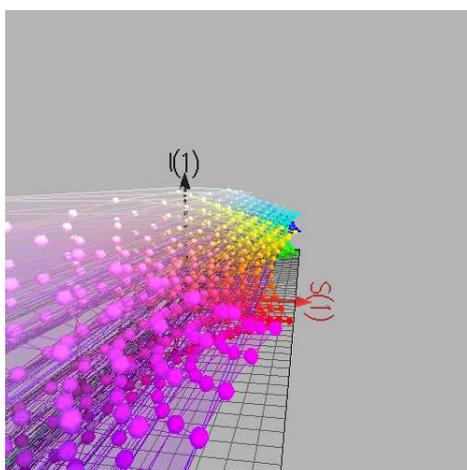


Figura 4.12: Espaço de cores HSI, gerado no RGBCube

A figura 4.13 apresenta uma boa visão da componente  $I$  (intensidade). Examinando a linha  $I$  no sentido contrário ao do eixo, é possível perceber uma variação da cor branca até a cor preta.

Na figura 4.14 algumas características do espaço tridimensional foram modificadas para facilitar o entendimento da componente  $S$ . A variação da saturação é bem visível quando concentramos a observação no eixo  $S$ .

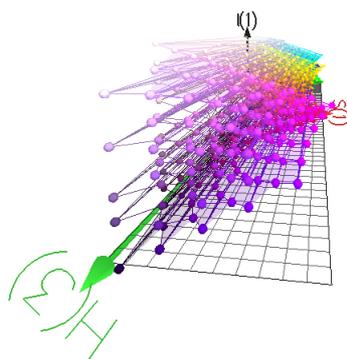


Figura 4.13: Espaço de cores HSI, gerado no RGBCube

Perceba que quanto mais saturada está uma cor, mais próxima ela está do final do eixo.

Na figura 4.15 é possível notar a variação da matiz no espaço de cores *HSV*.

Com a figura 4.16 (que também está representando o espaço de cores *HSV*) é possível perceber a variação da componente *I* e da componente *S* sobre uma cor púrpura. Observar que, da direita para a esquerda varia de um cor de rosa até um roxo bem puro e de cima para baixo varia de uma cor mais clara para uma cor mais escura. No ponto esquerdo superior é possível observar que a cor inicial é branca e vai se tornando preta conforme dirigimos nossa visão diagonalmente para baixo e para a direita.

#### 4.2.4 TSL - Matiz, Saturação, Luminância

Um espaço normalizado crominante-luminante TSL é uma transformação do RGB Normalizado em valores mais intuitivos, próximos da matiz e saturação no significado original. A principal característica do espaço de cores *TSL* é a diferença no cálculo da matiz quando comparado com os espaços de cores *HSI*, *HSV* e *HSL*. A equação 4.3 apresenta a fórmula para transformação do espaço de cores *RGB* para o espaço de cores *TSL*.

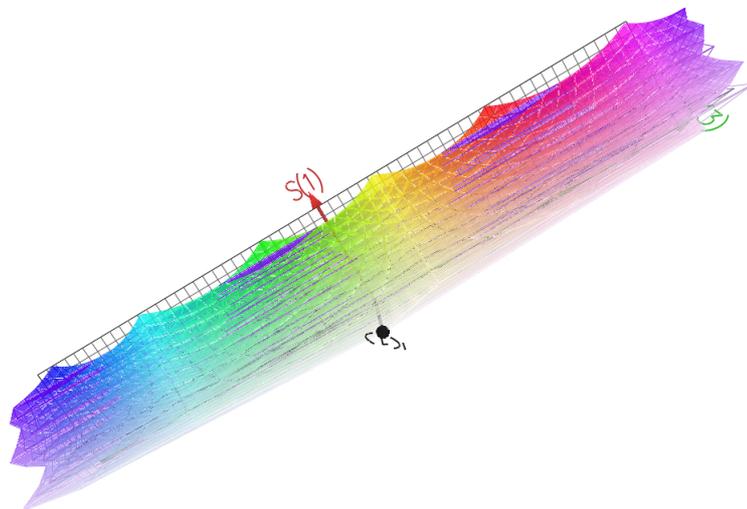


Figura 4.14: Espaço de cores HSI, gerado no RGBCube

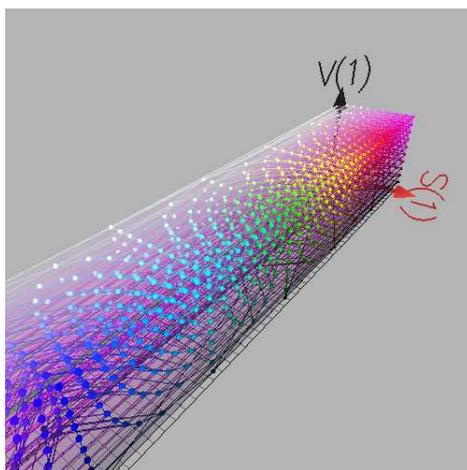


Figura 4.15: Espaço de cores HSV, gerado no RGBCube

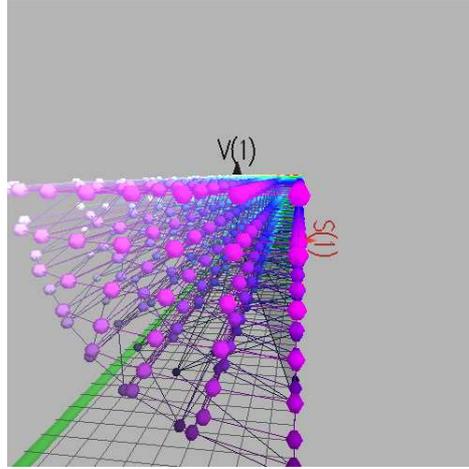


Figura 4.16: Espaço de cores HSV, gerado no RGBCube

$$\begin{aligned}
 S &= [9/5(r'^2 + g'^2)]^{1/2} \\
 T &= \begin{cases} \arctan(r'/g')/2\pi + 1/4, & g' > 0 \\ \arctan(r'/g')/2\pi + 3/4, & g' < 0 \\ 0, & g' = 0 \end{cases} \\
 L &= 0.299R + 0.587G + 0.114B
 \end{aligned}$$

Onde  $r'$  é igual a  $r - 1/3$ ,  $g'$  é igual a  $g - 1/3$  e  $r, g$  são provenientes da equação 4.1. Através de distribuições Gaussianas, Terrillon conclui em [17] que o espaço de cores  $TSL$  é superior aos outros oito espaços de cores utilizados em seu trabalho.

### 4.2.5 YCbCr

$YCbCr$  é um sinal codificado  $RGB$  não-linear, comumente utilizado pelos estúdios de televisão europeus e para trabalhos de compressão de imagens (arquivos MPEG são compactados em  $YCbCr$ ). A cor é representada pelo brilho ou luma (luminância calculada do  $RGB$  não-linear<sup>9</sup>), construída através de três parâmetros: (1) uma soma dos pesos dos valores  $RGB$  ( $Y$ ), (2) o valor  $Cb$  e (3) o valor  $Cr$ . Os valores  $Cb$  e  $Cr$  são obtidos subtraindo-se a luma dos valores vermelho ( $R - Y$ ) e azul ( $B - Y$ ) do espaço de cores  $RGB$ , respectivamente. A equação 4.3 apresenta o processo de transição do espaço de cores  $RGB$  para o  $YCbCr$ .

<sup>9</sup>Ver <ftp://www.inforamp.net/pub/users/poynton/doc/colour/ColorFAQ.ps.gz>

$$\begin{aligned}
 Y &= 0.299R + 0.587G + 0.114B \\
 C_r &= R - Y \\
 C_b &= B - Y
 \end{aligned}
 \tag{4.3}$$

A simplicidade na transformação e a separação explícita dos componentes luminância e crominância faz este espaço de cores se tornar atrativo para modelagem das cores dos pixels pele[29],[5],[31]. Na sub-seção 6.1.2 abaixo iremos analisar um módulo do ImageJ para detecção de pele que utiliza esse espaço de cores.

As figuras 4.17, 4.18 e 4.19 mostram graficamente o espaço de cores YCbCr. Em todas as três figuras é possível observar como a crominância se espalha pelos eixos  $Cb$  e  $Cr$ . Porém, somente na figura 4.19 que é possível observar totalmente a variação da luminância pelo eixo  $Y$ . Observem que para  $Y$  mínimo a cor mostrada é preta para qualquer crominância e para  $Y$  máximo a cor mostrada é branca para qualquer crominância.

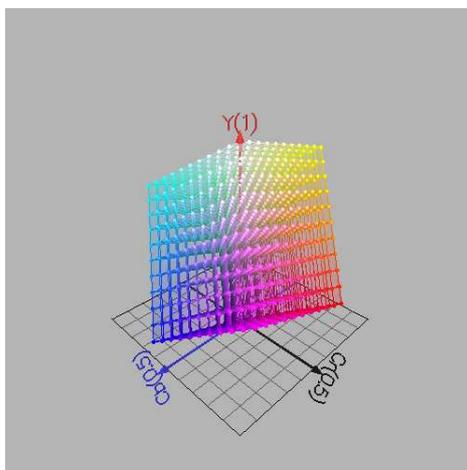


Figura 4.17: Espaço de cores YCbCr, gerado no RGBCube

### 4.2.6 Outros Espaços de Cores

Além do YCrCb, diversas outras transformações lineares do espaço de cores RGB foram implementadas para detecção de pele, como por exemplo, YES [1], YUV [23], YIQ [7],[11] e CIE-xyz [17].

