

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ  
DEPARTAMENTO ACADÊMICO DE INFORMÁTICA  
CURSO DE ENGENHARIA DE COMPUTAÇÃO**

**RENAN FELIPE AREND**

**DETECÇÃO E HISTÓRICO DE NEVOS EM IMAGENS DA PELE HUMANA**

**TRABALHO DE CONCLUSÃO DE CURSO**

**PATO BRANCO  
2015**

**RENAN FELIPE AREND**

**DETECÇÃO E HISTÓRICO DE NEVOS EM IMAGENS DA PELE HUMANA**

Trabalho de Conclusão de Curso de graduação, apresentado à disciplina de Trabalho de Conclusão de Curso 2, do Curso Superior de Engenharia de Computação, da Universidade Tecnológica Federal do Paraná, Campus Pato Branco, como requisito parcial para obtenção do título de bacharel.

Orientador: Prof.<sup>a</sup> Dr.<sup>a</sup> Gilda Aparecida de Assis

**PATO BRANCO  
2015**

A Deus, pois sem ele eu não teria forças para  
essa jornada.

Aos meus pais, que me proporcionaram a base  
para todas minhas conquistas.

A minha namorada que me auxiliou e me  
proporcionou segurança nos momentos mais  
difíceis.

A professora Gilda de Assim, minha  
orientadora, pois sem ela não seria possível  
este projeto

## RESUMO

AREND, Renan F.. Detecção e Histórico de Nevos em Imagens da Pele Humana. 2015. 77 f. Monografia (Trabalho de Conclusão de Curso) - Curso de Engenharia de Computação, Universidade Tecnológica Federal do Paraná, Campus Pato Branco. Pato Branco, 2015.

Este trabalho tem como objetivo a utilização da visão computacional, a partir do processamento digital de imagens, para o desenvolvimento de métodos computacionais de aquisição, detecção, categorização, e armazenamento de informações de pintas presentes na pele humana. O projeto tem como finalidade o desenvolvimento de um protótipo de aquisição controlada de imagens, para detecção, quantização e categorização de pintas, com propósito de contribuição com as tecnologias de pesquisa e prevenção contra o câncer de pele. O desenvolvimento do trabalho tem como escopo o estudo na detecção de pintas através do desenvolvimento de algoritmos de processamento de imagens, e obtenção de resultados que proporcionem informações para armazenamento em banco de dados de imagens, visando a manutenção de um histórico de mudanças decorrente na pele. Foram desenvolvidos testes através de algoritmos implementados na linguagem c++ para processamentos por cor e por escala de tons de cinza. No processamento por cor foi realizada a segmentação entre fundo, pele e pintas. O processamento por escala de tons de cinza foi feito através da utilização de filtros de identificação de bordas, como base de processamento para identificação das pintas. O vetor de características utilizado foi composto pelos seguintes dados: Número de regiões identificadas, nas quais para cada região foi extraído o número de pixels, a média e o desvio padrão de cada canal de cor e as coordenadas mínimas e máximas dos eixos X e Y. As características extraídas das imagens foram armazenadas em um banco de dados e as imagens foram armazenadas no banco de dados através do método de metadados.

**Palavras-chave:** Identificação de pintas, processamento de imagens, visão computacional, banco de dados de imagens, metadados.

## ABSTRACT

AREND, Renan F.. Detection and Historic of Pints in Human Skin's Images. 2015. 77 f. Monograph (Work Course Conclusion) - Computer Engineering Course, Federal Technological University of Paraná, Campus Pato Branco. Pato Branco, 2015

This work aims at the use of computer vision, as from digital image processing, for the computational development of detection methods, categorization, obtainment and storage of information of spots present on human skin. The project has goal to develop a prototype of controlled image acquisition, for detecting, quantization and categorization of gifts spots, with purpose in contributing to the researches and prevention of skin cancer Technologies. The development work is scoped to the study to detect spots through the development of image processing algorithms, subsequent to obtaining results that provide information for storage in the image database, aiming to develop a historic of changes arising in the skin. Were developed tests using C ++ algorithms for processing for color and grayscale. The processing was performed by color segmentation between background, skin and spots. The processing by grayscale tones was made by use of edge identification filter as a basis for processing to identify the spots. The feature vector used was comprised of following data: The number of identified regions, in which for each region was extracted the number of pixels, the average and standard deviation of each color channel and the minimum and maximum coordinates of the axes X and Y. The extracted features of images are stored in a database and the images were stored using the metadata method.

**Keywords:** pints identification, image processing, computer vision, image database, metadata.

## LISTA DE FIGURAS

Figura 1 - Matriz $N \times M$ que representa a função $f(x, y)$ .....	16
Figura 2 - Representação da vizinhança de $p$ .....	16
Figura 3 - Diagrama de Fluxo dos passos fundamentais no processamento de imagens .....	17
Figura 4 – Exemplos de histogramas de quatro tipos básicos de imagens .....	20
Figura 5 - Máscara 3x3 genérica.....	22
Figura 6 - Representação genérica níveis de cinza .....	23
Figura 7 - Máscara $G_x$ Figura 8 - Máscara $G_y$ .....	23
Figura 9 - Representação da máscara espacial da Equação 2.4-7 .....	24
Figura 11 – Exemplo de diagrama de entidades e relacionamentos de uma fábrica	27
Figura 12 - Fluxograma de procedimentos do projeto subsistema 1 .....	33
Figura 13 - Fluxograma do processo de Segmentação de Nevos da pele .....	34
Figura 14 – Obtenção de características de cada região .....	35
Figura 15 – Exemplos de captura de diferentes objetos alvo. ....	36
Figura 16 – Fluxograma do Subsistema 2.....	38
Figura 17 – Exemplo de Transformação espacial de imagem (Registro) .....	40
Figura 18 – Diagrama entidade e relacionamento do Banco de Dados .....	42
Figura 19 – Exemplo de armazenamento de Imagem.....	44
Figura 20 – Exemplo inserção de marcadores .....	45
Figura 21 - Exemplo de intensidade de cores Matrix Laboratory (MATLAB).....	46
Figura 22 – Resultado Segmentação e transformação para fundo (cor preta) dos Marcadores .....	48
Figura 23 - a) Imagem Original, b) Imagem Binarizada.....	48
Figura 24 – a) Imagem Original (Marcadores), b) Imagem Binarizada.....	49
Figura 25 – Resultado segmentação entre braço (objeto de interesse) e fundo. ....	50
Figura 26 – Representação de informações das Cores no MATLAB (R.G.B.) .....	51
Figura 27 – Representação de informações dos Tons de Cinza MATLAB ( <i>Index</i> ). ..	52
Figura 28 – a) Imagem de entrada. b) Imagem criada para identificação dos nevos	53
Figura 29 – a) Imagem de Entrada, b) Imagem resultado da Inversão de Cores.....	54
Figura 30 – Imagem de Cores Invertidas MATLAB. ....	55
Figura 31 – Imagem de Cores Invertidas em Tons de Cinza MATLAB. ....	55
Figura 32 - a) Imagem de Entrada, b) Imagem Resultado de Segmentação de Nevos. ....	56
Figura 33 - Exemplo de Imagem em tons de Cinza .....	57
Figura 34 – a) Imagem de entrada, b) Resultado processamento Sobel (cores invertidas).....	58
Figura 35 – a) Imagem entrada, b) Resultado processamento filtro de Canny(cores invertidas).....	60
Figura 36 - Resultado de identificação de regiões por cores.....	62
Figura 37 – Resultado de regiões com número de pixels maior que 12.....	62
Figura 38 – Resultado Tabela Pessoa .....	63
Figura 39 – Resultado Tabela Imagem .....	64
Figura 40 – Resultado Tabela Região .....	65
Figura 41 – Resultado Tabela Marcador .....	65
Figura 42 – Resultado Tabela Laudo .....	66
Figura 43 – Imagens de entrada para processamento.....	69
Figura 44 – Imagens pós-processamento. ....	69

## LISTA DE TABELAS

Tabela 1 – Exemplos de imagens capturadas conforme suas distâncias. ....	36
Tabela 2 – Resultados da extração de características Imagem 1. ....	69
Tabela 3 – Resultados da extração de características Imagem 2. ....	70

## LISTAGEM DE CÓDIGOS

Listagem 1 – Código de identificação dos Marcadores (azul e vermelho) .....	47
Listagem 2 – Algoritmo de Binarização .....	49
Listagem 3 – Algoritmo de Identificação de Nevos.....	53
Listagem 4 – Algoritmo de Inversão de Cores.....	54
Listagem 5 - Algoritmo de Identificação de Nevos de Imagem com Inversão de Cores .....	56
Listagem 6 – Algoritmo de processamento do Filtro de Sobel. ....	58
Listagem 7 – Algoritmo de Processamento do Filtro de Canny.....	59
Listagem 8 - Exemplo de Algoritmos de Rotulação.....	60
Listagem 9 – Código de rotulação por cores das regiões. ....	61
Listagem 10 – Código SQL de criação da tabela Pessoa .....	63
Listagem 11 – Código SQL de criação da tabela Imagem .....	63
Listagem 12 – Código SQL de criação da tabela Região.....	64
Listagem 13 – Código SQL de criação da tabela .....	65
Listagem 14 – Código SQL de criação da tabela Laudo .....	66
Listagem 15 – Código de criação e inicialização da variável MYSQL.....	67
Listagem 16 – Código de criação e inicialização da variável MYSQL.....	67
Listagem 17 – Função de retorno de informações do Banco de Dados.....	68



## LISTA DE SIGLAS

<i>BLOB</i>	<i>Binary Large Object</i>
<i>CIE</i>	<i>Comission Internationale de l'Éclairage</i>
<i>DICOM</i>	<i>Digital Imaging and Communications in Medicine</i>
<i>DISIMA</i>	<i>Distributed Image database Management system</i>
<i>INCA</i>	Instituto Nacional do Câncer José Alencar Gomes da Silva
<i>JPEG</i>	<i>Joint Photographic Experts Group</i>
<i>MATLAB</i>	<i>Matrix Laboratory</i>
<i>RGB</i>	<i>Red, Green, Blue</i>
<i>SGBD</i>	Sistema de Gerenciamento de Banco de Dados
<i>SQL</i>	<i>Structured Query Language</i>
<i>UTFPR</i>	Universidade Tecnológica Federal do Paraná

## Sumário

<b>1. INTRODUÇÃO</b> .....	<b>12</b>
1.1. CONSIDERAÇÕES INICIAIS.....	12
1.2. OBJETIVOS.....	13
<b>1.2.1. Objetivo Geral</b> .....	<b>13</b>
<b>1.2.2. Objetivos Específicos</b> .....	<b>13</b>
1.3. JUSTIFICATIVA.....	14
<b>2. REFERENCIAL TEÓRICO</b> .....	<b>15</b>
2.1. IMAGEM DIGITAL .....	15
<b>2.1.1. Amostragem e Quantização</b> .....	<b>15</b>
<b>2.1.2. Vizinhança</b> .....	<b>16</b>
<b>2.1.3. Conectividade</b> .....	<b>16</b>
<b>2.1.4. Processamento De Imagens</b> .....	<b>17</b>
2.1.4.1. Passos Fundamentais .....	17
<b>2.1.5. Realce</b> .....	<b>18</b>
2.1.5.1. Realce de Imagens por Filtragem de Domínio Espacial .....	18
2.1.5.2. Realce por Filtragem no Domínio da Frequência .....	19
<b>2.1.6. Histograma</b> .....	<b>19</b>
<b>2.1.7. Segmentação De Imagens</b> .....	<b>21</b>
<b>2.1.8. Detecção de Descontinuidade</b> .....	<b>21</b>
2.1.8.1. Detecção de Bordas .....	22
<b>2.1.9. Limiarização</b> .....	<b>24</b>
<b>2.1.10. Filtro de Inversão</b> .....	<b>25</b>
2.2 BANCO DE DADOS DE IMAGEM .....	25
<b>2.2.1. Banco De Dados</b> .....	<b>25</b>
<b>2.2.2. Entidades e Relacionamentos</b> .....	<b>26</b>
2.2.2.1. Propriedades .....	27
<b>2.2.3. Tipos de Dados</b> .....	<b>28</b>
<b>2.2.4. Métodos de Armazenamento de Imagens</b> .....	<b>28</b>
2.2.4.1. Tipo <i>BLOB</i> .....	28
2.2.4.2. Metadados.....	29
<b>3. MATERIAIS E MÉTODO</b> .....	<b>31</b>
3.1. MATERIAIS.....	31
3.2. MÉTODO .....	31
<b>4. PROJETO</b> .....	<b>33</b>
4.1. SUBSISTEMA 1 .....	33
4.2. SUBSISTEMA 2.....	38
4.3. BANCO DE DADOS .....	40
<b>4.3.1. Diagrama Entidade e Relacionamento</b> .....	<b>41</b>
<b>5. IMPLEMENTAÇÃO</b> .....	<b>44</b>
5.1. AQUISIÇÃO DE IMAGENS.....	44
5.2. PROCESSAMENTO POR COR .....	45
<b>5.2.1. Segmentação dos Marcadores Anatômicos</b> .....	<b>46</b>
<b>5.2.2. Segmentação do Objeto de Interesse do Fundo</b> .....	<b>48</b>
<b>5.2.3. Segmentação de Nevos da Pele</b> .....	<b>51</b>
<b>5.2.4. Segmentação de Nevos da Pele com Inversão de Cores</b> .....	<b>53</b>
5.3. PROCESSAMENTO EM TONS DE CINZA .....	56
<b>5.3.1. Filtro de Sobel</b> .....	<b>57</b>
<b>5.3.2. Filtro de Canny</b> .....	<b>59</b>

5.4. EXTRAÇÃO DE CARACTERÍSTICAS.....	60
5.5. BANCO DE DADOS .....	62
<b>5.5.1. Armazenamento.....</b>	<b>66</b>
<b>5.5.2. Recuperação .....</b>	<b>68</b>
5.6. RESULTADOS E DISCUSSÕES.....	69
<b>6. CONCLUSÃO .....</b>	<b>72</b>
<b>REFERÊNCIAS.....</b>	<b>74</b>

## 1. INTRODUÇÃO

Nesta seção são apresentados a problemática, objetivos e justificativas do trabalho proposto.

### 1.1. CONSIDERAÇÕES INICIAIS

O processamento de imagens está intrinsecamente presente em nossas vidas, tendo em vista que até mesmo em redes sociais são utilizadas técnicas de acentuação ou suavização de características da imagem. O método tem sido aplicado em áreas essenciais para a sociedade, como, por exemplo, a área da saúde. Na medicina, as tecnologias de processamento de imagem e visão computacional colocam à disposição dos profissionais um conjunto de ferramentas para o tratamento e manipulação de imagens em busca de um melhor diagnóstico, como no caso da melhoria das imagens capturadas por raio-X para facilitar a interpretação e diagnóstico a partir das imagens (SOARES, 2012).

Outro exemplo de uso de processamento de imagens e visão computacional na medicina é para a detecção do câncer. Esse é um problema na área da saúde para a humanidade e tem se tornado uma das principais preocupações globais atualmente, uma vez que políticas de prevenção e conscientização são fortes aliadas no combate ao câncer (BOTELHO, 2007, p. 3).

Segundo estimativas do Instituto Nacional do Câncer (INCA), o câncer de pele nos anos de 2014 e 2015 tem sido e será o mais incidente no Brasil (MINISTERIO DA SAÚDE, 2014). A partir desses dados, foram elaborados e implantados planos de conscientização contra o câncer de pele, instruindo a população acerca da importância de alguns cuidados básicos no dia-a-dia para a prevenção da doença. Segundo o Dr. Jefferson Luiz Gross, diretor do Núcleo de Pulmão e Tórax do Hospital A. C. Camargo, “atualmente, muitos tipos de câncer podem ser curados, desde que tratados em estágios iniciais, demonstrando-se a importância do diagnóstico precoce” (HOSPITAL A.C.CAMARGO, 2014, p. 3).