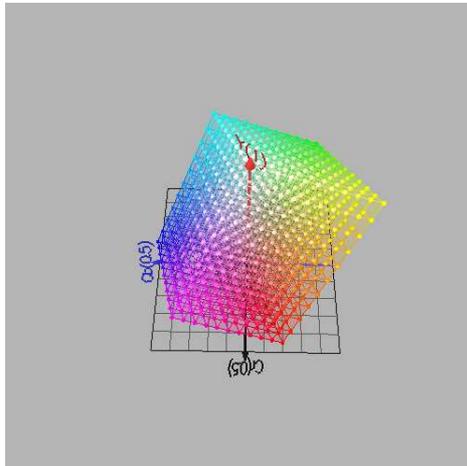


Figura 4.18: Espaço de cores YCbCr, gerado no RGBCube

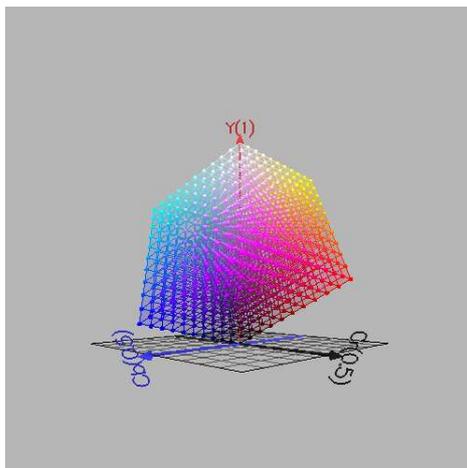


Figura 4.19: Espaço de cores YCbCr, gerado no RGBCube

### 4.2.7 Espaços de Cores Independentes de Dispositivos

Um espaço de cores dependente de dispositivos é um espaço de cores onde a cor resultante depende do equipamento e do sistema utilizado para produzi-lo. Por exemplo a cor produzida utilizando valores de pixels  $RGB = \{250, 134, 67\}$  será alterada conforme a variação do brilho e do contraste utilizado no monitor. Da mesma maneira, se alterarmos variações dos componentes vermelho, verde e azul do monitor, teremos características ligeiramente diferentes em espaços de cores luminantes/crominantes e a cor produzida irá mudar. Desde que  $RGB$  é um espaço de cores que é dependente do sistema utilizado, ele é um espaço de cores dependente de dispositivos. Um espaço de cores independente é um espaço de cores onde as coordenadas usadas para especificar a cor produzirão as mesmas cores em qualquer sistema aplicado. Um exemplo de espaço de cores independente de dispositivos é o  $CIEL^*a^*b^*$  (conhecido como  $CIELAB$  e baseado no sistema de cores humano).

### 4.2.8 Comparações entre espaços de cores

Em uma primeira análise, a seleção do espaço de cores parece ser crucial para as técnicas de detecção de pele que utilizam a cor como elemento principal. Uma questão importante é: qual é o melhor espaço de cores para a detecção de pele? Ou, refazendo a pergunta: há um espaço de cores ótimo para classificação de pele? Surpreendentemente, muitos artigos relacionados à detecção e classificação de pele não apresentam uma justificativa convincente para a escolha do espaço de cores. Isso provavelmente ocorre pela possibilidade de se obter, em quase todo espaço de cores, resultados aceitáveis na detecção de pele (quando uma base de dados limitada é utilizada). Apenas alguns poucos artigos utilizados para a detecção de pele se dedicaram à análise comparativa de diferentes espaços de cores [5],[17],[15],[16],[37]. Muitos autores consideraram seriamente o problema da seleção do espaço de cores e apresentaram justificativas para a optimalidade (ou adequação) da sua escolha para o modelo de pele que eles escolheram [46],[45],[38],[34]. Um espaço de cores “adequado” para a modelagem da pele é geralmente avaliado por dois conjuntos de parâmetros distintos. O primeiro conjunto de parâmetros é o erro de classificação do conjunto de treinamento, geralmente calculado após definidos os parâmetros do modelo de cores. É um princípio muito conhecido para a avaliação do desempenho dos classificadores. Em [42] Vezhnevets afirma que esse princípio define claramente quão ótimo é o ajuste do modelo escolhido ao conjunto de imagens. O segundo conjunto de parâmetros são as cores de pele e não-pele sobrepostas (objetos que possuam uma cor parecida com a

da pele e estejam próximos dos pixels pele) e a densidade do aglomerado de pele. Essas medidas são independentes à estratégia de modelagem de cores e são determinadas para avaliar o desempenho da detecção de pele ‘em geral’. Com essas medidas é possível ter uma impressão global bem segura a respeito da distribuição dos pixels pele e dos pixels não-pele nas amostras do conjunto de treinamento, mas sua viabilidade para a avaliação do espaço de cores é duvidosa.

A avaliação da influência dos espaços de cores ‘em geral’ (incluindo sobreposição de pixels pele/não-pele, forma dos aglomerados de pele, etc) indiferentemente a algum método de modelagem, não consegue dar a impressão de quão bom o espaço de cores é para a modelagem, pois diferentes métodos de detecção reagem de maneiras diversas às mudanças de espaços de cores.

# Capítulo 5

## Modelos Estatísticos e Algoritmos

Este capítulo explicará alguns modelos estatísticos e algoritmos utilizados para segmentação de pele.

### 5.1 Técnica baseada em Tabelas de Conversão (*Look-Up Table*)

Aqui apresentaremos um método que segmenta os pixels-pele do resto da imagem através de uma técnica baseada em tabelas de conversão.

Para avaliar o desempenho desta técnica, as imagens que fazem parte do conjunto de treinamento precisam estar previamente segmentadas nos pixels-pele. Com isso teremos a posição e o valor (para cada componente de cada espaço de cores) de todos os pixels-pele existentes na figura. Por exemplo, em um espaço de cores  $YCbCr$  teremos a posição  $(x, y)$  do pixel-pele e os valores  $Y, Cb$  e  $Cr$  do espaço de cores. Para cada valor de cada espaço de cores existe um histograma (ver 2.2.2 e 6.1.2) que representa o valor propriamente dito no eixo  $x$  e o número de vezes que ele ocorre na imagem no eixo  $y$ .

Para fins explicativos, assumiremos que todos os outros pixels da imagem (pixels não-pele que foram retirados) são pretos e utilizaremos somente dois componentes de um espaço de cores para representar a pele. No caso, o espaço de cores escolhido para a explicação é  $RGB$  Normalizado e os componentes são  $r$  e  $g$  (ver seção 4.2).

Uma tabela (ver tabela 5.1 abaixo) é montada com o eixo  $x$  contendo as informações de  $r$  variando do mínimo valor encontrado no histograma de  $r$  até o valor máximo encontrado no histograma e com o eixo  $y$  contendo as informações de  $g$ , variando da mesma maneira.

Na tabela 5.1 apresentada abaixo é apresentada a variação de  $g.m \rightarrow g.M$  ( $g$  mínimo até  $g$  Máximo) e a variação de  $r.m \rightarrow r.M$  ( $r$  mínimo até  $r$

Máximo).

$g.M$	0	0	0	0	0	0	0	0	0
$\vdots$	0	0	0	0	0	0	0	0	0
$\vdots$	0	0	0	0	0	0	0	0	0
$\vdots$	0	0	0	0	0	0	0	0	0
$\vdots$	0	0	0	0	0	0	0	0	0
$\vdots$	0	0	0	0	0	0	0	0	0
$\vdots$	0	0	0	0	0	0	0	0	0
$\vdots$	0	0	0	0	0	0	0	0	0
$\vdots$	0	0	0	0	0	0	0	0	0
$g.m$	0	0	0	0	0	0	0	0	0
	$r.m$	...	...	...	...	...	...	...	$r.M$

Tabela 5.1: Tabela  $r/g$

Como a imagem que estamos trabalhando já está segmentada nos pixels-pele, todos os pixels que são diferentes de preto são pele. Assim, para cada pixel diferente de preto encontrado, incrementa-se uma célula da tabela. A posição do valor marcado na tabela corresponde ao valor  $r$  e  $g$  daquele pixel. Assim, se os únicos pixels-pele detectados na imagem forem 15 pixels com valores de  $r = 7$  e  $g = 5$ , e 8 pixels com valores de  $r = 2$  e  $g = 3$ , teremos a seguinte tabela 5.2:

5	0	0	0	0	0	0	0	15	0
4	0	0	0	0	0	0	0	0	0
3	0	0	8	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
	0	1	2	3	4	5	6	7	8

Tabela 5.2: Tabela  $r/g$

A tabela 5.2 é a tabela de conversão da imagem. Para calcular a probabilidade de um pixel fazer parte da pele, temos que saber o valor das componentes  $r$  e  $g$  desse pixel. Tendo esses valores em mãos, é necessário localizar o valor correspondente ao número dos eixos na tabela de conversão

(por exemplo, se o pixel tem o valor 7 para  $r$  e 5 para  $g$ , o valor correspondente seria 15). Depois de localizado esse valor, é necessário dividi-lo pela soma de todos os valores da tabela. O resultado será a probabilidade desse pixel fazer parte da pele. A equação 5.1 explica matematicamente o que foi dito.

$$P(r = 7, g = 5|pele) = \frac{15}{(15 + 8)} = 0.65 = 65\% \quad (5.1)$$

Como mostrado na equação 5.1 e baseando-se na tabela de conversão 5.2, a probabilidade de um pixel com componentes  $r = 7$  e  $g = 5$  pertencer à pele é de 65%.