Conforme o Instituto Nacional de Câncer José Alencar Gomes da Silva afirma que, no Brasil, as estimativas para 2014 e 2015, apontam para a ocorrência de aproximadamente 576 mil casos novos de câncer, incluindo os casos de pele não

melanoma, reforçando a magnitude do problema do câncer no país. O câncer de pele do tipo não melanoma (182 mil casos novos) será o mais incidente na população brasileira, seguido pelos tumores de próstata (69 mil), mama feminina (57 mil), cólon e reto (33 mil), pulmão (27 mil), estômago (20 mil) e colo do útero (15 mil) (MINISTÉRIO DA SAÚDE, 2014, p. 26).

Há uma grande preocupação em se diagnosticar eventuais tumores malignos na pele, que podem começar sob a forma de nevos. Trata-se de pequenas manchas ou pequenas elevações na pele. São popularmente conhecidos como pintas, manchas ou verrugas. A maioria dos nevos surge em decorrência da exposição solar, podendo possuir um formato regular ou não (PEREIRA, 2012).

## 1.2. OBJETIVOS

A seguir apresentam-se os objetivos gerais e específicos do trabalho proposto.

### 1.2.1. Objetivo Geral

Realizar um estudo de caso de identificação computacional de nevos de pele em imagens digitais, por meio da aplicação de procedimentos de processamento de imagens e visão computacional. Esta identificação se relacionará na presença, ou não, do nevo na pele, buscando proporcionar meios de identificação de características de diferenças entre imagens, através da construção de um banco de dados, com objetivo de possibilitar a criação um histórico de aquisição das mesmas e rotulação de características dos nevos presentes nas imagens.

### 1.2.2. Objetivos Específicos

- Adquirir, pré-processar e filtrar as imagens.
- Segmentar, detectar e reconhecer os objetos de interesse.

- Extrair as características e informações das imagens relevantes para o processo de tomada de decisão.
- Manter um histórico das imagens.
- Realizar testes e validação utilizando um conjunto de imagens de membro superior.
- Documentar os métodos e resultados.

### 1.3. JUSTIFICATIVA

A área de processamento de imagens vem sendo utilizada em diversos domínios. Além de aplicações tecnológicas, técnicas de processamento de imagens digitais são utilizadas para resolver uma variedade de problemas. (GONZALEZ; WOODS, 2000, p. 2).

Atualmente, tecnologias computacionais para auxiliar na prevenção e detecção de iminentes cânceres de pele vêm sendo utilizadas, tendo como principal foco manter acompanhamento sobre variações da pele, sendo uma delas a análise e comparação de um histórico de imagens do mesmo indivíduo. Tal procedimento é realizado com base em uma coleção de fotografias. As imagens do corpo do paciente são armazenadas e analisadas por especialistas na área dermatológica, com a intenção de procurar nevos que possam, eventualmente, representar risco de se transformarem em câncer de pele. Pode-se também realizar comparações do histórico de imagens armazenadas em outros procedimentos já realizados, procurando avaliar o surgimento e o desaparecimento dos nevos.

A análise das imagens com a finalidade de se detectar qualquer anomalia nos nevos é um trabalho que exige do profissional certo tempo e destreza, sendo que, por fatores externos durante a captura das imagens (iluminação, posição das imagens, entre outros), esta análise pode não representar uma ferramenta norteadora da tomada de decisão do profissional de saúde.

Com o objetivo de proporcionar um melhoramento na interpretação de imagens digitais, o processamento de imagens pode ser aplicado, proporcionando um auxílio em informações visuais, tanto para o ser humano quanto para percepção automática de computadores.

## 2. REFERENCIAL TEÓRICO

Alguns dos conceitos básicos necessários serão brevemente explicados, formulando uma base teórica do trabalho proposto.

### 2.1. IMAGEM DIGITAL

A representação computacional de imagens digitais, pode ser referida à função bidimensional de intensidade da luz  $f(x, y)$ , onde  $x$  e  $y$  são as coordenadas espaciais e o valor da função  $f$  no ponto identificado por  $(x, y)$  representa o brilho (ou níveis de cinza) do ponto na imagem. Neste contexto, a imagem pode ser considerada como uma matriz cujos índices de linhas e de colunas identificam o ponto de imagem e o valor do ponto da matriz corresponde ao nível de cinza daquele ponto (GONZALEZ; WOODS, 2000, p.4-5).

As imagens possuem formas de serem representadas digitalmente através de padrões estabelecidos. O primeiro padrão de imagens foi definido em 1931 pelo comitê *Commission Internationale de l'Éclairage* (CIE), sendo chamado de CIE-RGB. O termo RGB define as cores primárias vermelha (*Red*), verde (*Green*) e azul (*Blue*), cujo seus valores são utilizados como forma de identificação da cor final (SCURI, 2002, p. 36,37).

#### 2.1.1. Amostragem e Quantização

Gonzalez (2000, p. 21) explica que para adequação de imagem para o processamento computacional a função  $f(x, y)$  precisa ser digitalizada tanto espacialmente (amostragem) quanto em amplitude (quantização).

A Figura 1 demonstra uma matriz  $N \times M$  de imagem em que cada elemento da função  $f(x, y)$  possui uma quantidade discreta.

$$f(x, y) \approx \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0, M-1) \\ f(1,0) & f(1,1) & \dots & f(1, M-1) \\ \vdots & \vdots & \ddots & \vdots \\ f(N-1,0) & f(N-1,1) & \dots & f(N-1, M-1) \end{bmatrix}$$

Figura 1 - Matriz  $N \times M$  que representa a função  $f(x, y)$   
 Fonte: Gonzalez e Woods (2000, p. 22).

### 2.1.2. Vizinhança

Um pixel  $p$  na imagem, coordenadas  $(x, y)$ , possui dois vizinhos *horizontais*, dois vizinhos *verticais* e quatro vizinhos nas suas *diagonais*, como mostra a Figura 2 (GONZALEZ; WOODS, 2000, p. 26).

$(x-1, y-1)$	$(x-1, y)$	$(x-1, y+1)$
$(x, y-1)$	$p$	$(x, y+1)$
$(x+1, y-1)$	$(x+1, y)$	$(x+1, y+1)$

Figura 2 - Representação da vizinhança de  $p$   
 Fonte: Gonzalez e Woods (2000, p. 26).

### 2.1.3. Conectividade

A conectividade é um método importante para a detecção de bordas e regiões em uma imagem digital. Para verificar se dois pixels estão conectados, é necessário determinar se eles, de alguma forma, são adjacentes, de forma que seus níveis de cinza ou canais de cor satisfaçam certos critérios de similaridade (GONZALEZ; WOODS, 2000, p.27, 28).



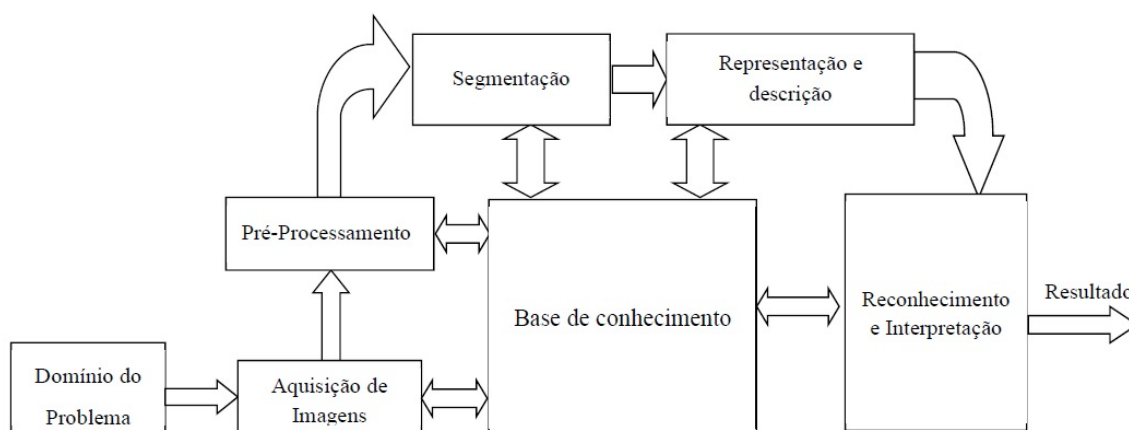
## 2.1.4. Processamento De Imagens

O processamento de imagens tem como principais objetivos a melhoria de informação visual para a interpretação humana e o processamento de dados para percepção automática de máquinas (GONZALEZ; WOODS, 2000, p. 1).

A representação, descrição e interpretação de elementos na imagem, quer tenha sido processada ou não, compreende uma área denominada Visão Computacional.

### 2.1.4.1. Passos Fundamentais

Para a execução de uma tarefa de processamento de imagens e visão computacional é necessário a utilização de passos fundamentais, como demonstra o diagrama de fluxo na Figura 3.



**Figura 3 - Diagrama de Fluxo dos passos fundamentais no processamento de imagens**  
 Fonte: Gonzalez e Woods (2000, p. 5)

O passo inicial do processo é a **aquisição de imagem**, que consiste em adquirir uma imagem digital; após, passa-se à etapa de **pré-processamento** da imagem adquirida no passo anterior, tendo como objetivo o melhoramento da imagem, aumentando as chances de sucesso do processo dos próximos passos.

O terceiro passo trata da **segmentação** na qual, em termos básicos, divide-se uma imagem obtida em partes ou objetos constituintes mais significativos, sendo considerada uma das tarefas mais difíceis no processamento de imagens digitais. O resultante do processo de segmentação é direcionado para a **representação e**

**descrição**, tendo como objetivo a representação dos dados como regiões completas ou fronteira. Essa representação é considerada apenas como parte da solução para transformar os dados iniciais em resultados adequados para o processamento computacional, resultando em informações quantitativas de interesse ou básicas para discriminação dos objetos.

O estágio de **reconhecimento e interpretação** tem como função o reconhecimento dos objetos e atribuição de um significado ao conjunto de objetos reconhecidos. Interligando-se a todos os processos está a **base de conhecimento**, sendo que o conhecimento sobre o domínio do problema é a base, controlando a interação entre os passos conforme o conhecimento do resultado esperado, podendo ser de importância mesmo o mais simplório até o mais complexo dos casos (GONZALEZ; WOODS, 2000, p. 5-7).

### 2.1.5. Realce

O realce de imagens tem como objetivo tornar a imagem mais apropriada para a aplicação do que a imagem original (GONZALEZ; WOODS, 2000, p. 115), acentuando as características importantes para o processo de decisão e atenuando ou eliminando as demais.

#### 2.1.5.1. Realce de Imagens por Filtragem de Domínio Espacial

Gonzalez e Woods descrevem o termo *Domínio Espacial* como agregado de pixels que compõem uma imagem, e consideram que os métodos no domínio espacial são procedimentos que operam diretamente sobre estes pixels (GONZALEZ; WOODS, 2000, p. 115).

O uso de Filtragem Espacial na área do processamento computacional de imagens tem como propósito o realce ou atenuação de características das imagens para proporcionar um melhor desempenho conforme os objetivos da análise. São utilizados diversos filtros espaciais como a filtragem passa-baixa, a filtragem por mediana, a filtragem passa-alta, os filtros por derivadas, entre outros (GONZALEZ, WOODS; 2000, p.134-142).

### 2.1.5.2. Realce por Filtragem no Domínio da Frequência

Os filtros do domínio da frequência entram como opção para solução de muitos problemas que não são facilmente tratáveis através de técnicas espaciais. O método é fundamentado pelo teorema da convolução. Seja  $f(x, y)$  uma imagem e um operador linear invariante com posição  $h(x, y)$ , a convolução desses dois fatores resulta em  $g(x, y)$  (equação 1) (GONZALEZ; WOODS, 2000, p. 117).

$$g(x, y) = h(x, y) * f(x, y).$$

1

Fonte: Gonzalez e Woods (2000, p. 117).

Partindo do teorema da convolução a relação no domínio da frequência é verificada conforme a equação 2

$$G(u, v) = H(u, v)F(u, v)$$

2

Fonte: Gonzalez e Woods (2000, p. 117).

em que  $H(u, v)$  é determinado como função de transferência óptica, o qual determina o cômputo em frequência da imagem desejada.

Alguns filtros que utilizam da frequência são: filtro *passa-baixa*, filtro *passa-baixa Butterworth*, filtro *passa-alta*, filtro *passa-alta Butterworth*, filtragem *Homomórfica* (GONZALEZ; WOODS, 2000, p. 142-154).

### 2.1.6. Histograma

Um histograma proporciona fundamentalmente uma estimativa da probabilidade de ocorrência dos níveis de cinza ou canais de cor em uma imagem, podendo ele, ser representado em uma função discreta

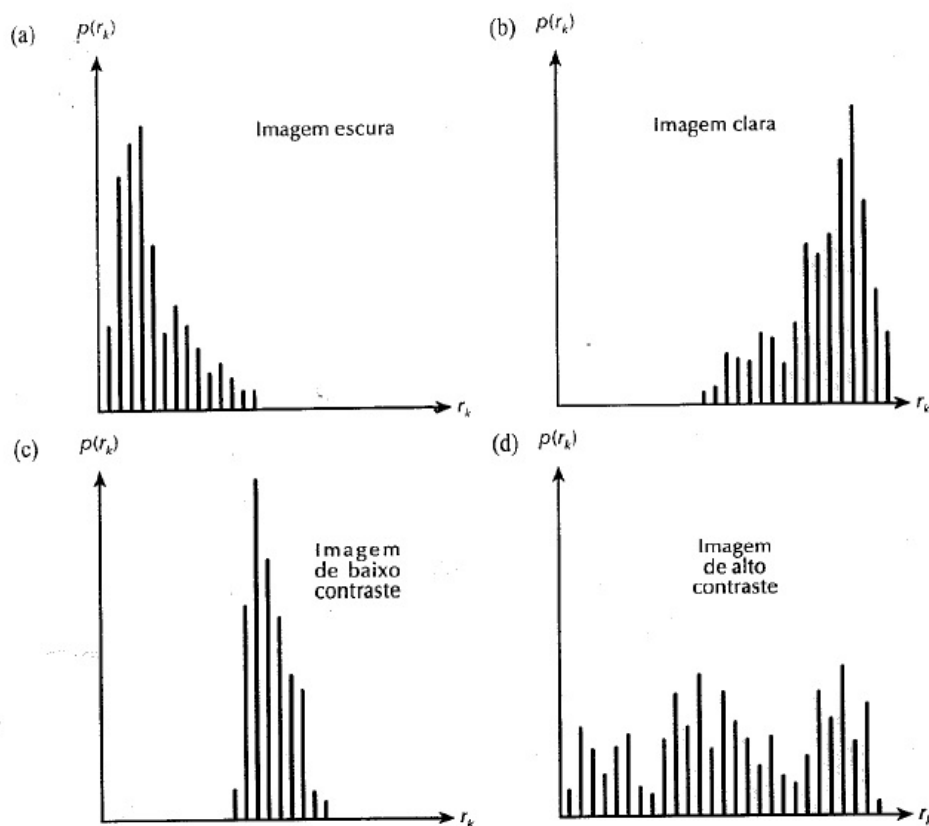
$$p(r_k) = \frac{n_k}{n}$$

3

Fonte: Gonzalez e Woods (2000, p. 122).

em que  $r_k$  é o  $k$ -ésimo nível de cinza,  $n_k$  o número de pixels na imagem com esse nível de cinza e o  $n$  o número total de níveis de cinza na imagem em que  $k$  varia conforme o intervalo de níveis de cinza da imagem  $[0, L - 1]$ . No caso de imagens coloridas utiliza-se um raciocínio análogo, com um histograma para cada canal de cor.

O resultado de um histograma pode ser representado por um gráfico da função  $p(r_k)$ , possuindo este, informações úteis para o realce do contraste de uma imagem como no exemplo das figuras 4(a), 4(b), 4(c) e 4(d) (GONZALEZ; WOODS, 2000, p 122-124).



**Figura 4 – Exemplos de histogramas de quatro tipos básicos de imagens**  
 Fonte: Gonzalez e Woods (2000, p. 124).

O histograma da Figura 4(a) representa uma imagem em que os níveis de cinza se concentram na extremidade escura (esquerda) de seu intervalo, correspondendo assim a imagem com características predominantemente escuras. O histograma 4(b) possui características de uma imagem com tendência a ser clara, cujos níveis de cinza são representados na extremidade clara (direita) de seu intervalo (GONZALEZ, WOODS, 2000, p. 123).