## 5.2 Método de Bayes

Como mostrado em [47],[10],[24] e também na equação 2.1 da seção 2.2.1 deste trabalho, o Teorema de Bayes escolhe a hipótese mais provável, dado o valor de uma característica escolhida.

Obedecendo às regras estabelecidas em [5] para o Método de Bayes, existem duas classes mutuamente exclusivas quando aplicamos este teorema na detecção de pele: pele ( $s$ ) e não-pele ( $\neg s$ ). A característica citada acima e utilizada aqui é o valor bidimensional do pixel, que chamaremos de  $\bar{x}$ . Também são utilizados dois histogramas, um para os pixels detectados como pele e outro para os pixels detectados como não-pele. Existem duas possibilidades quando estamos construindo a regra de decisão Bayesiana.

No primeiro caso, é suposto que a probabilidade de um pixel ser pele é equivalente à probabilidade de o mesmo pixel não ser pele ( $P(s) = P(\neg s)$ ). Isso corresponde à estimativa de máxima verossimilhança<sup>(1)</sup>. Para este caso, se a taxa de detecção encontrada no resultado da equação 5.2 for maior que o valor 1, o pixel é classificado como pele. Caso contrário, não.

$$\frac{P(s|\bar{x})}{P(\neg s|\bar{x})} = \frac{P(\bar{x}|s)}{P(\bar{x}|\neg s)} \quad (5.2)$$

No segundo caso nos deparamos com valores (possivelmente) diferentes de probabilidades de pele e não-pele para o mesmo pixel ( $P(s) \neq P(\neg s)$ ), pois os valores de probabilidade são previamente estimados do conjunto de

<sup>1</sup>[http://www.cnr.colostate.edu/class\\_info/fw663/MLEstimation.PDF](http://www.cnr.colostate.edu/class_info/fw663/MLEstimation.PDF)

treinamento. Essa propriedade corresponde à estimativa de máximo *a posteriori*<sup>(2)</sup>. Para este caso, se a taxa de detecção encontrada no resultado da equação 5.3 for maior que o valor 1, o pixel é classificado como pele. Se a taxa não for maior do que 1 o pixel é classificado como não-pele.

$$\frac{P(s|\bar{x})}{P(\neg s|\bar{x})} = \frac{P(\bar{x}|s)P(s)}{P(\bar{x}|\neg s)P(\neg s)} \quad (5.3)$$

Para a obtenção dos valores das probabilidades das equações 5.3 e 5.2, pode ser utilizada uma distribuição normal Gaussiana. Um modelo Gaussiano será apresentado na seção 5.3.

## 5.3 Modelos Gaussianos

Modelos Gaussianos para detecção de pele foram analisados em [17] na tentativa de determinar qual teria o melhor desempenho sobre nove espaços crominantes. Nesta seção explicaremos rapidamente o modelo Gaussiano simples (distribuição normal).

### 5.3.1 Distribuição Normal

Para o modelo Gaussiano simples, foi definida em [17] uma função de probabilidade como a equação 5.4 abaixo:

$$P[\bar{x}(i, j)/W_s] = (2\pi)^{-1} |\Sigma_s|^{-1/2} \exp[\lambda_s^s(i, j)/2] \quad (5.4)$$

Explanando-se a fórmula 5.4, temos que o vetor  $\bar{x}(i, j)$  é igual aos valores crominantes<sup>3</sup> para o pixel com a posição  $(i, j)$ . A classe que representa a pele é chamada de  $W_s$ .  $\Sigma_s$  é a matriz de covariância para a crominância da pele.  $\lambda_s(i, j)$  é a distância de Mahalanobis do vetor  $\bar{x}(i, j)$  até o vetor da média  $\mu_s = [\mu_{x_s} | \mu_{y_s}]^T$  obtida da crominância da pele. A fórmula para a distância de Mahalanobis está apresentada na equação 5.5.

$$[\lambda_s(i, j)]^2 = [\bar{x}(i, j) - \mu_s]^T \Sigma_s^{-1} [\bar{x}(i, j) - \mu_s] \quad (5.5)$$

A equação 5.5 define superfícies elípticas no espaço crominante de escala  $\lambda_s(i, j)$ , centralizado na  $\mu_s$  e onde os principais eixos são determinados por  $\Sigma_s$ .

<sup>2</sup><http://cnx.rice.edu/content/m11268/latest/>

<sup>3</sup>Adotaremos os nomes de (*Cb*, *Cr*) aos valores crominantes para facilitar o entendimento do método.

Os valores de  $\lambda_s(i, j)$  para um pixel com as coordenadas  $(i, j)$  determinam a probabilidade de um pixel pertencer à classe  $W_s$  (classe que representa a pele, ver equação 5.4)). Quanto maior for o valor de  $\lambda_s(i, j)$ , menor é a probabilidade de um pixel  $(i, j)$  pertencer à classe  $W_s$ .

Ambas  $\mu_s$  e  $\Sigma_s$  são estimadas utilizando-se um conjunto de imagens de entrada (em [17] foram utilizadas 110 amostras de pele gravadas com uma única câmera montada em um computador SGI). Após determinados ambos  $\mu_s$  e  $\Sigma_s$ ,  $[\lambda_s(i, j)]^2$  é calculada para cada pixel sobre todas as amostras de imagem e sobre 5 extensas regiões que não contém pixels-pele. O resultado é comparado ao parâmetro  $\lambda_{s,t}^2 > 0$  para limiarização. Para a classificação nas classes pele e não-pele, um valor padrão de limiarização é obtido quando a proporção de verdadeiros-positivos (VP - pixels que são pele e foram detectados como pele) se tornar igual à proporção de verdadeiros-negativos (VN - pixels que não são pele e foram detectados como pele) [32]. Ou, equivalentemente, quando a proporção de falsos-negativos (FN - pixels que não são pele e não foram detectados como pele) for igual à proporção de falsos-positivos (FP - pixels que não são pele e foram detectados como pele). Para o entendimento dos FN e dos FP, basta observar que  $VP + FN = 1$  e  $TN + FP = 1$ . Sobre pequenas variações de luz, o uso de somente uma câmera não exige um limiar adaptativo (como realizado em [32]).

Finalmente, a segmentação da cor da pele é realizada calculando-se  $[\lambda_s(i, j)]^2$  para cada pixel das imagens de teste e comparando o valor resultante com o limiar padrão  $\lambda_{s,t}^2$ .

O pixel  $(i, j)$  é associado ao valor 1 se  $\lambda_s \leq \lambda_{s,t}$  e é associado ao valor 0 se  $\lambda_s > \lambda_{s,t}$ . O resultado da limiarização é uma imagem binária que contém somente os pixels-pele.

Abaixo está apresentada uma tabela de resultados percentuais para alguns espaços de cores testados em [17].

	Distribuição Normal	Misturas Gaussianas
Espaço de Cores	Taxa de acerto	Taxa de acerto
<i>TSL</i>	83.32	82.87
<i>r - g</i>	81.35	80.38
<i>HSV</i>	76.43	83.46
<i>YIQ</i>	67.59	59.57
<i>YES</i>	60.69	56.53

Tabela 5.3: Resultados para os espaços de cores crominantes

## 5.4 Desempenho de alguns métodos

Para uma avaliação justa do desempenho de diferentes métodos de modelagem da cor da pele, é preferível utilizar condições idênticas de teste. Infelizmente, muitos métodos de detecção de pele fornecem resultados nos próprios bancos de dados, que geralmente não estão disponíveis. O banco de dados de imagens de teste e treinamento mais famoso para detecção de pele é o banco de dados Compaq<sup>4</sup>[18].

A tabela 5.4 apresenta os melhores resultados de diferentes métodos aplicados sobre o conjunto de imagens Compaq, relatados pelos autores.

Método	Autor	PV	PF
Bayes SPM no <i>RGB</i>	[7]	93.4%	19.8%
Limiar de um eixo no <i>YIQ</i>	[7]	94.7%	30.2%
Modelos de Entropia Máxima	[8]	80%	8%
SOM em <i>TS</i>	[12]	78%	32%
Bayes SPM no <i>RGB</i>	[18]	80%-90%	8.5%-14.2%
Misturas Gaussianas	[18]	80%-90%	9.5%-15.5%
Modelo Elíptico Limitado	[22]	90%	20.9%
Distribuição Normal no <i>CbCr</i>	[22]	90%	33.3%
Misturas Gaussianas no <i>IQ</i>	[22]	90%	30.0%

Tabela 5.4: Desempenho de detectores de pele

É possível observar na tabela 5.4 taxas positivas verdadeiras (PV - pixels pele que foram detectados corretamente) e taxas positivas falsas (PF - pixels não-pele que foram detectados erroneamente) para alguns diferentes métodos. Os métodos citados utilizam diferentes separações no banco de dados, entre fases de testes e treinamento. Além disso, os métodos utilizam diferentes estratégias de aprendizado, baseado nas abordagens de cada autor. Apesar de todas as diferenças, é possível ter uma idéia geral sobre o desempenho de cada método

<sup>4</sup>Possui quase 20 mil imagens. Cada imagem colorida possui uma equivalente binarizada, com seus pixels nomeados

# Capítulo 6

## Ferramentas de Apoio e Experimentos

Neste capítulo estão sendo apresentadas algumas ferramentas de apoio para a aprendizagem dos principais conceitos para as técnicas de segmentação de pele, bem como os experimentos realizados sobre as mesmas.