Pode-se analisar níveis de contrastes de imagens através de seu histograma, como os casos das figuras 4(c) e 4(d), sendo a 4(c) de baixo contraste por possuir uma escala estreita de níveis de cinza e a 4(d) de alto contraste, a qual possui um espalhamento significativo na escala de níveis de cinza (GONZALEZ, WOODS, 2000, p. 123,124).

### 2.1.7. Segmentação De Imagens

A extração de informações relevantes é de grande importância para a realização de análise de imagens, sendo que a segmentação é considerada, geralmente, como primeiro passo para a investigação das mesmas. A segmentação consiste em subdividir uma imagem nas suas partes ou objetos constituintes, cujo nível de subdivisão a ser realizado depende particularmente do problema (GONZALEZ; WOODS, 2000, p. 295).

### 2.1.8. Detecção de Descontinuidade

A detecção de descontinuidade no domínio espacial se baseia no procedimento de cálculo da soma dos produtos dos coeficientes pelos níveis de cinza contidos na região englobada por uma máscara (Figura 5), sendo a resposta desta máscara em qualquer ponto de uma imagem 3x3, representada por

$$R = w_1 z_1 + w_2 z_2 + \dots + w_9 z_9 \quad 4$$

$$R = \sum_{i=1}^9 (w_i z_i) \quad 5$$

Fonte: Gonzalez e Woods (2000, p. 296).

Na equação acima  $z_i$  é o nível de cinza do pixel associado comutado ao coeficiente da máscara  $w_i$  (GONZALEZ; WOODS, 2000, p. 296).

$w_1$	$w_2$	$w_3$
$w_4$	$w_5$	$w_6$
$w_7$	$w_8$	$w_9$

**Figura 5 - Máscara 3x3 genérica**  
**Fonte: Gonzalez e Woods (2000, p. 296).**

As formas principais de detecção de descontinuidade são por detecção de pontos, de linhas e de bordas.

#### 2.1.8.1. Detecção de Bordas

A abordagem considerada mais comum para a detecção de descontinuidades em níveis de cinza segundo Gonzalez e Woods (2000, p. 297) é a Detecção de Bordas. Uma borda é fundamentada como o limite entre duas regiões com níveis de cinza relativamente distintos, sendo avaliadas formas de determinação de descontinuidade dos níveis de cinza.

Utilizam-se métodos de primeira e segunda derivada para identificação de bordas, em que a magnitude da primeira derivada é utilizada na detecção da existência de uma borda, enquanto que o sinal obtido pela segunda derivada identifica o pixel como lado escuro ou claro da borda na imagem. (GONZALEZ, WOODS, 2000, p. 297)

Para a realização de detecção de bordas através da derivada de primeira ordem são utilizados operadores gradientes na imagem  $f(x, y)$ . A equação 4 mostra como na posição  $(x, y)$  é representado o vetor gradiente.

$$\nabla f = \begin{pmatrix} G_x \\ G_y \end{pmatrix} = \begin{pmatrix} \frac{d}{dx} f \\ \frac{d}{dy} f \end{pmatrix}$$

Diversas máscaras foram propostas na literatura como formas de aproximação do vetor gradiente. Um exemplo são os operadores de Sobel, no qual as derivadas são baseadas em máscaras do operador Sobel aplicadas aos níveis de cinza dos pixels vizinhos do pixel atual (Figura 6) (GONZALEZ, WOODS, 2000, p. 299).

$z_1$	$z_2$	$z_3$
$z_4$	$z_5$	$z_6$
$z_7$	$z_8$	$z_9$

**Figura 6 - Representação genérica níveis de cinza**  
**Fonte: Gonzalez e Woods (2000, p. 299).**

Sendo que os  $z$  são os níveis de cinza dos pixels em qualquer posição da imagem

-1	-2	-1
0	0	0
1	2	1

-1	0	1
-2	0	2
-1	0	1

**Figura 7 - Máscara Gx**                      **Figura 8 - Máscara Gy**  
**Fonte: Gonzalez e Woods (2000, p. 299).**

As Figuras 7 e 8 representam as máscaras de detecção de bordas dos operadores de Sobel, sendo uma para o cômputo horizontal (Gx) e outra para o vertical (Gy) (GONZALEZ, WOODS, 2000, p. 299).

O operador de Canny é utilizado também como detector de bordas. Seu algoritmo utiliza múltiplos passos para realizar a detecção da possibilidade de bordas presentes, seu filtro é de convolução de primeira ordem. O filtro de Canny realiza a suavização de ruídos e localização de bordas, através da combinação de um operador diferencial com um filtro Gaussiano.

Para as derivadas de segunda ordem pode ser utilizado o operador laplaciano de uma função bidimensional  $f(x, y)$  definida por

$$\nabla^2 f = \frac{d^2}{dx^2} f + \frac{d^2}{dy^2} f$$

7

Fonte: Gonzalez e Woods (2000, p. 300).

Assim como no gradiente, o laplaciano pode ser implementado digitalmente de diferentes formas, em uma região de 3 x 3 a forma mais frequentemente encontrada é

$$\nabla^2 f = 4z_5 - (z_2 + z_4 + z_6 + z_8)$$

8

Fonte: Gonzalez e Woods (2000, p. 301).

em que os  $z$  já foram definidos na Figura 6. Para a definição digital do laplaciano é necessário que o coeficiente associado com o pixel central seja positivo e os outros pixels externos sejam negativos, considerando que o laplaciano é uma derivada, a soma dos seus coeficientes precisa ser nula (Figura 9) (GONZALEZ, WOODS, 2000, p. 300-302).

0	-1	0
-1	4	-1
0	-1	0

Figura 9 - Representação da máscara espacial da Equação 2.4-7

Fonte: Gonzalez e Woods (2000, p. 301).

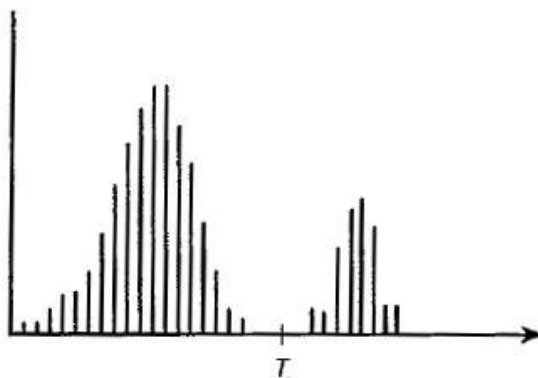
### 2.1.9. Limiarização

Uma das abordagens mais importantes para a segmentação de imagens é a Limiarização, que é a separação de objetos do fundo através da seleção de um valor  $T$  que separe dois ou mais grupos de níveis de cinza ou canais de cor da imagem. Cada ponto  $(x, y)$  da imagem tal que  $f(x, y) > T$  é denominado um ponto do objeto, caso



$f(x, y) < T$  o ponto é denominado de fundo, podendo  $T$  possuir mais de um valor conforme a quantidade desejada de separação (GONZALEZ; WOODS, 2000, p.316).

A Figura 10 apresenta um histograma de valores de cinza com o valor limiar representado por  $T$ .



**Figura 10 – Histograma de Limiarização**  
 Fonte: Gonzalez e Woods (2000, p. 316).

### 2.1.10. Filtro de Inversão

Um dos métodos auxiliares de processamento de imagens é o método de operação de inversão, seu efeito gerado por um filtro de inversão se dá através do processo de inversão dos valores de matiz, saturação e brilho de uma imagem digital (BOUDREAUX, 2012, p.1).

## 2.2 BANCO DE DADOS DE IMAGEM

São apresentados conceitos teóricos sobre banco de dados para a aplicação de um banco de dados de imagem digital.

### 2.2.1. Banco De Dados

Um sistema de Banco de Dados é considerado basicamente, como sua finalidade geral, armazenador de informações, que permite que os usuários busquem e atualizem essas informações quando solicitarem. Essas informações podem ser qualquer elemento que tenha algum significado ao usuário, sendo ele indivíduo ou organização, a que o sistema deva servir (DATE, 2004, p. 6).

Segundo Date, “um banco de dados é uma coleção de dados persistentes, usada pelos sistemas de aplicação de uma determinada empresa”. Ao referenciar a

palavra *empresa*, o autor indica como um termo genérico para qualquer indivíduo, organização, corporação ou empresa (DATE, 2004, p. 7).

O banco de dados permite ao usuário o armazenamento a recuperação a atualização e a remoção de dados, possibilitando uma melhor organização no armazenamento e recuperação (consulta) dos mesmos, uma vez que os bancos de dados computacionais são cada vez mais numerosos (KIOSKEA, 2014, p. 1,2).

Para melhorar o controle dos dados assim como dos usuários é utilizado um sistema de gestão de banco de dados (SGBD), o SGBD é um conjunto de aplicações de software que facilitam e permitem o gerenciamento do banco de dados (KIOSKEA, 2014, p. 2).

Um sistema de Banco de Dados proporciona o controle centralizado dos dados operacionais, diminuindo consideravelmente a redundância e a inconsistência dos arquivos. O controle centralizado de um Banco de Dados proporciona também a padronização dos dados, assim como mantem a integridade dos mesmos (BITTENCOURT, 2004, p. 4).

A integridade de um Banco de Dados é uma forma de assegurar que os dados sejam corretos, não permitindo que informações armazenadas sejam modificadas incorretamente. Existe em um Banco de Dados a Integridade Referencial e a Integridade Transacional (BITTENCOURT, 2004, p. 5).

A Integridade Referencial é quando os registros de relacionamento fazem referência às entidades existentes. Já a Integridade Transacional é quando as transações efetuadas devem ocorrer com segurança, ao completar ou não a transação, os dados devem se manter íntegros (BITTENCOURT, 2004, p. 5).

O Banco de Dados ao proporcionar um controle centralizado, também proporciona um compartilhamento dos dados, cujas novas aplicações possam ser implementadas sobre o mesmo e sem causar conflitos (BITTENCOURT, 2004, p. 4).

### **2.2.2. Entidades e Relacionamentos**

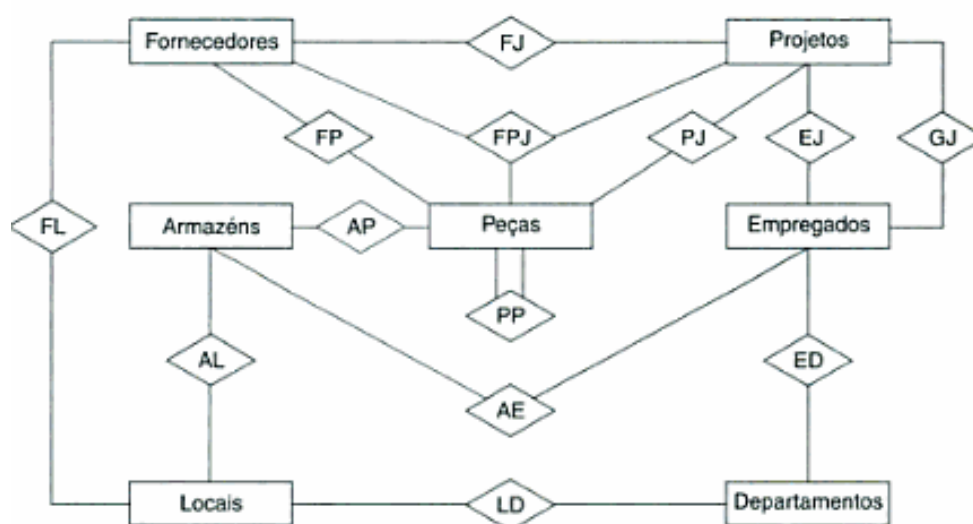
No projeto de um sistema de banco de dados é preciso registrar informações sobre o mesmo, como exemplo de uma fábrica, que precisa registrar as peças usadas, os fornecedores, os empregados e entre outros. Essas informações são consideradas,

em um banco de dados, as entidades básicas sobre as quais a fábrica precisa registrar suas informações (DATE, 2004, p. 11).

Segundo Rocha e Santanchè, “um banco de dados estruturado de acordo com o modelo relacional corresponde a uma coleção de relações” (ROCHA; SANTANCHÈ, 2012, p. 5).

Como forma de interligação das entidades criadas, são utilizados os relacionamentos. A representação desses relacionamentos é feita por losangos e linhas de conexão.

Os relacionamentos, no geral, fazem parte dos bancos de dados tanto quanto as entidades, devendo ser representados no banco de dados relacional, assim como as entidades básicas conforme mostra o exemplo da Figura 11 (DATE, 2004, p. 11,12).



**Figura 10 – Exemplo de diagrama de entidades e relacionamentos de uma fábrica**  
Fonte: Date (2004, p.11).

#### 2.2.2.1. Propriedades

Em um banco de dados é possível registrar informações a respeito das entidades e relacionamentos citados anteriormente, essas informações são denominadas no sistema como propriedades. Como exemplo, as peças da fábrica possuem, cada uma, um peso. No geral, propriedades podem ser simples ou complexas, tendo um tipo simples ou estruturado (DATE, 2004, p. 13).

### 2.2.3. Tipos de Dados

Um tipo é conceitualmente um conjunto de valores, como exemplo de alguns já definidos, o tipo *INTEGER*, que considera o conjunto de todos os inteiros, o tipo *VARCHAR*, o conjunto de todas as *strings* de caracteres (DATE, 2004, p. 95).

Para este trabalho foram utilizados alguns tipos de dados para a criação das entidades e relacionamentos do banco de dados, entre eles o tipo *Binary Large Object (BLOB)*. O tipo *BLOB* pode ser usado como uma forma de armazenamento de imagens em um banco *Structured Query Language (SQL)* (DATE, 2004, p. 110, 111). Outro método possível é armazenar as imagens em um outro local (diretório) fora do banco e somente referencia-las através de seu caminho (metadados).

### 2.2.4. Métodos de Armazenamento de Imagens

Um objeto armazenado no banco de dados de multimídia deve ser descrito de modo que possa ser facilmente localizado e recuperado. É utilizada a indexação como determinação de quais dados devem ser referenciados para descrever os objetos multimídias e permitir a pesquisa e recuperação dos mesmos (AIRES, 2005, p. 28).

Os métodos citados anteriormente para armazenamento de imagens possuem algumas vantagens e desvantagens, cabendo neste trabalho a escolha de qual método é o mais adequado para o projeto proposto.

#### 2.2.4.1. Tipo *BLOB*

Como citado anteriormente existem uma série de tipos de campos ou dados definidos em um banco de dados. A principal diferença entre o tipo *BLOB* e os outros tipos é a quantidade de informação que se torna possível de armazenar. Um *blob* é basicamente um campo binário que possui a capacidade de armazenar uma série variada de dados (MORAES DE OLIVEIRA, 2013, p.1).

Existem quatro tipos de *blob* em um banco de dados SQL, o *TINYBLOB*, *BLOB*, *MEDIUMBLOB* e o *LOB*, suas diferenças estão somente no tamanho máximo de bytes que podem armazenar (MORAES DE OLIVEIRA, 2013, p.1).

Algumas características ao utilizar esse método de armazenamento de imagens são,

- armazenamento da imagem em um campo do banco de dados;
- alguns SGBD possuem extensões espaciais para suportar dados matriciais;
- imagens são armazenadas em uma localização central, única;
- possibilidade de portabilidade;
- relativo aumento do tamanho do Banco de Dados;
- baixa velocidade nas operações (PIRES DE CASTRO FILHO; LOURENÇO, 2010, p. 17).

#### 2.2.4.2. Metadados

Metadados são identificados como descritores de dados, cuja finalidade é organizar e documentar, de maneira estruturada, o conteúdo de dados, facilitando por si a recuperação e o acesso dos dados descritos (MACHADO, 2002, p. 55).

Os metadados estão baseados na literatura pelos conceitos de padrão, modelo e arquitetura. Modelos de metadados descrevem, sobre qualquer domínio de aplicação, o conteúdo e o significado dos dados. Os padrões de metadados são modelos que, sobre um consenso das sociedades pesquisadoras passaram a ser utilizados como padrões. Já a arquitetura de metadados possui como função descrever o ambiente provedor de interoperabilidade entre os padrões de metadados presentes (GARCIA, 1999, p. 36).