### 6.1 ImageJ

Nesta seção será apresentado uma introdução básica a respeito do programa ImageJ e de alguns plugins interessantes para auxílio na segmentação da pele.

#### 6.1.1 Descrição

ImageJ é um programa para processamento de imagens de código aberto implementado na linguagem Java <sup>1</sup>. O software foi inspirado no programa para Macintosh NIH Image <sup>2</sup>. O ImageJ pode ser executado de maneira online através de um *applet* java ou como uma aplicação local em qualquer computador que possua a máquina virtual Java 1.1 ou posterior.

Com o ImageJ é possível analisar, editar, processar, salvar e imprimir imagens 8-bit, 16-bit e 32-bit. Ele consegue rodar diversos tipos de imagens, incluindo TIFF, GIF, JPEG, BMP, DICOM, FITS e “RAW” ao mesmo tempo.

É possível calcular a área e as estatísticas dos valores de seleções definidas pelo usuário. É possível medir as distâncias entre diversos pontos e ângulos. É possível criar histogramas dos espaços de cores das imagens. Ele também

---

<sup>1</sup><http://java.sun.com>

<sup>2</sup><http://rsb.info.nih.gov/nihimage/Default.html>

suporta funções-padrão para processamento de imagens como manipulações de contraste, detecções de borda, filtros diversos e recortes de imagens.

O programa realiza transformações geométricas como alterações de escala, rotação e translação. As imagens podem ser aproximadas até uma escala de 32:1 e afastadas até 1:32. Todas funções de análise e processamento estão disponíveis em qualquer fator de ampliação. O programa suporta qualquer número de janelas (imagens) simultaneamente, limitado somente pela memória disponível.

A calibração espacial está disponível para fornecer medidas em unidades como milímetros. Densidade ou calibração em tons de cinza também está disponível.

O ImageJ foi implementado com código aberto e proporciona escalabilidade através de plugins escritos em Java. Aquisições personalizadas, análises e plugins de processamento podem ser desenvolvidos usando o editor do ImageJ e um compilador Java. Através dos plugins escritos pelos usuários é possível resolver quase todos os problemas de processamento ou análise de imagens.

## 6.1.2 Plugins

### *Skin Extractor*

O plugin *Skin Extractor* foi desenvolvido por Alain Lebet <sup>3</sup>, professor de Engenharia de Computação na universidade Lycee Diderot. Este plugin é um detector de pele para imagens *YCbCr* baseado em limiares crominantes.

O plugin realiza uma transformação de espaços de cores de *RGB* para *YCbCr* para cada imagem e, após a classificação, abre uma janela contendo o aglomerado de componentes crominantes em um plano cartesiano *Cb* x *Cr*.

O plugin realiza comparações simples dos valores de limiares pedidos com os valores de crominância existentes na imagem. Os valores sugeridos na inicialização do plugin variam entre [95, 140] para *Cb* e [140; 165] para *Cr*. Esses valores foram baseados nos valores de [14], porém foram adaptados de acordo com a base de dados utilizada por Alain Lebet. Em [14] os valores variam entre [90; 130] para *Cb* e [130; 180] para *Cr*. Em [9] também são citados valores diferentes para a crominancia: variam entre [77; 127] para *Cb* e [133; 173] para *Cr*. Vale lembrar que esses valores foram definidos para cada conjunto de treinamento testado nesses artigos e podem variar de conjunto de imagens para conjunto de imagens.

O *SkinExtractor* é um plugin simples de se instalar e executar. Para iniciar o seu funcionamento é só compilar e rodar com o menu do ImageJ

---

<sup>3</sup><http://frosch74.free.fr/index.html>

*Compile/Run.*

Realizamos testes em imagens obtidas de diversas maneiras. Existem três níveis de classificação para as imagens. As imagens de nível um são chamadas de *Hands.Easy.Back*. São quatro imagens de mãos de diferentes cores posicionadas sobre fundos simples. Para as imagens de nível dois adotamos o nome de *Hands.Complex.Back*. São oito imagens de uma mesma mão posicionadas sobre fundos complexos. As imagens de nível três são chamadas de *Skin.Complex.Back*. São 15 imagens com pessoas posicionadas sobre fundos complexos. Todas as imagens podem ser encontradas em <http://www.glug.ucdb.br/~roger/imagens.testes.zip> e estão sub-divididas com essa nomenclatura citada.

Para uma comparação entre os dois artigos e os valores-padrão do programa, realizamos testes com os valores citados em cada um, totalizando 81 testes. Para facilitar a explicação e os resultados utilizaremos alguns padrões:

- Para o conjunto de valores-padrão do programa (valores que o autor do programa definiu como ótimos para sua base dados, citados acima) adotaremos a nomenclatura de  $A$ ;
- Para o conjunto de valores definidos em [14], adotaremos a nomenclatura de  $B$ ;
- E para o conjunto de valores definidos em [9], adotaremos a nomenclatura de  $C$ .

Abaixo apresentamos 3 conjuntos de imagens resultantes de um conjunto de imagens original *Hands.Easy.Back* após passar pelo detector de pele *SkinExtractor*, de Alain Lebet. O conjunto de imagens inicial mostrado na figura 6.1 foi testado com os três conjuntos de valores ( $A$ ,  $B$  e  $C$ ).



Figura 6.1: *Hands.Easy.Back* originais de entrada

Podemos observar na figura 6.2 que o conjunto de valores de  $A$  não detecta tão bem a cor vermelha. Alguns aglomerados de pixels onde havia maior concentração da cor vermelha não foram detectados nas imagens aplicadas. Porém, os pixels não-pele foram identificados corretamente, e foram excluídos das imagens resultantes.



Figura 6.2: *Hands.Easy.Back* com  $C_b$  e  $C_r$  iguais a  $A$

No conjunto de valores de  $B$  ocorre o contrário. Os aglomerados de cor avermelhada são detectados corretamente enquanto que alguns pixels que não são pele são detectados como pele, principalmente os cinzas mais claros. É possível comprovar isso através da análise da figura 6.3.



Figura 6.3: *Hands.Easy.Back* com  $C_b$  e  $C_r$  iguais a  $B$

Na figura 6.4 foi utilizado o conjunto de valores de  $C$ . Nessa detecção podemos perceber um equilíbrio estatístico muito bom em ambos os tipos de pixels (pele e não-pele). Alguns aglomerados avermelhados não foram detectados e algumas bordas (sombrias da mão) foram adicionadas como pele, mas, comparado aos outros valores ( $A$  e  $B$ ) nas figuras 6.2 e 6.3, respectivamente, foi o que apresentou maior equilíbrio.



Figura 6.4: *Hands.Easy.Back* com  $C_b$  e  $C_r$  iguais a  $C$

As figuras 6.5 representam o conjunto de figuras *Hands.Complex.Back* originais.



Figura 6.5: *Hands.Complex.Back* de entrada

As figuras 6.6 representam o conjunto de figuras *Hands.Complex.Back* depois de passadas pelo detector de pele de Alain Lebet com os valores de  $A$ .



Figura 6.6: *Hands.Complex.Back* testados com os valores de  $A$

Para o conjunto de valores  $A$  a detecção não foi boa. Das 6 imagens resultantes apresentadas neste trabalho (ver figura 6.6), 4 estão completamente desfocadas, uma foi parcialmente detectada e a outra foi completamente não-detectada. Por causa do fundo de cor laranja da imagem, o detector de pele de Alain Le Bret confundiu as cores e detectou somente o fundo como pele.

As figuras 6.7 representam o conjunto de figuras *Hands.Complex.Back* depois de passadas pelo detector de pele de Alain Le Bret com os valores de  $B$ .



Figura 6.7: *Hands.Complex.Back* testados com os valores de  $B$

Para o conjunto de valores de  $B$  obtivemos uma detecção um pouco mais razoável do que em  $A$ , porém ainda assim fraca. Muitos pixels-pele foram detectados corretamente, porém o problema com o fundo laranja ainda persistiu. Também obtivemos muitos pixels-não pele detectados erroneamente como pele e um problema similar ao do fundo laranja com um fundo cor de madeira.

As figuras 6.8 representam o conjunto de figuras *Hands.Complex.Back* depois de passadas pelo detector de pele de Alain Le Bret com os valores de  $C$ .



Figura 6.8: *Hands.Complex.Back* testados com os valores de  $C$

O problema não foi corrigido com os valores de  $C$ . Na verdade, os resultados obtidos com os testes em  $C$  são bastante similares aos obtidos com os valores em  $B$ . Muitos pixels não-pele foram detectados erroneamente e o problema da cor laranja e da cor madeira ainda persistiu.

Os testes realizados com as imagens *Skin.Complex.Back* não foram apresentados neste trabalho, porém os mesmos podem ser encontrados em <http://www.glug.ucdb.br/~roger/imagens.testes.zip>. Os resultados para *Skin.Complex.Back* foram semelhantes aos encontrados nas imagens com mãos em fundos complexos, tendo bastante margem de erro para os pixels-pele.