Existem algumas soluções comerciais de modelos de metadados como o projeto DISIMA, o Metadados para Documentação e Recuperação de Imagens, *Digital Imaging and Communications in Medicine* (DICOM) e entre outros (MACHADO, 2002, p. 60-69).

Algumas características de se utilizar o método de indicação de objetos por metadados são:

- aponta a localização dos arquivos;
- banco de dados relativamente menor;

- maior eficiência nas operações;
- armazenamento de imagens descentralizado;
- maior dificuldade em lidar com arquivos inválidos ou ausentes;
- pouca portabilidade (PIRES DE CASTRO FILHO; LOURENÇO, 2010, p. 17).

### **3. MATERIAIS E MÉTODO**

Nesta seção são detalhados os métodos e materiais utilizados no presente projeto.

#### **3.1. MATERIAIS**

- Câmera fotográfica
- Adesivos coloridos
- Suporte para Câmera
- Computador
- Softwares para Programação
- Softwares para Banco de Imagens

#### **3.2. MÉTODO**

O trabalho foi dividido nas seguintes fases, podendo elas serem realizadas paralelamente:

##### **a) Revisão da literatura**

Foram estudados inicialmente os conceitos de processamento digital de imagens e banco de dados de imagens, analisando os principais assuntos que foram inseridos no projeto para a realização do objetivo.

##### **b) Ambiente de Trabalho**

Nesta fase objetivou-se a montagem e melhoria do ambiente de trabalho para a captura das imagens, integrando a câmera com o sistema de iluminação. A princípio a câmera é fixada em um suporte, onde não há controle na iluminação indireta na qual sua intensidade pode variar conforme a aquisição das imagens.

##### **c) Processamento e Análise de Imagens**

As imagens capturadas no ambiente de trabalho foram tratadas com base nos conceitos teóricos de processamento de imagens digitais, através de algoritmos para manipulação e categorização das mesmas.

##### **d) Avaliação dos Resultados**

As imagens resultantes do processamento digital foram avaliadas pelo próprio autor, para análise de identificações de possíveis nevos por software, e sua estrutura de propriedades foi documentada, incluindo métodos utilizados, níveis de iluminação, tamanho de imagem, entre outros.

Foi utilizado para este trabalho um sistema de banco de dados, que possibilitou o armazenamento, acesso e manutenção de um histórico das imagens coletadas e geradas após o processamento das mesmas. Após será realizada a recuperação das imagens armazenadas com propósito de comparação de suas características.

Foi adotado com objetivo de armazenamento para histórico de imagens, um banco de dados do tipo MySQL (MYSQL), através do aplicativo de servidor *Apache* (APACHE HTTP SERVER), um servidor web livre, nele é contido uma interface em PHP (HYPERTEXT PREPROCESSOR) o *PhpMyAdmin* (PHPMYADMIN), o qual possibilitou o gerenciamento e administração do banco de dados.



## 4. PROJETO

O sistema proposto é composto de dois subsistemas: 1 - Aquisição, processamento e armazenamento das imagens no banco de dados (Figura 12). 2 – Recuperação das imagens do banco de dados através de seleção por metadados (nome e parte do corpo), registro das imagens e comparação dos vetores de características (Figura 16).

### 4.1. SUBSISTEMA 1

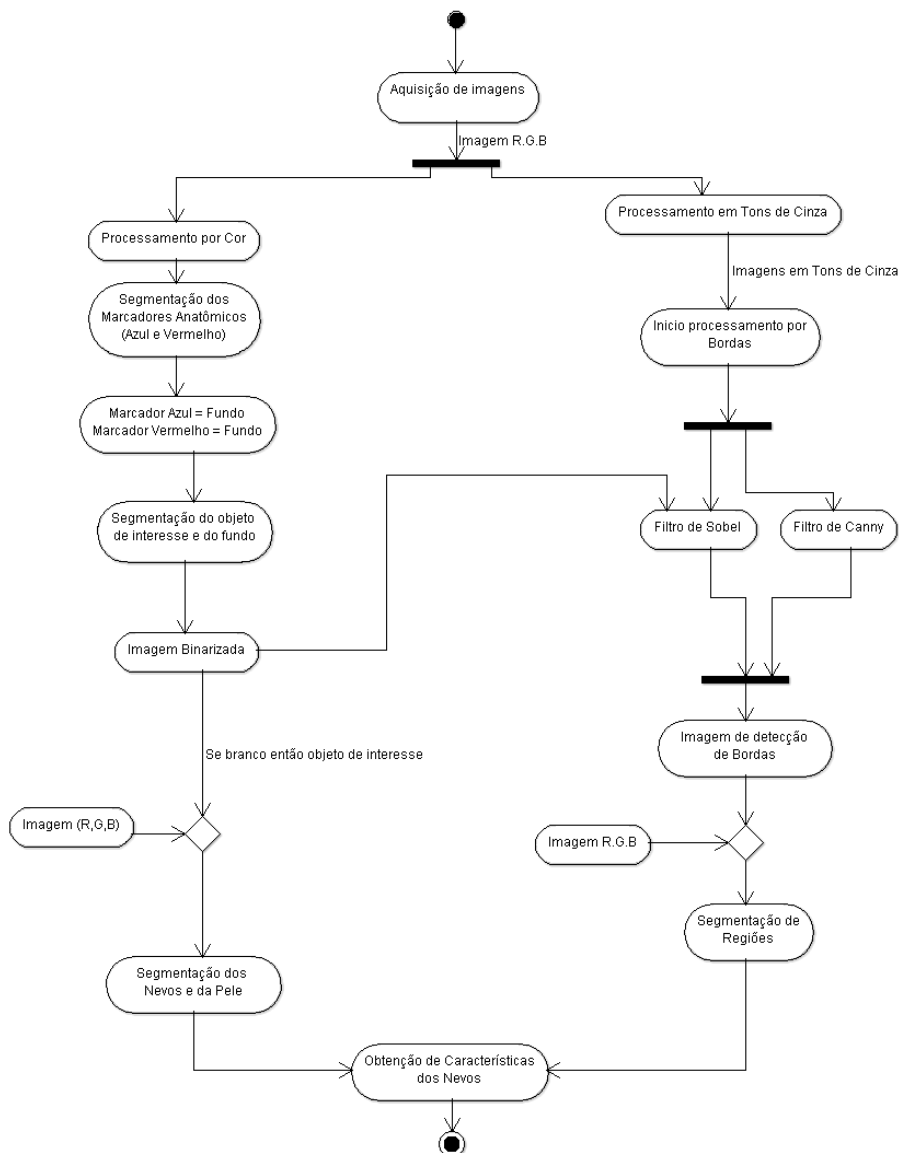


Figura 11 - Fluxograma de procedimentos do projeto subsistema 1.

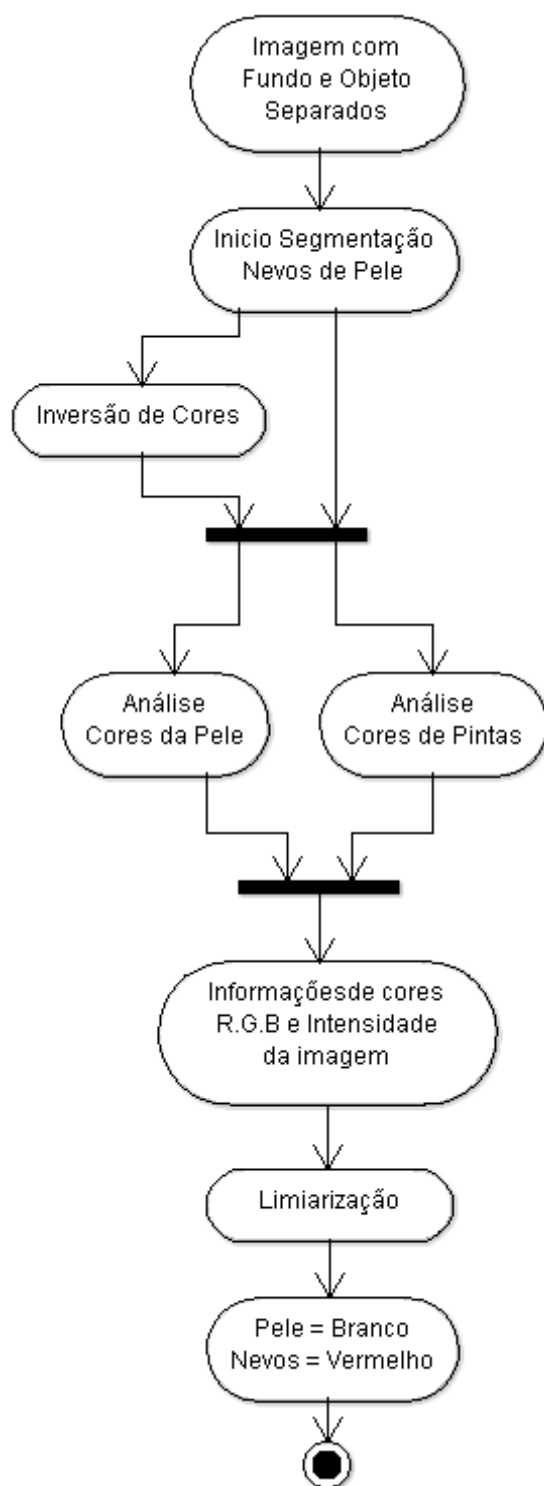
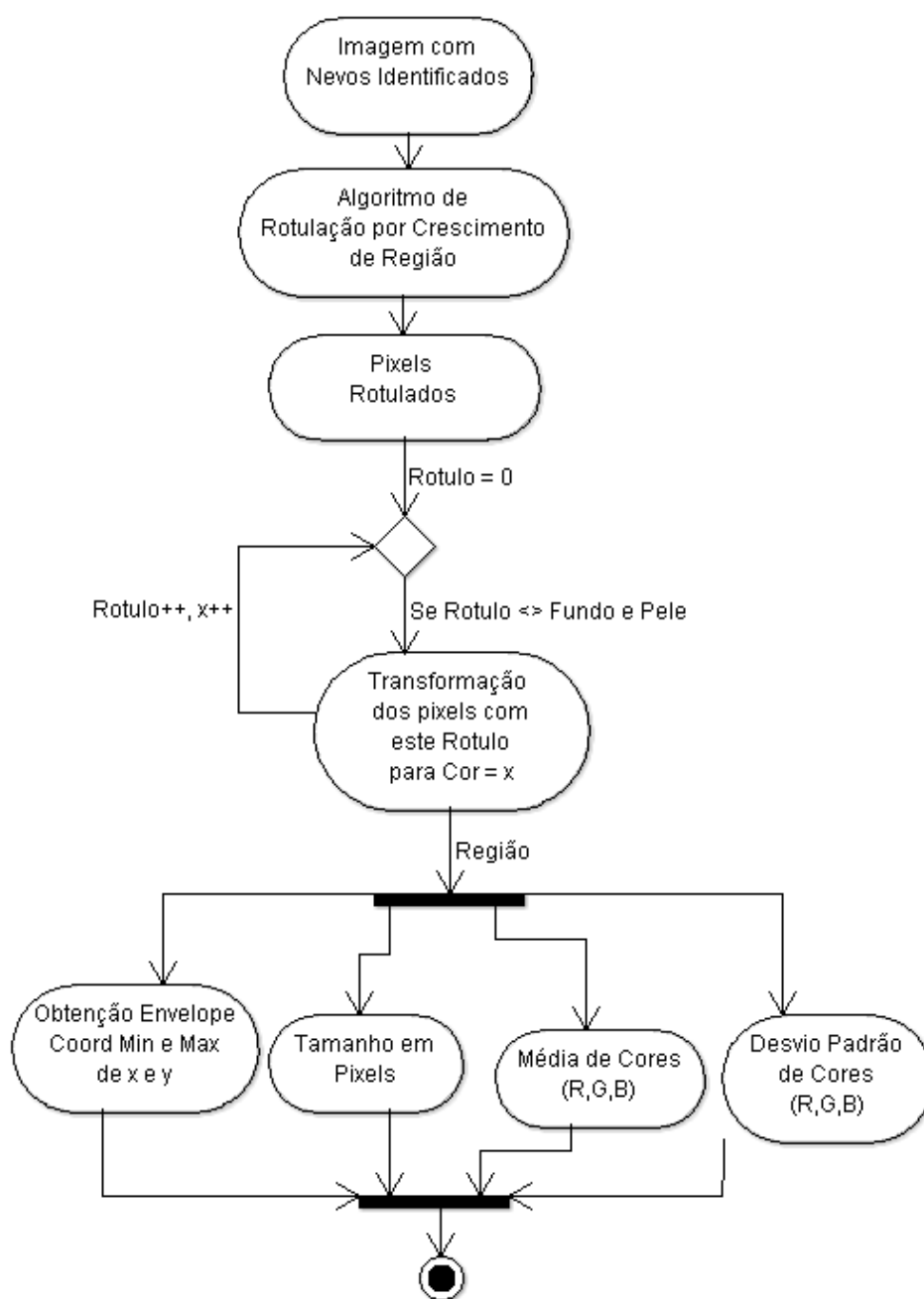


Figura 12 - Fluxograma do processo de Segmentação de Nevos da pele






**Figura 13 – Obtenção de características de cada região**

Para a realização dos procedimentos do projeto, foram inicialmente coletadas, em um ambiente com iluminação artificial e natural, um acervo de imagens, sendo elas do antebraço esquerdo e direito e de ambas as mãos, provenientes do mesmo indivíduo (Figura 15).



**Figura 14 – Exemplos de captura de diferentes objetos alvo.**

Durante o procedimento foi realizada uma variação na distância da câmera com o objeto alvo, iniciou-se com uma distância de 31 cm até 40 cm com variação de 1 cm, para cada valor de distância foram capturadas 5 imagens, buscando para cada imagem diferenciar a posição do objeto alvo em relação a captura anterior (Tabela 1).

Distância (cm)	Imagem (.jpg)
31	
36	
40	

**Tabela 1 – Exemplos de imagens capturadas conforme suas distâncias.**

As imagens foram capturadas com um fundo de cor verde homogêneo como estratégia para um melhor processamento de segmentação do objeto de interesse (ex: antebraço, mão...) e o fundo. Sua iluminação era proveniente de luzes artificiais internas e da iluminação natural externa, não possuindo um controle sobre a intensidade de iluminação durante o procedimento.

Após análise das imagens adquiridas foi identificado que a melhor configuração de extração de imagens, para obtenção de melhores resultados na segmentação por cor com a webcam utilizada foi a distância de 36 cm da câmera com a base cuja parte do corpo está apoiada.

Com o objetivo da realização de registro e comparação de características entre duas ou mais imagens, para cada parte do corpo foram definidos dois pontos anatômicos, e foi inserido um marcador em cada ponto, para auxiliar no registro da imagem. O registro da imagem permite a recuperação de imagens correspondentes à mesma parte do corpo e análise das mesmas a partir do ponto base (ponto anatômico).

O subsistema 1 teve em sua sequência uma divisão de suas formas de processamento de imagens, sendo eles processamento por cor e em tons de cinza. Essas duas formas de processamento podem ser utilizadas tanto isoladamente, quanto em conjunto para a obtenção do resultado desejado.

Para obtenção das características dos resultados de imagens, é necessário determinar quais características devem ser observadas e analisadas. Essas características são descritores de informações da imagem, tais como descritores de fronteiras, regionais e relacionais (GONZALEZ; WOODS, 2002, p. 353-369, 399-402).

Os descritores de fronteiras têm como finalidade identificar diâmetro de uma fronteira, sendo o diâmetro a distância entre dois pixels da fronteira. Buscam também a identificação de curvatura, que é definida pela taxa de mudança de inclinação entre outras aplicações (GONZALEZ; WOODS, 2002, p. 353,354).

Os descritores regionais buscam identificar a área de uma região, definindo o número de pixels contidos em sua fronteira, também avalia-se o perímetro, determinado por quantidade de pixels contidas em uma fronteira, são descritos também pelos eixos principais de uma região, as propriedades topológicas entre outras (GONZALEZ; WOODS, 2002, p. 359,360).