Através dos testes e da análise do código de Alain Lebet, percebemos que a luminância ( $Y$ , ver seção 4.2 e 4.1) é um componente de pouca importância para este plugin. Através dos componentes crominantes foi possível realizar uma detecção de pele razoável para imagens com fundos não muito complexos. O plugin apresentou boas características quanto à velocidade de detecção, qualidade na classificação dos pixels com fundos simples e praticidade.

## Histograma de Cores

Este plugin para ImageJ foi criado por Dimiter Prodanov <sup>4</sup> e tem por função mostrar graficamente o histograma de cores em sua forma bi-dimensional (intensidade x número de ocorrências; ver seção 2.2.2). Ele trabalha apenas com imagens *RGB* porém apresenta muito bem cada componente. O plugin também mostra a média e o desvio padrão de cada componente. Neste plugin realizamos testes com imagens que possuíam somente a cor da pele.

Para a análise do plugin com elementos-pele, utilizamos um conjunto de imagens <sup>5</sup>. Este conjunto de imagens foi criado pela junção dos pesquisadores da *Rutgers University* com os doutores do Departamento de Dermatologia, na *University of Medicine and Dentistry of New Jersey*. As imagens têm por objetivo investigar a aparência visual da superfície da pele, considerando peles normais e peles que podem estar sendo afetadas por alguma doença. As imagens que não são pele são de autoria própria. Todas as imagens podem ser encontradas em <http://www.glug.ucdb.br/~roger/imagens.testes.zip>.

Através da análise das imagens-pele é possível perceber uma grande quantidade da componente  $R$  quando comparada às outras componentes, como apresentado nas imagens 6.10,6.14,6.18,6.22 e 6.26. Também é possível perceber através dos histogramas que as imagens não-pele não apresentam muita quantidade de componente vermelha (6.12,6.16,6.20,6.24 e 6.28).

<sup>4</sup><http://rsb.info.nih.gov/ij/plugins/colorhistogram.html>

<sup>5</sup>Obtido em [http://www.caip.rutgers.edu/rutgers\\_texture/cvg/](http://www.caip.rutgers.edu/rutgers_texture/cvg/)

A tabela 6.1 apresenta através de números o que foi obtido nas imagens:



Figura 6.9: Amostra Pele I

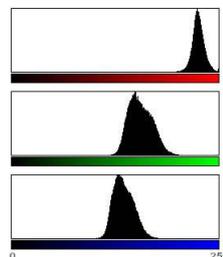


Figura 6.10: Pele Histograma I



Figura 6.11: Amostra N-Pele I

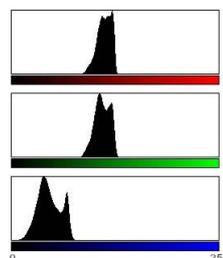


Figura 6.12: N-Pele Histograma I



Figura 6.13: Amostra Pele II

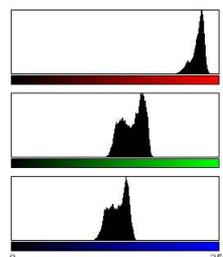


Figura 6.14: Pele Histograma II

Observando-se o componente  $R$  na tabela 6.1 é possível perceber a diferença de valor entre as imagens pele e as imagens não-pele. Os valores de  $R$  são sempre mais baixos para as imagens não-pele do que para as imagens pele.

Algumas análises a respeito dos fatos e das imagens sobre a componente  $R$ :

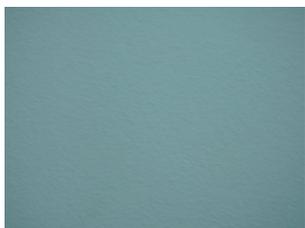


Figura 6.15: Amostra N-Pele II

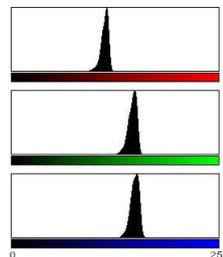


Figura 6.16: N-Pele Histograma II

	Imagens	Média			Desvio Padrão		
		<i>R</i>	<i>G</i>	<i>B</i>	<i>R</i>	<i>G</i>	<i>B</i>
Pele	6.9	230.30	159.03	138.12	8.08	14.88	12.89
	6.13	230.00	149.12	131.04	7.77	13.70	11.50
	6.17	150.67	101.03	88.94	9.99	7.80	6.51
	6.21	212.03	153.81	135.29	22.24	25.28	23.22
	6.25	160.45	112.50	99.49	26.57	17.57	15.71
Não Pele	6.11	114.31	111.74	46.12	8.62	9.38	14.27
	6.15	114.54	148.96	151.75	0.00	0.00	0.00
	6.19	81.64	78.83	44.91	12.27	13.44	13.48
	6.23	0.01	21.65	61.00	0.17	5.77	7.58
	6.27	115.15	114.72	116.79	33.89	34.26	31.56

Tabela 6.1: Resultados do Histograma



Figura 6.17: Amostra Pele III

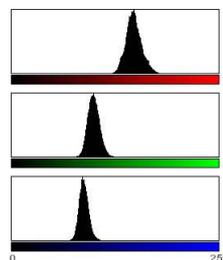


Figura 6.18: Pele Histograma III



Figura 6.19: Amostra  
N-Pele III

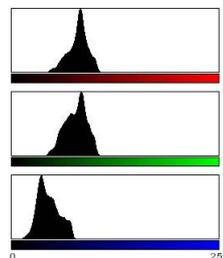


Figura 6.20: N-Pele  
Histograma III

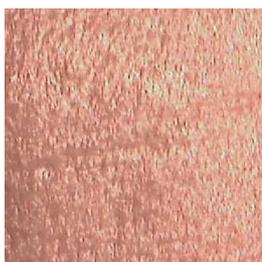


Figura 6.21: Amostra  
Pele IV

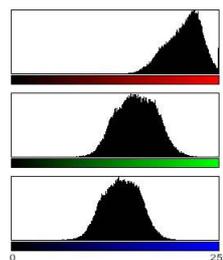


Figura 6.22: Pele His-  
tograma IV



Figura 6.23: Amostra  
N-Pele IV

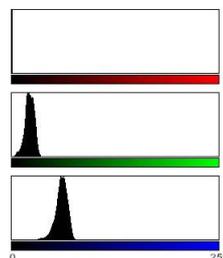


Figura 6.24: N-Pele  
Histograma IV

- Apesar de a imagem 6.13 aparentar ser mais avermelhada que a imagem 6.9, a imagem 6.9 possui uma média maior que a 6.13. Isso ocorre principalmente pela imagem 6.9 possuir um desvio padrão (espalhamento dos dados) maior que imagem 6.13. Enquanto os dados avermelhados da imagem 6.9 estão melhor distribuídos pela imagem, os dados avermelhados da 6.13 estão acumulados em um canto;
- As imagens 6.9 e 6.21 são as que mais se aproximam de um vermelho puro ( $R = 255$ ) e mesmo assim não parecem tão avermelhadas quanto a 6.13, graças ao desvio padrão;
- As imagens 6.21 e 6.25 possuem grandes extensões de “tons” de vermelho, porém muito bem distribuídos (altos valores de desvio padrão). Dessa maneira, a imagem não parece ser tão avermelhada;
- As imagens não pele possuem valores de  $R$  muito menores do que as imagens pele.

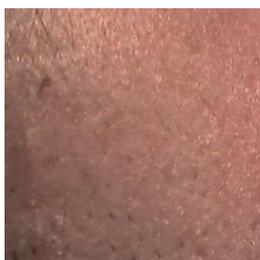


Figura 6.25: Amostra Pele V

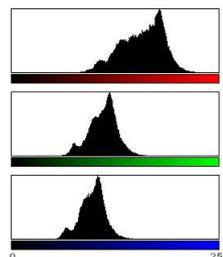


Figura 6.26: Pele Histograma V



Figura 6.27: Amostra N-Pele V

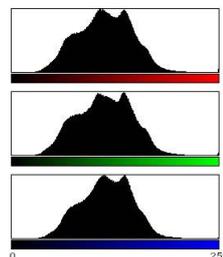


Figura 6.28: N-Pele Histograma V

Com isso, podemos concordar com [39] a respeito da componente  $R$  em um espaço de cores  $RGB$ . Um classificador implementado com a teoria mostrada

na seção 2.1 teria grande taxa de acerto nos pixels-pele, porém também traria à imagem resultante uma grande quantidade de pixels vermelhos de outros objetos. Uma alternativa para resolver esse problema seria inserir no código do classificador o cálculo do desvio padrão para eliminar (ao menos) os objetos que se encontrassem “longe” da pessoa que estiver sendo detectada.

Gostaríamos de lembrar que não foram realizados testes com peles negras neste classificador.

### 3D *Color Inspector*

Este plugin <sup>6</sup> criado por Kai Uwe Barthel <sup>7</sup> é muito interessante para o entendimento de alguns espaços de cores apresentados na seção 4.2. Sua instalação é muito simples, basta compilá-lo no ImageJ através do menu *Plugins/Compile and Run*.

O plugin funciona com todos os tipos de imagens aceitos pelo ImageJ e mostra a distribuição da cor da imagem corrente em um espaço de cores 3D muito interessante.

O programa possui visualizações em diversos espaços de cores (dos citados na seção 4.2: *RGB*, *YCbCr*, *HSV*) e diversas maneiras de analisar cada um. É possível modificar o número de cores das imagens para histogramas de 4x4x4, 5x5x5 ou 6x6x6, além do histograma original da imagem. É possível também modificar o histograma resultante, de maneira que as cores sejam apenas pontos pretos no espaço 3D branco. A perspectiva do espaço de cores 3D pode ser modificada, dando outras impressões a respeito de onde cada componente está localizada. A escala pode ser modificada, permitindo uma análise mais sucinta. Na janela do plugin é exibida uma cópia da imagem em tamanho ajustado. Ao clicar com o mouse sobre a imagem carregada são mostrados os valores dos componentes do espaço de cores e o número de vezes que aquela combinação aparece na imagem. É possível rotacionar o ângulo do espaço de cores que está sendo analisado, dando ótimas maneiras de ver como se comportam as cores em um espaço tridimensional.

A imagem 6.29 mostra a tela principal do *Color Inspector 3D* analisando uma imagem que contém pele no espaço *RGB*. As imagens 6.30 e 6.31 mostram o cubo *RGB* um pouco rotacionado.

As imagens 6.32, 6.33 e 6.34 (*RGB*, *YCbCr* e *HSV*, respectivamente) mostram três visualizações diferentes de histogramas em uma mesma imagem, cada uma referente a um espaço de cores já apresentados em na seção 4.2.

---

<sup>6</sup><http://rsb.info.nih.gov/ij/plugins/color-inspector.html>

<sup>7</sup>Outras publicações deste professor podem ser encontradas em <http://www.f4.fhtw-berlin.de/~barthel/paper/paper.htm>