Neste projeto busca-se, através de um histórico de imagens, a obtenção de características relevantes para a identificação de mudanças nos nevós. As características selecionadas foram: tamanho em pixels de cada região, média e desvio padrão para cada canal de cor (R,G,B) de cada região, e as coordenadas de um envelope retangular que contém a região (mínimos e máximos de x e y).

Para a extração de características das imagens processadas, foram inicialmente rotulados os pixels de cada nevo presente na imagem, através da rotulação de regiões por rotulação de crescimento.

A rotulação de regiões tem por base rotular cada região conexa presente na imagem, uma maneira de se implementar este processo é através do algoritmo de crescimento de região. Este algoritmo é iniciado através de um pixel randômico presente na imagem, a partir deste pixel são rotulados os pixels semelhantes vizinhos e conexos a ele (pixels não pertencentes ao fundo e a pele), este pixel é considerado como semente para o crescimento desta região conexa. Após a rotulação de todos os pixels conexos da região, é repetido o algoritmo até serem rotulados todos os pixels presentes na imagem (ALSINA, 2011, p. 9 e 10).

#### 4.2. SUBSISTEMA 2



**Figura 15 – Fluxograma do Subsistema 2**

Para o projeto do Subsistema 2 (Figura 16) inicia-se através da recuperação das imagens armazenadas pelo Subsistema 1 no Banco de Dados. Esta recuperação

é feita através de uma seleção dos metadados presentes no banco, como nome do indivíduo e parte do corpo e marcadores anatômicos.

Sobre as imagens selecionadas é realizado o registro das mesmas, que consiste em realizar o alinhamento das imagens, através da rotação de uma imagem com base na outra, trata-se da integração das imagens com objetivo de criar uma visualização composta entre elas, possibilitando então a comparação do vetor de características presentes nas duas imagens (SHARMA; THÉ, 2013, p.1).

Registro é uma transformação geométrica que relaciona as coordenadas de imagens, cujo objetivo é eliminar distorções existentes em uma imagem ou entre duas imagens. Algumas das aplicações do registro de imagens é na integração de imagens obtidas por sensores ou câmeras diferentes, imagens adquiridas em tempos diferentes, imagens obtidas em posições diferentes e entre outros (SPRING; Aula 3, 1996, p.1).

O registro de imagem é utilizado em aplicações que necessitam obter informações precisas a partir de imagens ou realizar comparação entre elas, como no sensoriamento remoto e na medicina. Um exemplo de uso é a detecção de tumores a partir de imagens de exames de ressonância magnética (SHARMA; THÉ, 2013, p.1).

Para a realização de registro de imagens em uma aplicação, é necessária a determinação da abordagem considerada eficaz para o processo e isso pode ser um procedimento complexo e de elevado tempo. Uma das abordagens é a seleção de um ou mais pontos de referência (marcadores) presentes na imagem e também a elaboração de um método de comparação de informações para o seu alinhamento correto (SHARMA; THÉ, 2013, p.1).

Em um procedimento de registro é necessário:

- 1) Escolher os pontos de controle da imagem: Esta etapa consiste na identificação precisa de pontos (marcadores) presentes nas imagens. Estes pontos são considerados pontos equivalentes nas imagens analisadas.
- 2) Definir a equação de mapeamento: escolher a equação matemática que fará a reamostragem dos pixels.
- 3) Definir o processo de Interpolação: escolher o processo que adicionará novos pontos de pixels baseados nos pixels existentes.

(SPRING; Aula 3, 1996, p.1).

A Figura 17 exemplifica o registro de duas imagens (imagem de ajuste e de referência).

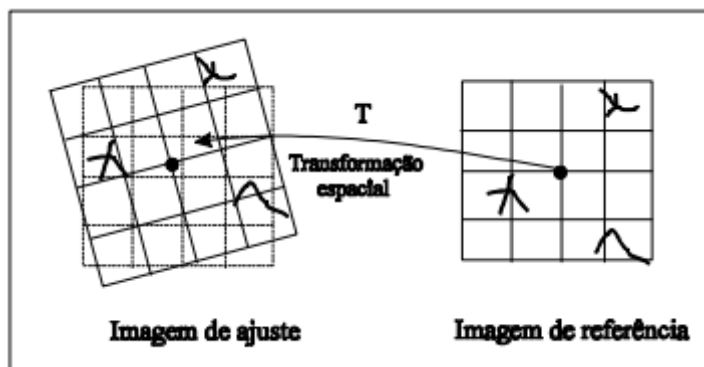


Figura 16 – Exemplo de Transformação espacial de imagem (Registro)  
 Fonte: Spring (1996, p. 2).

Existem duas maneiras de registrar imagens: o modo automático e o modo manual.

O modo automático é baseado na similaridade ou dissimilaridade entre duas imagens, as quais são baseadas no deslocamento relativo existente entre as mesmas. A similaridade é determinada de maneira direta pela correlação entre a mesma área presente nas duas imagens, quanto maior a correlação, maior é a similaridade das áreas (SILVA, 2001).

O registro manual é baseado na identificação de pontos de controle das imagens e, ao invés de correlacionar as coordenadas em pixel/linha de uma imagem às coordenadas geográficas de outra imagem, irá correlacionar as coordenadas pixel/linha dos pontos de controle correspondentes (de cada imagem) entre si (SILVA, 2001).

Após o registro das imagens, a imagem alinhada resultante possui seu vetor de características espaciais alterado, como, por exemplo, a sua posição em pixels do envelope de cada região da imagem (máximos e mínimos valores de pixels dos eixos  $x$  e  $y$ ). Portanto a comparação do vetor de características entre as duas imagens deve ser efetuada a partir deste vetor de características alterado pelo processo de registro de imagens.

#### 4.3. BANCO DE DADOS

Na criação do Banco de Dados para o projeto é utilizado um banco local de dados, o qual permite a interação entre o mesmo e o processamento das imagens,



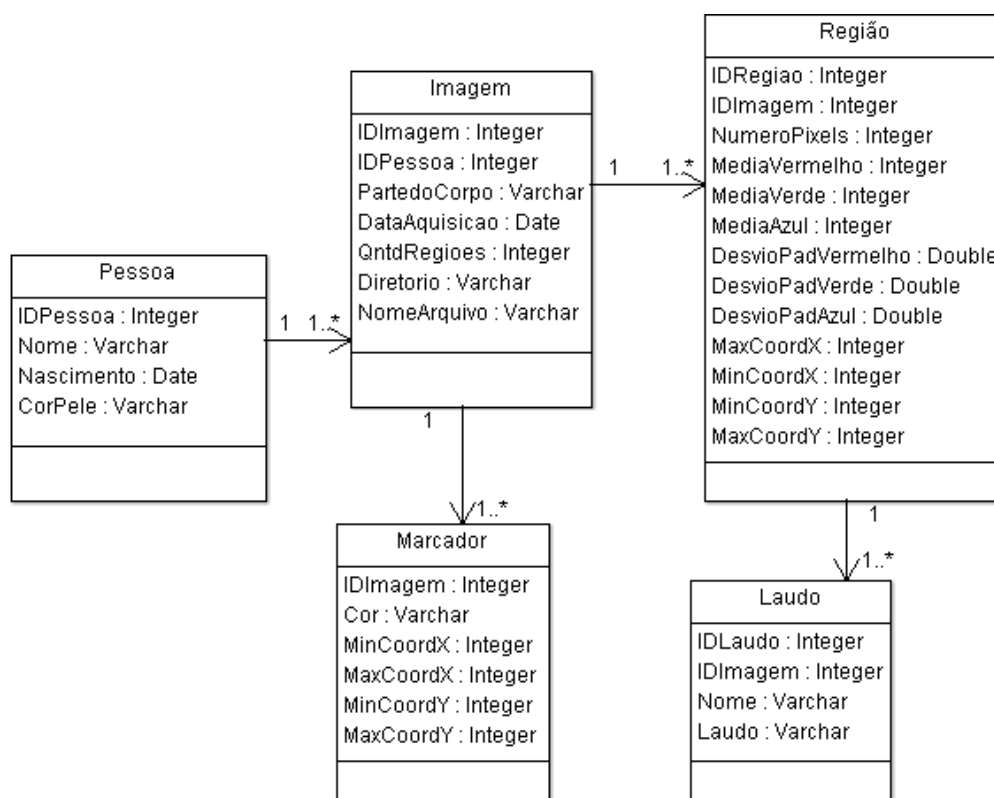
possibilitando o gerenciamento através de um SGBD para criação, alteração e remoção de tabelas, e inserção, remoção e alteração dos dados.

O Banco de Dados foi projetado através da análise das necessidades de informações para serem armazenadas juntamente com as imagens. Após estudo de métodos de armazenamento de imagens, foi estipulado que o armazenamento das imagens seria pelo caminho em que a imagem está armazenada (Diretório) e o nome na qual ela foi definida (NomeArquivo).

Foram criadas tabelas para armazenar os dados e as chaves estrangeiras para possibilitar o relacionamento entre elas. Como exemplo, a tabela “Pessoa” tem um relacionamento de “1 para 1...\*” com a tabela “Imagem”, indicando que mais de uma imagem pode ser referenciada a uma única pessoa.

#### 4.3.1. Diagrama Entidade e Relacionamento

A Figura 18 é a representação do diagrama de Entidades e Relacionamentos do Banco de Dados criado. A primeira tabela “Pessoa” contém as informações de indexação, do nome, da data de nascimento e da cor da pele, ela possui um relacionamento com a tabela “Imagem” de “1 para 1...\*”.



**Figura 17 – Diagrama entidade e relacionamento do Banco de Dados**

A tabela “Imagem” armazena as seguintes informações: indexação da imagem, indexação da pessoa referente à chave primária “IDPessoa” da tabela “Pessoa”, a parte do corpo, a data da aquisição, a quantidade de regiões obtidas pelo processamento, o diretório na qual a imagem está armazenada e o nome do arquivo de imagem. Esta tabela possui relação de “1 para 1” com a tabela “Laudo”, de “1 para 1...\*” com a tabela “Região” e “1 para 1...\*” com a tabela “Marcador”.

A tabela “Região” representa cada região identificada no processamento da imagem, e contém as informações de indexação da região, indexação da imagem através da chave-primária “IDImagem” da tabela “Imagem”, o número de pixels, a média e o desvio padrão de cada canal de cor (R,G,B) e as coordenadas mínimas e máximas referentes à imagem da região.

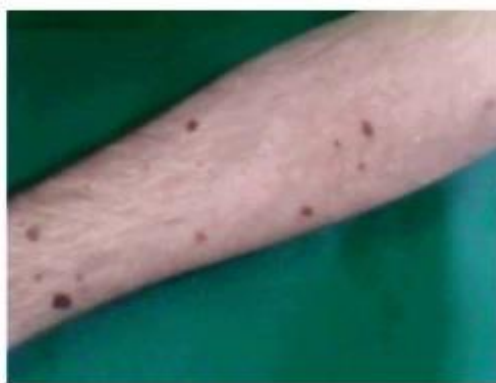
A tabela “Laudo” é referente ao laudo de cada região e contém informações de indexação do laudo, indexação da região a que o laudo se refere, através da chave-primária “IDRegiao”, o nome do médico que criou o laudo e o texto referente ao laudo da região/nevo.

A tabela “Marcador” identifica os marcadores presentes na imagem, nela estão contidos a indexação da imagem “IDImagem” referente a chave-primária “IDImagem” da tabela “Imagem”, a cor do marcador, e as coordenadas mínimas e máximas do marcador presente na imagem.

## 5. IMPLEMENTAÇÃO

### 5.1. AQUISIÇÃO DE IMAGENS

As imagens capturadas possuem o formato compactado de imagem *Joint Photographic Experts Group* (JPEG) (GOMES; COSTA, 2005). A resolução das imagens é 640x480 pixels e elas foram categorizadas através do nome de seus arquivos digitais e nome do diretório onde foram armazenadas, que indicavam a distância entre a câmera e o braço, possibilitando uma melhor organização na realização do projeto (Figura 19).



35cm-20-11-1

**Figura 18 – Exemplo de armazenamento de Imagem.**

Para a construção dos algoritmos de processamento de imagens se utilizou a biblioteca *FiltersDll* (FILTERSDLL, 2009) editada na linguagem em C, a qual proporcionou as ferramentas de filtros e edições de imagens necessários para a realização do projeto.

Esta biblioteca proporciona a identificação das informações de cada pixel da imagem a ser processada, obtendo informações sobre cores, intensidade, tamanho e entre outras, possibilitando a edição, comparação e criação de imagens digitais.

A partir da definição do processo de recuperação das imagens armazenadas no banco de dados, verificou-se a necessidade de se utilizar dois marcadores, os quais teriam a função de auxiliar no registro de modo manual das imagens. Isto é, o alinhamento das imagens seria feito com base nos pixels referentes aos marcadores, basicamente, como um *ponto de controle* (Figura 20).

Utilizou-se dois adesivos de cores distintas entre si para a inserção dos marcadores no objeto alvo do estudo em pontos anatômicos conhecidos (articulações), cuja identificação foi realizada através dos canais de cores (R, G, B) por intermédio do programa MatLab, proporcionando-se a diferenciação entre a pele e os marcadores (Figura 20).

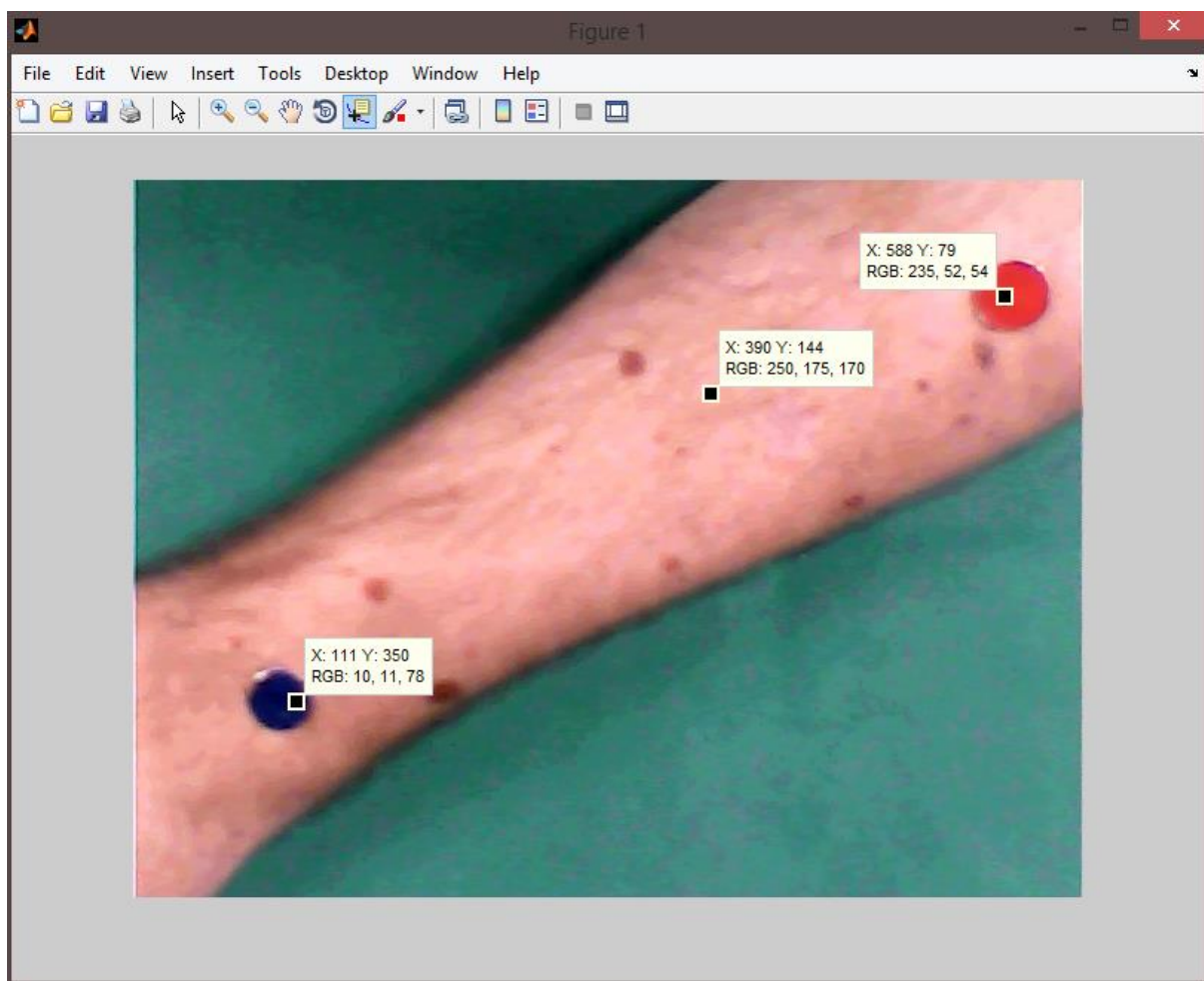


Figura 19 – Exemplo inserção de marcadores

## 5.2. PROCESSAMENTO POR COR

Os processamentos por cores são considerados métodos que envolvem a análise e modificação de acordo com as cores que a imagem representa através de seus pixels. Os pixels das imagens capturadas possuem informações no modelo de cores *Red, Green, Blue* (R,G,B) e representam a intensidade da cor em cada pixel como ilustra a Figura 21.

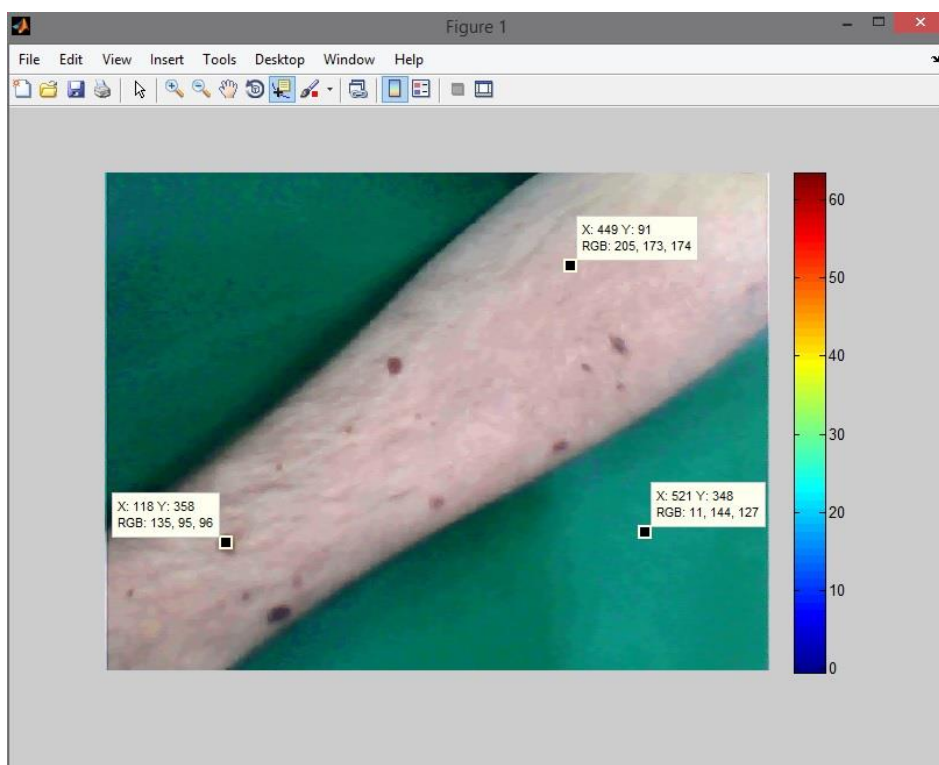


Figura 20 - Exemplo de intensidade de cores Matrix Laboratory (MATLAB).

### 5.2.1. Segmentação dos Marcadores Anatômicos

O processo de segmentação dos marcadores deu-se pelo processamento por cor através dos resultados da análise manual efetuada com o auxílio do MatLab, identificando-se os limiares para segmentação dos marcadores inseridos sobre a pele. Nessa etapa identificaram-se também os valores da posição dos marcadores nas coordenadas mínimas e máximas de X e Y. A Listagem 1 apresenta o código de segmentação dos marcadores.

```
int bminx = 1000, bmaxx = 0, bminy = 1000, bmaxy = 0;
int rminx = 1000, rmaxx = 0, rminy = 1000, rmaxy = 0;
if((redcolor <= 50) && (greencolor <= 50) && (bluecolor >= 60)){
    if(row < bminx){
        bminx = row;
    }
    if(row > bmaxx){
        bmaxx = row;
    }
    if(col < bminy){
```

```
    bminy = col;
}
if(col > bmaxy){
    bmaxy = col;
}
    color2 = image_color32(0, 0 , 0);
    *src2 = color2;
}else if((redcolor > 185) && (greencolor < 80) ){
    color2 = image_color32(0, 0 , 0);
    *src2 = color2;
if(row < rminx){
    rminx = row;
}
if(row > rmaxx){
    rmaxx = row;
}
if(col < rminy){
    rminy = col;
}
if(col > rmaxy){
    rmaxy = col;
}
}
else{
    color2 = image_color32(redcolor, greencolor , bluecolor);
    *src2 = color2;
}
```

**Listagem 1 – Código de identificação dos Marcadores (azul e vermelho)**

Após a identificação dos pixels dos marcadores, estes foram convertidos em “fundo”, a fim de que não fossem processados nas etapas posteriores juntamente com os pixels correspondentes à pele (Figura 22).

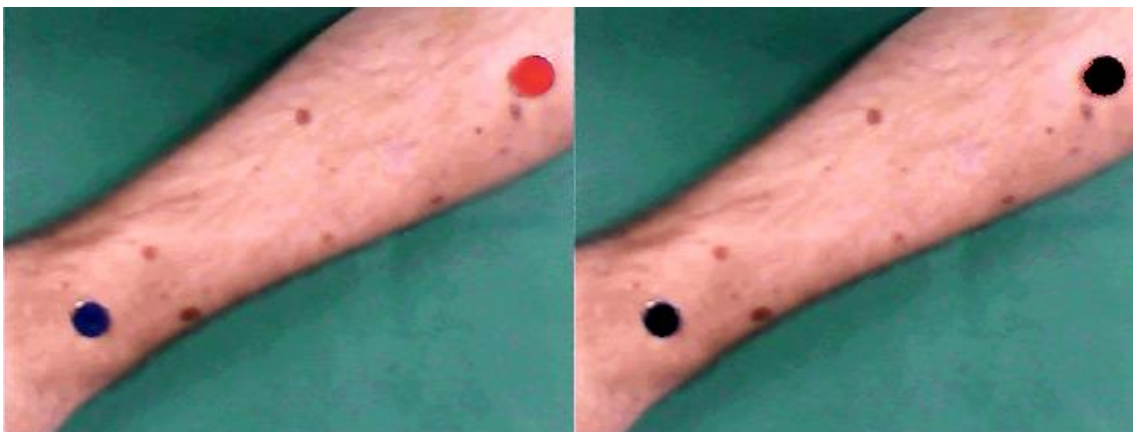


Figura 21 – Resultado Segmentação e transformação para fundo (cor preta) dos Marcadores

### 5.2.2. Segmentação do Objeto de Interesse do Fundo

A segmentação do fundo de uma imagem e do objeto de interesse é um procedimento inicial necessário para este projeto, possibilitando assim o foco do processamento somente na área do corpo.

Foi realizada a separação do fundo das imagens através da comparação do valor da cor vermelha de cada pixel, produzindo um limiar entre o fundo e o objeto. Isso foi determinado previamente com auxílio do programa *Matrix Laboratory* (MATLAB). Quando o valor da cor vermelha do pixel for menor que 49, o mesmo é considerado como parte do fundo e quando for maior ou igual a este limiar o pixel é considerado como pertencente ao objeto de interesse (braço), como mostram as Figura 23 e 24.

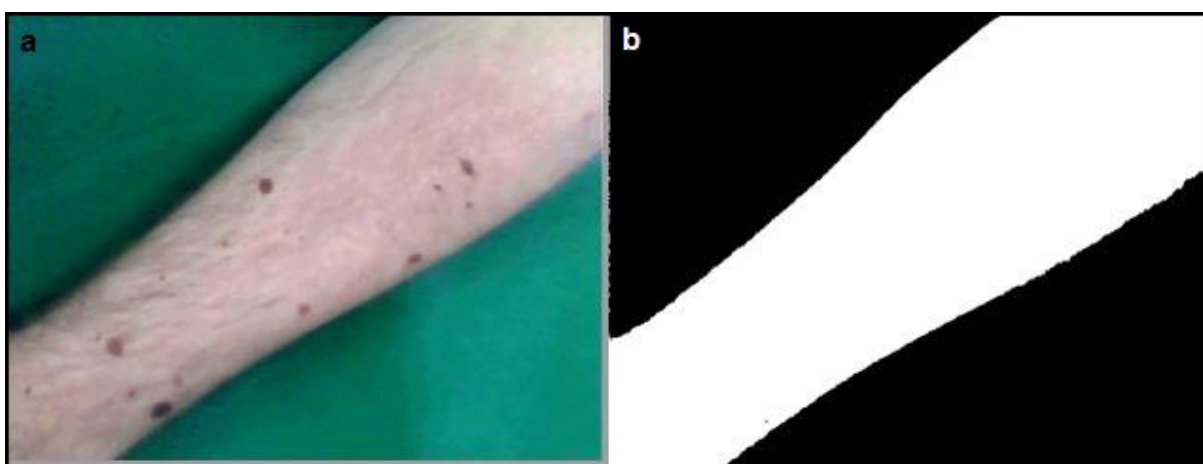


Figura 22 - a) Imagem Original, b) Imagem Binarizada.



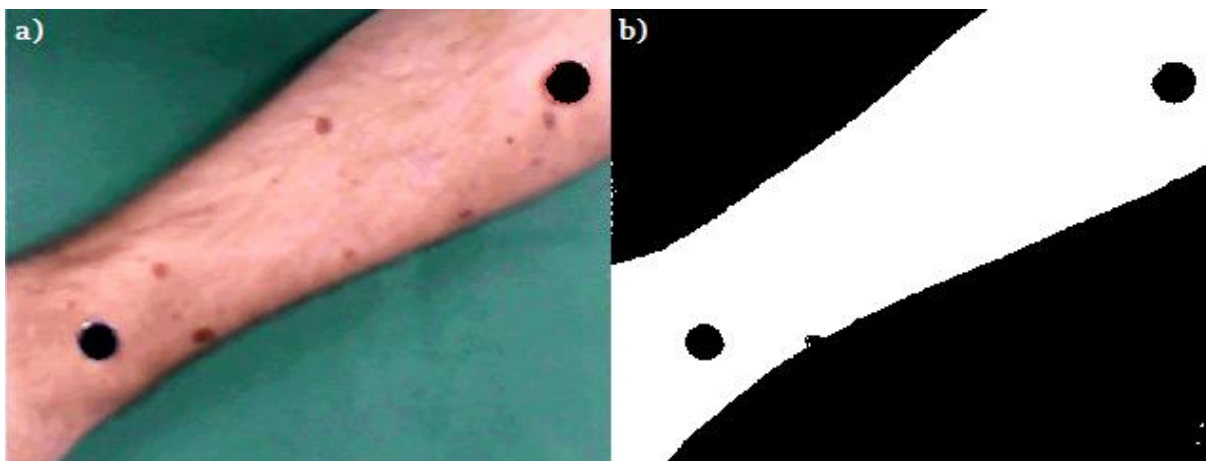


Figura 23 – a) Imagem Original (Marcadores), b) Imagem Binarizada

Os pixels da Figura 23-a que representam o fundo foram transformados para cor preto, zerando os valores dos três componentes de cor (0, 0, 0), enquanto os pixels da mesma imagem que foram classificados como objeto foram transformados para cor branca, armazenando o valor máximo (255, 255, 255), seu resultado representado na Figura 23-b.

A imagem resultante (Figura 23-b) é considerada como a imagem binarizada, nela se diferencia o objeto de interesse do fundo da imagem, permitindo que os próximos processamentos sejam realizados somente no objeto de interesse.

```

1 PFBtmap32 imageLoaded = helper_loadImage( entrada );
2 if( imageLoaded != NULL ){
3   PFBtmap32 imageSaida = image_createImageLike( imageLoaded );
4   TFColor32 *src = imageLoaded->Bits;
5   TFColor32 *src2 = imageSaida->Bits;
6   TFColor32 color2;
7   for(int row=0; row<imageLoaded->Height; row++){
8     for(int col=0; col<imageLoaded->Width; col++){
9       TFColor32 color = *src;
10      redcolor = image_redComponent( color );
11      greencolor = image_greenComponent( color );
12      bluecolor = image_blueComponent( color );
13      if(redcolor < 49 ){
14        color2 = image_color32(0, 0, 0);
15        *src2 = color2;
16      }else{
17        color2 = image_color32(255, 255, 255);
18        *src2 = color2;
19      }
20      src++;
21      src2++;
22    }
23  }
24  helper_saveImage( imageSaida, saida );
25  image_freeImage( imageSaida );
}

```

Listagem 2 – Algoritmo de Binarização

A listagem 2 apresenta a implementação do algoritmo de binarização. Na linha 1 a imagem de entrada é carregada, na sequência é verificado se a imagem existe (linha 2), é criada então uma imagem adicional com o mesmo formato da imagem de entrada (linha 3).

Nas linhas 4 e 5 são inicializados ponteiros para o início do segmento de memória da imagem de entrada e da imagem de saída. Na linha seguinte é criada uma variável auxiliar para o ponteiro da imagem de saída.

As linhas 7 e 8 percorrem os pixels de ambas as imagens, na linha 9 é inicializada uma variável de auxílio para o ponteiro da imagem carregada e é atribuído o valor do local de memória em que o ponteiro está indicado. Após, as linhas 10, 11 e 12 armazenam os valores das componentes de cor vermelha, verde e azul do pixel atual (R,G,B).

Uma verificação do componente de cor vermelha ocorre na linha 13, se a condição for verdadeira as linhas 14 e 15 transformam o pixel da imagem criada para cor preta, se a condição for falsa, as linhas 17 e 18 transformam o pixel da imagem criada para cor branca. As linhas 20 e 21 servem para incremento dos ponteiros e as linhas 24 e 25 são para salvar a imagem criada e liberar o espaço de memória alocado.

Após a binarização da imagem, é realizada uma comparação da imagem binarizada com a imagem original de entrada. Essa comparação é feita através da identificação dos pixels de cor preta, os quais representam o fundo, da imagem binarizada, transformando os pixels de mesma posição da imagem original para a cor preta, mantendo a imagem original os valores originais somente para os pixels do objeto de interesse os quais estão com a cor branca na imagem binarizada, resultando a imagem com o fundo e o objeto de interesse segmentados (Figura 25).



**Figura 24 – Resultado segmentação entre braço (objeto de interesse) e fundo.**

### 5.2.3. Segmentação de Nevos da Pele

Com a pele segmentada nas imagens capturadas, iniciou-se então o processo de identificação dos nevos presentes na imagem. Esse processo consiste na segmentação de regiões com cores que possam ser diferentes da cor de pele.

Inicialmente foi necessário buscar informações sobre as características das cores presentes na pele e das cores presentes nos nevos. Através da análise dessas informações desencadeou-se a comparação entre a pele e os nevos.

Com auxílio do MATLAB verificou-se os valores originais através das componentes de cor (R,G,B) e através da intensidade do tom de cinza. Observou-se, de forma empírica, uma diferença dos valores da componente vermelha, que com combinações da componente verde e a intensidade de tons de cinza, possibilitou a segmentação dos nevos de aproximadamente 65% das imagens testadas (de um total de 45 imagens capturadas e processadas).

A Figura 26 ilustra as informações obtidas de cores em cada pixel (R,G,B), evidenciando as diferenças entre as cores da pele e dos nevos. Já a Figura 27 mostra as informações extraídas da diferença de intensidade dos tons de cinza (*Index*).