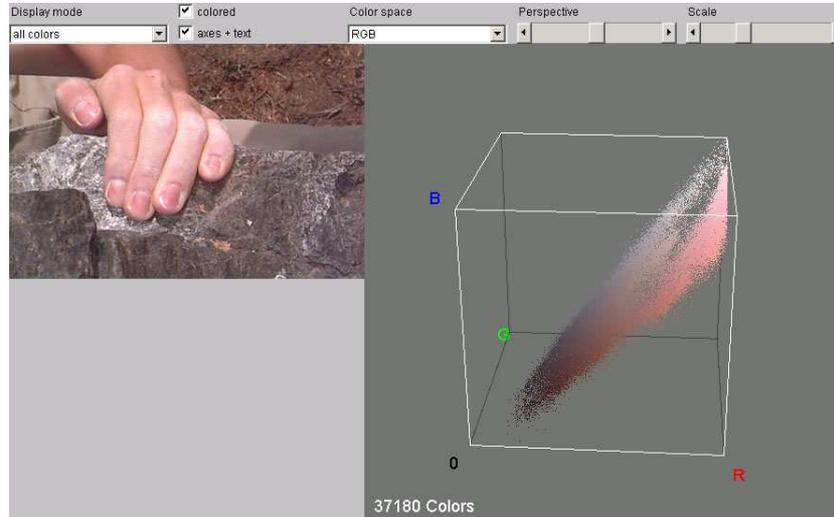


Figura 6.29: Amostra I no 3D Color Inspector

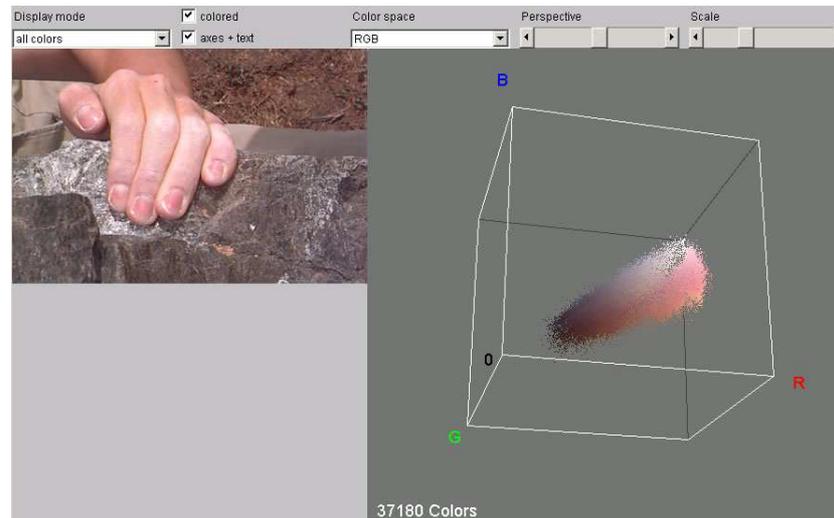


Figura 6.30: Amostra II no 3D Color Inspector

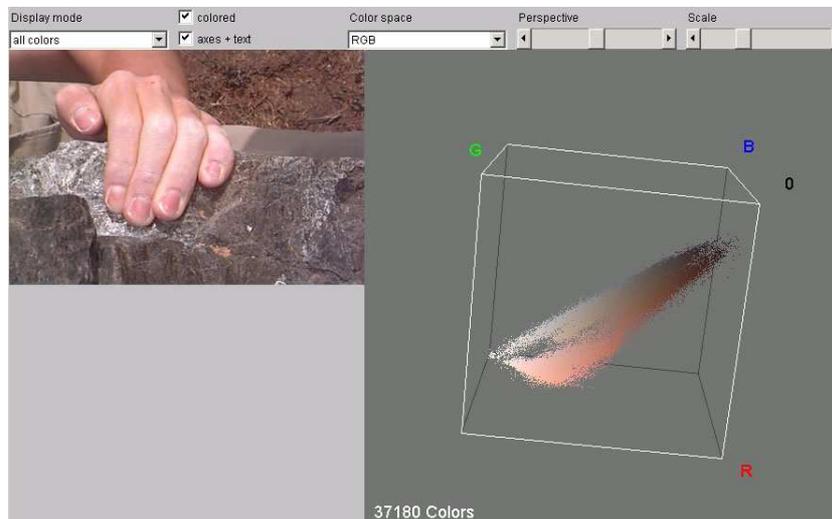


Figura 6.31: Amostra III no 3D Color Inspector

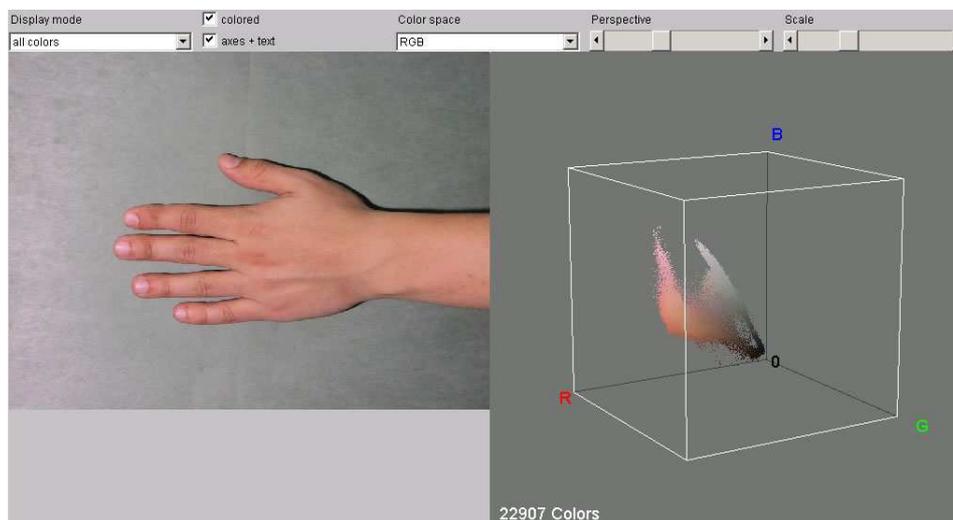


Figura 6.32: RGB

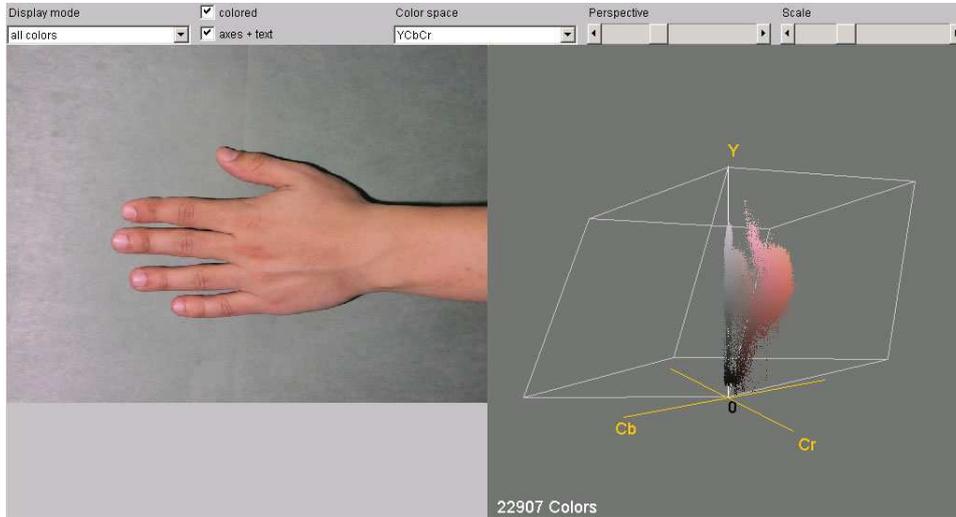


Figura 6.33: YCbCr

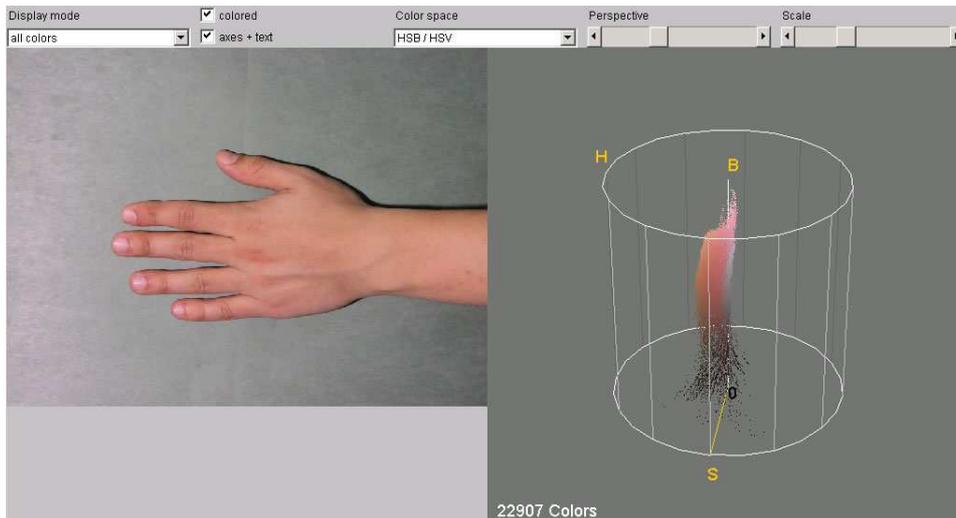


Figura 6.34: HSV

O plugin 3D *Color Inspector* é muito útil na parte gráfica. Permite realizar visualizações muito interessantes e esclarece muitas dúvidas a respeito dos espaços de cores. A representação da pele no espaço *RGB* também é dada de maneira correta, como podemos visualizar na figura 6.35, com resultados similares ao plugin da seção 6.1.2 (o aglomerado de cores representado na figura 6.35 se encontra muito mais próximo da componente *R* do que das componentes *G* e *B*). Porém, o cálculo prévio da média e do desvio padrão que o Histograma de Cores de Prodanov <sup>8</sup> realiza é muito importante para a detecção de pele, e é onde o 3D *Color Inspector* deixa algumas pendências.

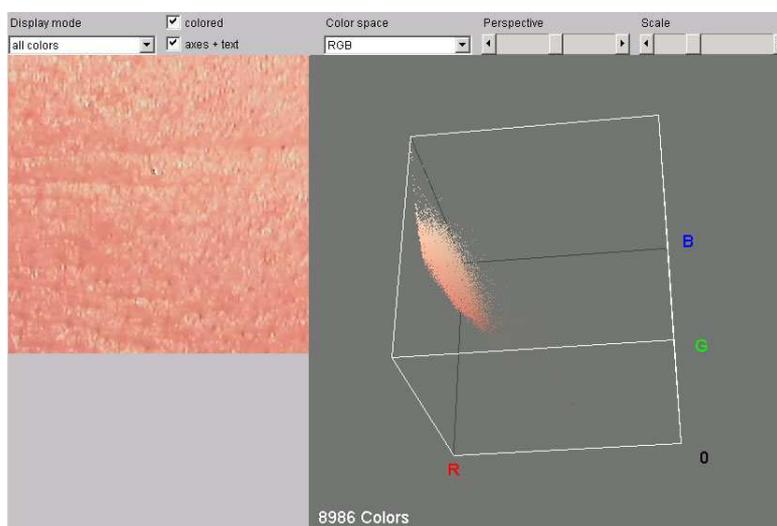


Figura 6.35: Pele

## 6.2 Colorspace

Colorspace <sup>9</sup> é um programa criado por Philippe Colantoni, pesquisador do *Laboratoire D'Informatique Graphique et D'Ingenierie de la Vision*. O objetivo do programa Colorspace é exibir informações das cores em diferentes espaços de cores. Ele pode realizar conversão de cores nos seguintes espaços de cores: *RGB*, *CMY*, *XYZ*, *xyY*,  $I_1I_2I_3$ , *UVW*, *LSLM*,  $L^*a^*b^*$ ,  $L^*u^*v^*$ , *LHC*, *HSV*, *HSI*, *HSVPolar*, *HSIPolar*, *LHS*, *YUV*, *YIQ*, *YCbCr*, *LMS*,  $LMS_{log}$ ,  $L_{log}M_{log}S_{log}$ ,  $L\alpha\beta$ ,  $\lambda SY$ ,  $\lambda SY_{Polar}$ . O programa Colorspace se encontra na versão 1.06.

Ele pode representar dois tipos de visualização para cada espaço de cores:

<sup>8</sup><http://rsb.info.nih.gov/ij/plugins/colorhistogram.html>

<sup>9</sup><http://www.couleur.org>

- Uma representação bidimensional (2D) das cores;
- Uma representação tridimensional (3D) das cores onde as coordenadas de cada cor (em um espaço de cores particular) são usadas para determinar a posição no espaço de cores 3D.

Utiliza GTK <sup>10</sup> como interface gráfica (GUI <sup>11</sup>). O método de visualização 3D utiliza OpenGL <sup>12</sup> para exibir as informações de cores 3D.

O programa é relativamente simples de ser utilizado. A primeira tela é basicamente um visualizador de imagens com alguns recursos avançados, como correção de cores e do valor gama.

O grande destaque deste programa são as visualizações dos espaços de cores 3D. Nesta ferramenta do programa é possível visualizar todos os espaços de cores baseando-se nas cores da imagem. Para efeitos de teste, apresentamos alguns destes espaços de cores nas figuras 6.36, 6.37, 6.38, 6.39. Estes espaços de cores são dependentes de dispositivos (ver seção 4.2.7), que dá a diferença na visualização humana de um para outro.



Figura 6.36: Imagem com o espaço de cores RGB

---

<sup>10</sup><http://www.gtk.org>

<sup>11</sup> *Graphical User Interface*

<sup>12</sup><http://www.opengl.org>



Figura 6.37: Imagem com o espaço de cores HSI



Figura 6.38: Imagem com o espaço de cores HSV



Figura 6.39: Imagem com o espaço de cores YCbCr

## 6.3 RGBCube

RGBCube <sup>13</sup> é outro programa criado por Philippe Colantoni que representa diversos espaços de cores de maneira tridimensional. Os espaços de cores que o programa pode representar são os seguintes: *RGB*, *CMY*, *XYZ*, *xyY*, *I<sub>1</sub>I<sub>2</sub>I<sub>3</sub>*, *UVW*, *LSLM*, *L\*a\*b\**, *L\*u\*v\**, *LHC*, *HSV*, *HSI*, *HSVPolar*, *HSIPolar*, *LHS*, *YUV*, *YIQ*, *YCbCr*, *LMS*, *LMS<sub>log</sub>*, *L<sub>log</sub>M<sub>log</sub>S<sub>log</sub>*, *Lαβ*, *λSY*, *λSY<sub>Polar</sub>*.

Ele não possui opção de visualização de imagens, porém a interface utilizada para representação das cores em cada espaço é bem intuitiva e facilita no entendimento de como as cores são tratadas.

Alguns exemplos de imagens geradas pelo RGBCube foram apresentadas em 4.2 com explicações a respeito.

---

<sup>13</sup><http://www.couleur.org>

# Capítulo 7

## Considerações Finais

### 7.1 Conclusão

O processamento das informações existentes em mãos e faces humanas é uma área de pesquisa com finalidades específicas, como interação homem-máquina, reconhecimento de mãos e faces, rastreamento de humanos, análise de expressões faciais e sistemas de reconhecimento de emoções humanas. Vários autores (ver referências) comprovam em seus trabalhos uma gama de utilidades ligadas a este tema. É uma das melhores maneiras de se realizar a detecção de humanos é através da pele, pois é invariante à escala, possui uma cor própria (diferente das cores existentes na natureza) e outras características que foram apresentadas no decorrer deste trabalho.

Neste trabalho foi apresentada algumas técnicas de detecção de pele já realizadas por alguns autores ([42],[8],[5]), bem como seus respectivos resultados e conclusões. Também foi apresentado alguns estudos básicos sobre histogramas, regra de bayes e espaços de cores.

Analizamos superficialmente o programa ImageJ e explicamos alguns plugins interessantes que facilitam na compreensão de muitos conceitos que são apenas citados na prática. Com isso esperamos auxiliar na escolha por espaços de cores, métodos de detecção de pele e técnicas para segmentação de pele.

### 7.2 Trabalhos Futuros

A sugestão mais importante para trabalhos futuros é aprofundar mais o estudo sobre a detecção da cor da pele. Muitos conceitos sobre as características da cor da pele e dos espaços de cores foram citados e aplicados neste trabalho, porém ainda faltam muitos exemplos de técnicas de visão computacional para

segmentação.