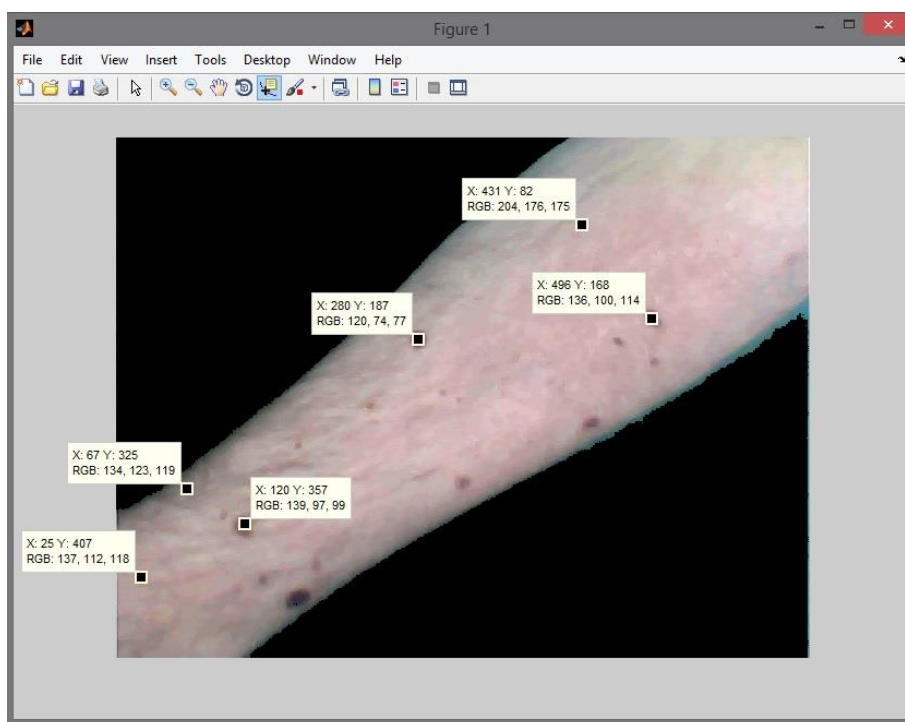
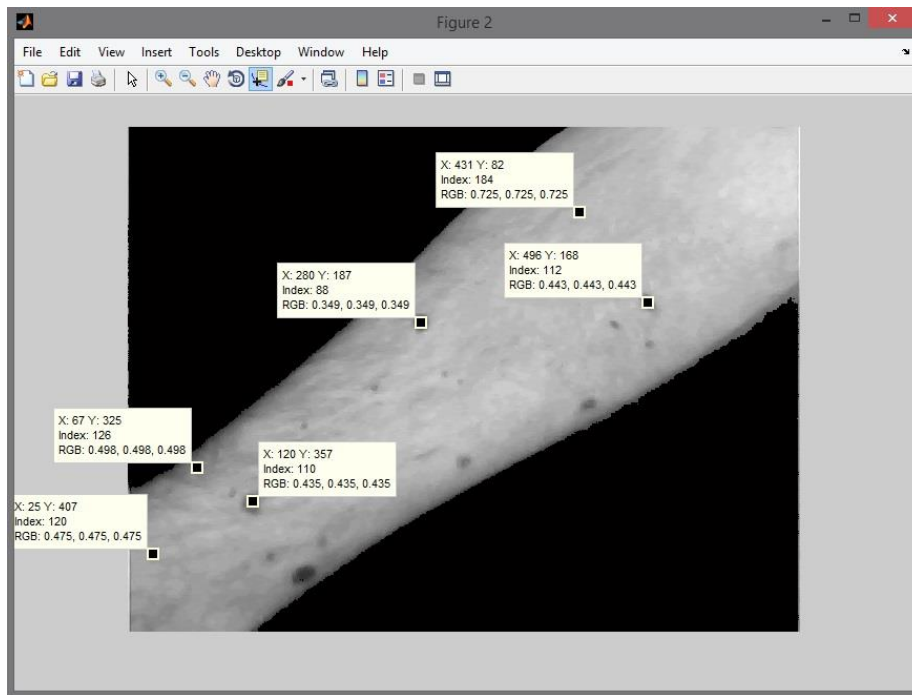


Figura 25 – Representação de informações das Cores no MATLAB (R.G.B.)



**Figura 26 – Representação de informações dos Tons de Cinza MATLAB (Index).**

Tendo como ponto de partida os valores observados após a análise de cores e intensidade, foi desenvolvido um algoritmo para segmentação de nevos da pele baseado em cor. A listagem 2 apresenta a comparação de intensidades e de cores do algoritmo de segmentação de nevos.

```

1  if((redcolor > 30) && (greencolor > 30) && (bluecolor >
2  30))
3  {
4      if(((redcolor < 160) && (redcolor > 90)) && (intensity
5  <125))
6      {
7          if((greencolor < (redcolor - 18)))
8          {
9              color2 = image_color32(255,0,0);
10             *src2 = color2;
11         }else {
12             color2 = image_color32(255,255, 255);
13             *src2 = color2;
14         }
15     }else if((redcolor < 70) && (greencolor < redcolor)){
16         color2 = image_color32(255,0, 0);
17         *src2 = color2;
18     }else {
19         color2 = image_color32(255,255, 255);
20         *src2 = color2;
21     }
22 }

```

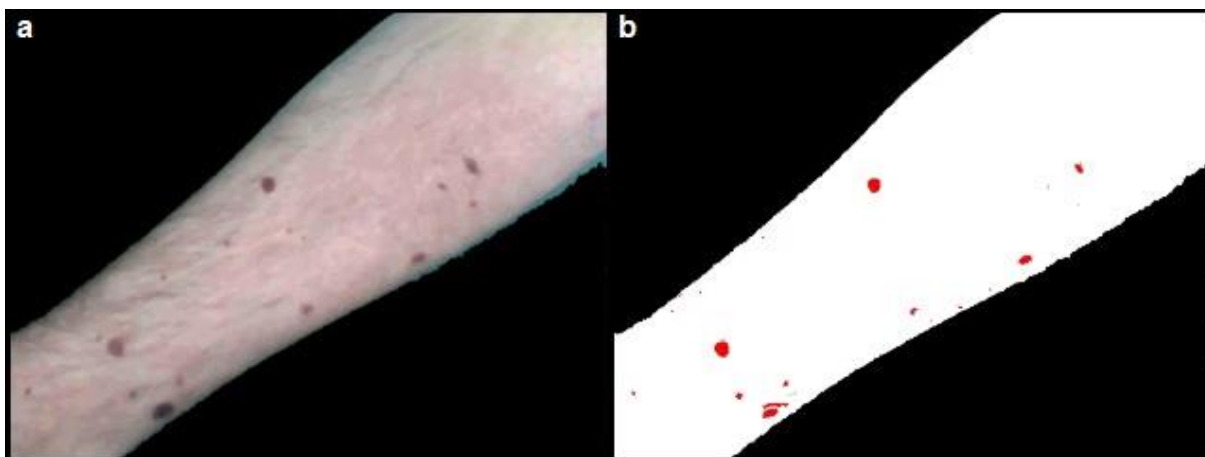
```

21 }else{
22     color2 = image_color32(0,0, 0);
23     *src2 = color2;
24 }

```

**Listagem 3 – Algoritmo de Identificação de Nevos**

A segmentação de nevos por cor no domínio espacial representada na listagem 3 consiste em um laço, o qual percorre a imagem de entrada e produz um valor para cada pixel na imagem de saída. Ele inicialmente realiza a verificação das componentes de cor do pixel (R,G,B), diferenciando o fundo da imagem da pele. Após, são realizadas duas verificações para diferenciar os pixels dos nevos e os pixels que pertencem a pele. Os pixels classificados como nevos são transformados na imagem de saída para a cor vermelha, já os pixels classificados como pele, são transformados para a cor branca, como forma de segmentação da imagem (Figura 28).



**Figura 27 – a) Imagem de entrada. b) Imagem criada para identificação dos nevos**

#### 5.2.4. Segmentação de Nevos da Pele com Inversão de Cores

Outra forma avaliada para a segmentação de nevos na imagem se deu através da inversão de cores da imagem de entrada. Utilizou-se como entrada a imagem resultante da separação da pele e fundo para um aumento de desempenho do resultado final. A inversão dos componentes de cor da imagem tem o processo representado na listagem 4.

```

1  PFBmp32 imageLoaded = helper_loadImage(entrada);
2  if( imageLoaded != NULL )
3  {
4      int32 filterInvert = filters_createFilter( "filterInvert" );

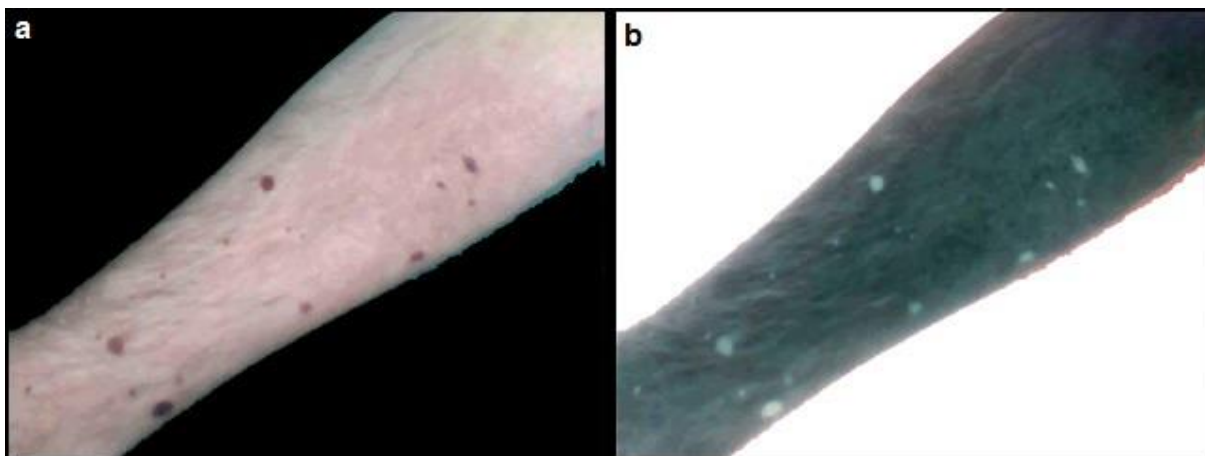
```

```
5   PFBitmap32 imageInvert = image_createImageLike( imageLoaded );
6   filters_setParameterImage( filterInvert, "inImage", imageLoaded );
7   filters_setParameterImage( filterInvert, "outImage", imageInvert );
8   filters_run( filterInvert );
9   helper_saveImage( imageInvert, saida );
10  image_freeImage( imageInvert );
11  filters_deleteFilter( filterInvert );
12  image_freeImage( imageLoaded );
13 }
```

**Listagem 4 – Algoritmo de Inversão de Cores.**

O algoritmo de inversão de cores carrega a imagem de entrada (linha 1), após, ele cria uma outra imagem para saída com os mesmos padrões da imagem de entrada (linha 5), na sequência é gerada a imagem de saída (linha 7) e o armazenamento da mesma (linha 9).

O resultado da aplicação deste algoritmo é a inversão das cores da imagem de entrada como mostram as Figuras 29-a e 29-b.



**Figura 28 – a) Imagem de Entrada, b) Imagem resultado da Inversão de Cores.**

Após o processamento de inversão de cores, a imagem de resultado (Figura 29-b), foi utilizada como entrada para o processo de segmentação de nevos presentes na pele. Com o auxílio do programa MATLAB, foi realizada a verificação das informações de cores e intensidade de tons de cinza contidos nos pixels das imagens representadas nas figuras 30 e 31.

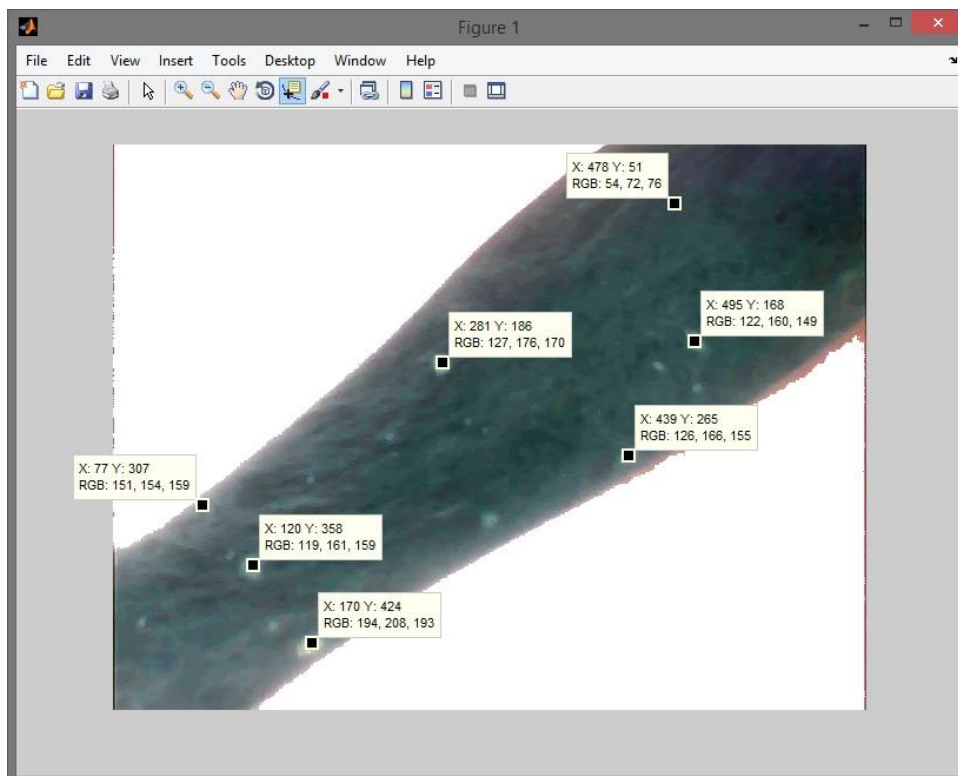


Figura 29 – Imagem de Cores Invertidas MATLAB.

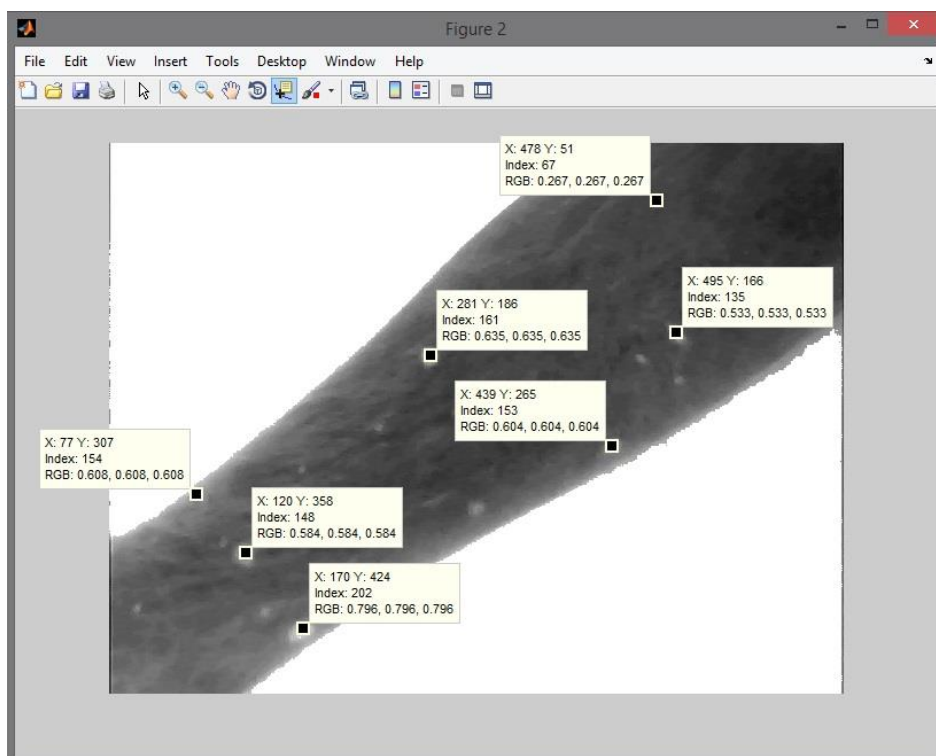


Figura 30 – Imagem de Cores Invertidas em Tons de Cinza MATLAB.