Técnicas baseadas em modelos estocásticos como predições de Markov de Segunda Ordem [20] e Modelos Ocultos de Markov [8] em conjunto com algoritmos de máxima estimativa [30],[20] precisam de estudos aprofundados para facilitar o entendimento e a escolha. Algoritmos baseados em Redes Bayesianas [42],[47],[24],[10], Distribuições Normais [40] e Misturas Gaussianas [42],[45],[6],[17],[45],[25] também são técnicas muito utilizadas para detecção e segmentação e necessitam de estudos comparativos mais abrangentes.

As principais diferenças entre detecção da pele de maneira paramétrica e não-paramétrica [42], modelos baseados em entropia máxima [8] e modelos baseados em Mapas Auto Organizáveis [42],[12],[19] também são tópicos importantes e necessitam de uma comparação para auxílio na escolha.

## Referências Bibliográficas

- [1] Saber A. and Tekalp A. M. Frontal-view face detection and facial feature extraction using color, shape and symmetry based cost functions. *Pattern Recognition Letters*, 19(8):669–680, 1998.
- [2] L. Torres A. Albiol and E. J. Delp. Optimum color spaces for skin detection. *IEEE International Conference on Image Processing, Thessaloniki, Greece*, 2001.
- [3] L. Torres Alberto Albiol and E. J. Delp. A simple and efficient face detection algorithm for video database applications. In *Proceedings of the International Conference on Image Processing*, volume 2, pages 239–242, Vancouver, Canada, 2000.
- [4] D.R. Bellhouse. On some recently discovered manuscripts of thomas bayes. *Historia Mathematica*, 29:383–394, 2002.
- [5] Boaz J. Super Benjamin D. Zarit and Francis K. H. Quek. Comparison of five color models in skin pixel classification. *ICCV'99 International Workshop on Recognition, Analysis and Tracking of Faces and Gestures in Real-Time Systems, RATFG-RTS'99*, pages 58–63, Setembro 1999.
- [6] C. A. Bouman. Cluster: An unsupervised algorithm for modelling gaussian mixtures. School of Electrical Engineering, Purdue University / <http://dynamo.ecn.purdue.edu/~bouman/software/cluster>, Setembro 2000.
- [7] J. Brand and J. Mason. A comparative assessment of three approaches to pixel-level human skin detection. In *Proc of the International Conference on Pattern Recognition*, 1:1056–1059, 2000.
- [8] Huicheng Zheng Bruno Jedynak and Mohamed Daoudi. Statistical models for skin detection. *Phys. Rev. Lett.*, 8, Junho 2003 Conference on Computer Vision and Pattern Recognition Workshop.

- [9] D. Chai and K.N. Ngan. Face segmentation using skin-color map in videophone applications. *IEEE Trans. Circuits Syst. Video Technol.*, 9:4, June 1999.
- [10] F.G Cozman. The javabayes system. *The ISBA Bulletin*, 7:16–21, June 2001.
- [11] M. Brandstein. C.Wang. Multi-source face tracking with audio and visual data. *In IEEE MMSP*, pages 169–174, 2003.
- [12] I. Craw D. Brown and J. Lewthwaite. A som based approach to skin detection with application in real time systems. *British Machine Vision Conference*, 2001.
- [13] Allan D. Jepson Gershon, Ron and John K. Tsotsos. Ambient illumination and the determination of material changes. *Journal of the Optical Society of America*, A(3-10):1700–1707, 1986.
- [14] Vincent Girondel, Laurent Bonnaud, and Alice Caplier. Hands detection and tracking for interactive multimedia applications. Laboratory of Image and Signal (LIS).
- [15] G. Gomez. On selecting colour components for skin detection. *In Proc. of the ICPR*, 2:961–964, 2000.
- [16] G. Gomez and E. Morales. Automatic feature construction and a simple rule induction algorithm for skin detection. *In Proc. of the ICML Workshop on Machine Learning in Computer Vision*, page 31–38, 2002.
- [17] H. Fukamachi J.-C. Terrillon, M. N. Shirazi and S. Akamatsu. Comparative performance of different skin chrominance models and chrominance spaces for the automatic detection of human faces in color images. *International Conference on Face and Gesture Recognition*, pages 54–61, 2000.
- [18] Michael J. Jones and James M. Rehg. Statistical color models with application to skin detection. *International Journal of Computer Vision*, 46(1):81–96, Janeiro 2002.
- [19] T. Kohonen. *Self-Organizing Maps*. Information Sciences, third edition edition, 2000.
- [20] Sigal L., Sclaroff S., and V. Athitsos. Skin color-based video segmentation under time-varying illumination. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26:862–877, 2004.

- [21] Sigal L., Sclaroff S., and Athitsos V. Estimation and prediction of evolving color distributions for skin segmentation under varying illumination. *CVPR*, pages 582–587, 2000.
- [22] J. Y. Lee and S. I. Yoo. An elliptical boundary model for skin color detection. *In Proc. of the 2002 International Conference on Imaging Science, Systems, and Technology*, 2002.
- [23] F. Marques and V. Vilaplana. A morphological approach for segmentation and tracking of human faces. *In International Conference on Pattern Recognition*, 1:5064–5068, 2000.
- [24] R.L. Marques and I. Dutra. Redes bayesianas: o que são, para que servem, algoritmos e exemplos de aplicações. Coppe Sistemas - UFRJ [ligeiro@cos.ufrj.br](mailto:ligeiro@cos.ufrj.br), [ines@cos.ufrj.br](mailto:ines@cos.ufrj.br).
- [25] S. Mckenna, S. Gong, and Y. Raja. Modelling facial colour and identity with gaussian mixtures. *Pattern Recognition*, 31(12):1883–1892, 1998.
- [26] D.A. Forsyth M.M. Fleck and C. Bregler. Finding naked people. *ECCV*, II:592–602, 1996.
- [27] A.P. Pentland N. Oliver and F. Berard. Lafter: Lips and face real time tracker. *CVPR97*, pages 123–129, 1997.
- [28] Stepan Obdrzalek. Pixel based skin detection. Internet, October, 26 2000.
- [29] S. L. Phung, A. Bouzerdoum, and D. Chai. A novel skin color model in ycbcr color space and its application to human face detection. *In IEEE International Conference on Image Processing (ICIPŠ2002)*, 1:289–292, 2002.
- [30] R. A. Redner and H. F. Walker. Mixture densities, maximum likelihood and the em algorithm. *SIAM Review*, 26:195–239, 1984.
- [31] M. Abdel-Mottaleb R.L. Hsu and A.K. Jain. Face detection in color images. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24(5):696–706, 2002.
- [32] E. Saber, A. M. Tekalp, R. Eschbach, and K. Knox. Automatic image annotation using adaptative color classification. *Graph. Models and Image Proc.*, 58(2):115–126, 1996.

- [33] D. Saxe and R. Foulds. Towards robust skin identification in video images. *in Proceedings of the Second International Conference on Automatic Face and Gesture Recognition, Killinton, Vermont, 1996.*
- [34] R. Schumeyer and K. Barner. A color-based classifier for region identification in video. in visual communications and image processing. *In Visual Communications and Image Processing, 3309:189–200, 1998.*
- [35] W. Skarbek and A. Koschan. Colour image segmentation – a survey. *Institute for Technical Informatics, 1994.*
- [36] K. Sobottka and I. Pitas. Face localization and facial feature extraction based on shape and color information. *in Proceedings of the IEEE International Conference on Image Processing, Lausanne, Switzerland, 1996.*
- [37] H. Stern and B. Efron. Adaptive color space switching for face tracking in multi-colored lighting environments. *In Proc. of the International Conference on Automatic Face and Gesture Recognition, page 249–255, 2002.*
- [38] M. Storring and E. G. H. Andersen. Skin colour detection under changing lighting condition. *In Araujo and J. Dias (ed.) 7th Symposium on Intelligent Robotics Systems, page 187–195, 1999.*
- [39] Moritz Störring. *Computer Vision and Human Skin Colour*. PhD thesis, Faculty of Engineering and Science, Aalborg University, Niels Jernes Vej 14, 9220 Aalborg, Denmark, 2004.
- [40] Dante A. C. Barone Tiberio S. Caetano, Silvia D. Olabarriaga. Performance evaluation of single and multiple-gaussian models for skin color modeling. *XV Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAPI'02), page 275, Outubro 2002.*
- [41] D. Tock and I. Craw. Tracking and measuring drivers' eyes. *Image and Vision Computing, 14:541–548, 1996.*
- [42] V. Vezhnevets, V. Sazonov, and A. Andreeva. A survey on pixel-based skin color detection techniques. *In Proc. Graphicon-2003, pages 85–92, 2003.*
- [43] H. Wang and S-F. Chang. A highly efficient system for automatic face region detection in mpeg video. *IEEE Transactions on circuits and system for video technology, 1997.*

- 
- [44] J. G. Wang and E. Sung. Frontal-view face detection and facial feature extraction using color and morphological operators. *Pattern recognition letters, vol. 20, no. 10*, 1999.
- [45] M. Yang and N. Ahuja. Gaussian mixture model for human skin color and its application in image and video databases. *In Proc. of the SPIE: Conf. on Storage and Retrieval for Image and Video Databases (SPIE 99)*, 3656:458–466, 1999.
- [46] M.-H. Yang and N. Ahuja. Detecting human faces in color images. *In International Conference on Image Processing (ICIP)*, 1:127–130, 1998.
- [47] N.L. Zhang and D. Poole. Exploiting causal independence in bayesian network inference. *Journal of Artificial Intelligence Research*, pages 301–328, 1996.