A análise empírica realizada tendo como base as informações obtidas das cores e da intensidade de cinza dos pixels, possibilitou a implementação do algoritmo de segmentação de nevos (listagem 4).

```

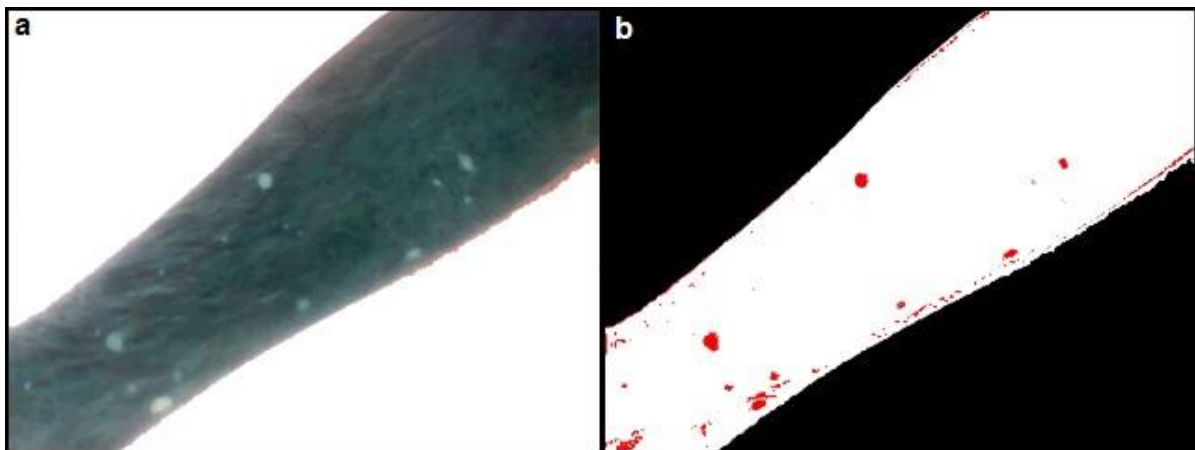
1  if((redcolor != 255) && (greencolor != 255) && (bluecolor != 255))
2  {
3      if((redcolor > 100) && (redcolor < 160) && (intensity > 110)){
4          if(((greencolor > (redcolor + 20)) || (greencolor < (redcolor - 20)))
5              || ((bluecolor > (redcolor + 20)) || (bluecolor < (redcolor - 20))))
6          {
7              color2 = image_color32(255,0,0);
8              *src2 = color2;
9          }else {
10             color2 = image_color32(255,255, 255);
11             *src2 = color2;
12         }
13     }else if(intensity > 190){
14         color2 = image_color32(255,0, 0);
15         *src2 = color2;
16     }else {
17         color2 = image_color32(255,255, 255);
18         *src2 = color2;
19     }
20 }else{
21     color2 = image_color32(0,0, 0);
22     *src2 = color2;
23 }

```

**Listagem 5 - Algoritmo de Identificação de Nevos de Imagem com Inversão de Cores**

O algoritmo representado pela listagem 5 é parte do processamento de identificação de nevos da pele com a imagem de entrada de cores invertidas. Nele foram comparados os valores de cores e intensidade de tons de cinza de cada pixel da imagem de entrada.

Os pixels identificados como pertencentes ao fundo da imagem de entrada (Figura 32-a), foram alterados para a cor preta na imagem de criação (Figura 32-b), os pixels identificados como parte da pele (Figura 32-a) foram alterados para a cor branca (Figura 32-b) e os pixels identificados como nevos (Figura 32-a) foram alterados para a cor vermelha (Figura 32-b).

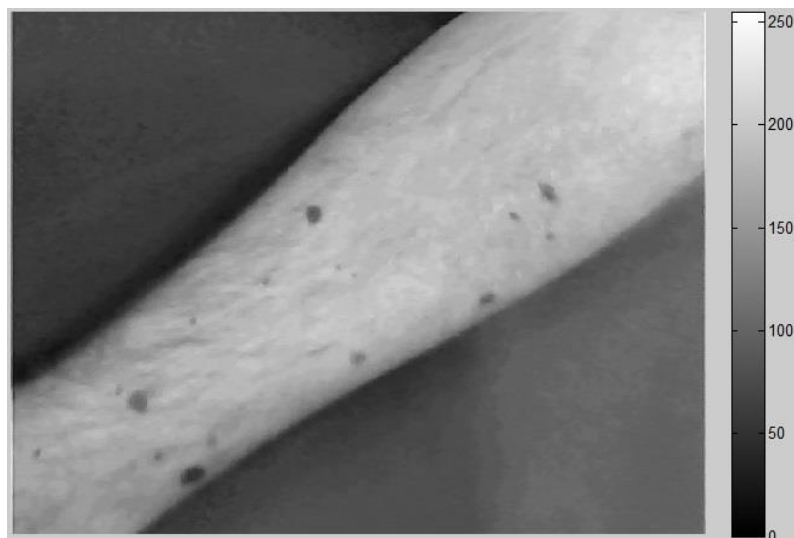


**Figura 31 - a) Imagem de Entrada, b) Imagem Resultado de Segmentação de Nevos.**

### 5.3. PROCESSAMENTO EM TONS DE CINZA



O processamento em tons de cinza se aplica a imagens com suas cores transformadas em valores de níveis de cinza. Seus valores variam de 0 a 255, o qual o valor 0 representa a cor preta como menor intensidade e o 255 representa a cor branca com maior intensidade, como ilustra a Figura 33.



**Figura 32 - Exemplo de Imagem em tons de Cinza**

As principais abordagens de processamento em tons de cinza são as detecções de descontinuidades de imagens. Neste projeto foram realizados testes utilizando os filtros de detecção de borda Sobel e Canny.

### 5.3.1. Filtro de Sobel

O Filtro de Sobel é um método de processamento de detecção de descontinuidades de uma imagem, detectando bordas horizontais e verticais.

Neste projeto foi inicializado um estudo sobre o método de operação de Sobel, através do algoritmo (listagem 5) implementado para o processamento do filtro de Sobel.

```

1 void Sobel(int thl, int thu)
2 {
3     PFBitmap32 imageLoaded = helper_loadImage( "foto_entrada.jpg" );
4     if( imageLoaded != NULL ){
5         __int32 filterSobel = filters_createFilter( "filterSobel" );
6         PFBitmap32 imageSobel = image_createImageLike( imageLoaded );
7         filters_setParameterImage( filterSobel, "inImage", imageLoaded );
8         filters_setParameterImage( filterSobel, "outImage", imageSobel );
9         filters_setParameterInteger( filterSobel, "blurIteration", 0 );
10        filters_setParameterInteger( filterSobel, "gain", 3 );
11        filters_setParameterInteger( filterSobel, "thresholdLower", thl );
12        filters_setParameterInteger( filterSobel, "thresholdUpper", thu );
13        filters_run( filterSobel );

```

```

14  helper_saveImage( imageSobel, saida );
15  image_freeImage( imageSobel );
16  filters_deleteFilter( filterSobel );
17  image_freeImage( imageLoaded );
18  }
19  }

```

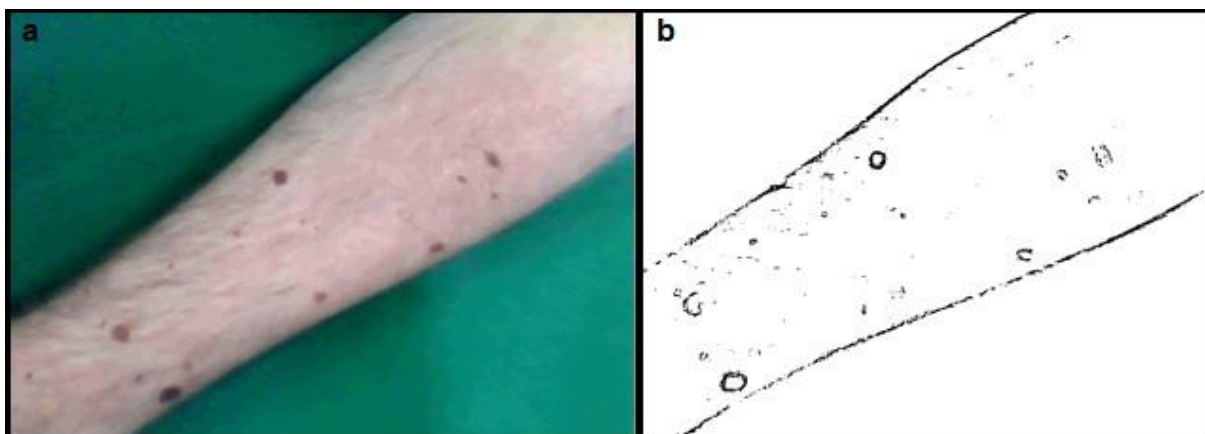
**Listagem 6 – Algoritmo de processamento do Filtro de Sobel.**

O algoritmo da listagem 6 executa a função do Filtro de Sobel, ele recebe como entrada na linha 1 valores inteiros de limiar inferior (thl) e do limiar superior (thu), na linha 3 é carregada a imagem de entrada. Após a verificação da imagem (linha 4), é carregado o filtro de Sobel (linha 5), na linha 6 uma nova imagem é criada com os mesmos parâmetros da imagem de entrada.

As linhas 7 a 12 realizam as configurações dos parâmetros do filtro de Sobel, definindo a imagem de entrada na linha 7, a linha 8 identifica a imagem de saída, a linha 9 configura o parâmetro de interação de *blur* da imagem de saída. Após são configurados o ganho, o qual incrementa os valores das bordas identificadas na imagem de saída possibilitando melhor visualização do resultado, além dos limiares inferior e superior (linhas 10, 11 e 12).

Depois de configurados os parâmetros do filtro o mesmo é executado (linha 13), após a imagem de saída é armazenada e são, por fim, liberados da memória a imagem de saída, o filtro utilizado e a imagem de entrada (linhas 15, 16 e 17).

Realizou-se a análise dos resultados dos testes de variação dos valores de limites inferior e superior, identificando-se visualmente como melhor resultado para a amostra os valores de 30 para o limite inferior e de 150 para o limite superior (Figura 34).



**Figura 33 – a) Imagem de entrada, b) Resultado processamento Sobel (cores invertidas).**

### 5.3.2. Filtro de Canny

Realizou-se o processamento do operador de Canny como outra forma de identificação de bordas das imagens.

O filtro de Canny, assim como o filtro de Sobel, realiza a detecção de bordas, criando a imagem de saída em escala de tons de cinza, aplicando a suavização de ruídos e detectando possíveis bordas.

```

1  PFBitmap32 imageLoaded = helper_loadImage( "foto_entrada.jpg" );
2  if( imageLoaded != NULL ){
3      int32 filterCanny = filters_createFilter( "filterCanny" );
4      PFBitmap32 imageCanny = image_createImageLike( imageLoaded );
5      filters_setParameterImage( filterCanny, "inImage", imageLoaded );
6      filters_setParameterImage( filterCanny, "outImage", imageCanny );
7      filters_setParameterInteger( filterCanny, "threshold1", th1 );
8      filters_setParameterInteger( filterCanny, "threshold2", th2 );
9      filters_run( filterCanny );
10     helper_saveImage( imageCanny, saida);
11     image_freeImage( imageCanny );
12     filters_deleteFilter( filterCanny );
13     image_freeImage( imageLoaded );
14 }

```

**Listagem 7 – Algoritmo de Processamento do Filtro de Canny.**

O algoritmo representado na listagem 7 realiza o processamento do filtro de Canny. Nele é recebido uma imagem de entrada (linha 1) e, após verificação da existência da imagem é chamada a função do filtro de Canny (linha 3) e criada uma nova imagem com os mesmos parâmetros da imagem de entrada (linha 4).

Após, são definidos os valores de parâmetros necessários para a execução do filtro, imagem de entrada, imagem de saída e limites 1 e 2. Finalizadas as configurações dos padrões, o filtro é executado (linha 9) e, então, as imagens e o filtro são desalocados da memória do sistema (linhas 11, 12 e 13).

Foram executados testes de processamento com uma variação dos valores dos limites e, após análise dos resultados das imagens, a imagem com melhor segmentação para esta amostra teve como valores de limites 80 para o primeiro limite, e 130 para o segundo limite, Figura 35-b.

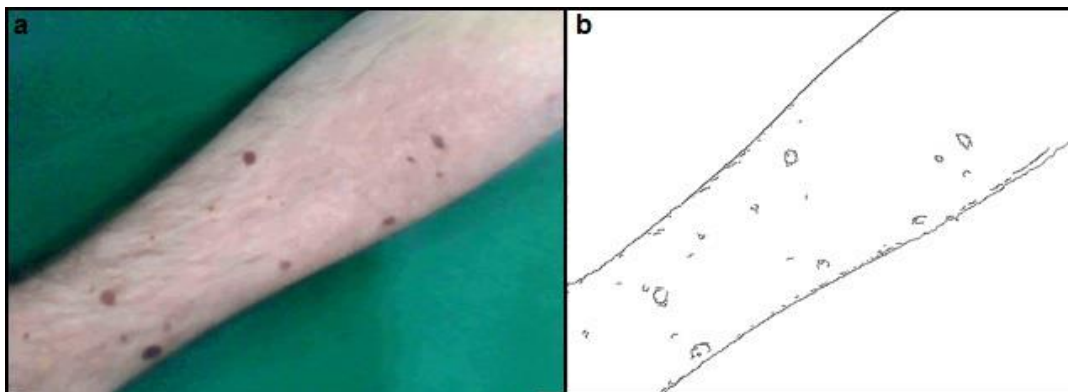


Figura 34 – a) Imagem entrada, b) Resultado processamento filtro de Canny (cores invertidas).

#### 5.4. EXTRAÇÃO DE CARACTERÍSTICAS

Para a extração das características primeiramente foi implementado um algoritmo de rotulação dos nevos segmentados nas imagens (regiões).

##### Algoritmo de Rotulagem de Região

1. Inicializar  $k = 1, j = 1$ .
2. Se  $I(k,j) = 1$ , fazer  $I(k,j) = 255$ . // Seleciona regiões que não são fundo.
3. Fazer  $k = k+1$ . Se  $k \leq m$ , ir para o passo 2.
4. Fazer  $k = 1, j = j+1$ . Se  $j \leq n$ , ir para o passo 2.
5. Inicializar  $k = 1, j = 1, i = 1$ . //  $i$  é o rótulo
6. Se  $I(k,j) = 255$ . // Achou semente
  - a. Fazer  $i = i+1$ . // Atualiza rótulo.
  - b. Aplicar algoritmo de crescimento de região a partir da semente  $(k,j)$ .
7. Fazer  $k = k+1$ . Se  $k \leq m$ , ir para o passo 6.
8. Fazer  $k = 1, j = j+1$ . Se  $j \leq n$ , ir para o passo 6.

##### Algoritmo de Crescimento de Região

1. Fazer  $I(k,j) = i, \text{push}(k,j), \text{push}(0,0)$ . // Rotula e empilha semente, // coloca marca na pilha.
2. Se  $j < n$  AND  $I(k,j+1) = 255$ , // Checa pixel acima.
  - a. Fazer  $I(k,j+1) = i$ . // Rotula pixel acima.
  - b. Push  $(k,j+1)$ . // Empilha pixel acima.
3. Se  $k > 1$  AND  $I(k-1,j) = 255$ , // Checa pixel esquerdo.
  - a. Fazer  $I(k-1,j) = i$ . // Rotula pixel esquerdo.
  - b. Push  $(k-1,j)$ . // Empilha pixel esquerdo.
4. Se  $j > 1$  AND  $I(k,j-1) = 255$ , // Checa pixel abaixo.
  - a. Fazer  $I(k,j-1) = i$ . // Rotula pixel abaixo.
  - b. Push  $(k,j-1)$ . // Empilha pixel abaixo.
5. Se  $k < m$  AND  $I(k+1,j) = 255$ , // Checa pixel direito.
  - a. Fazer  $I(k+1,j) = i$ . // Rotula pixel direito.
  - b. Push  $(k+1,j)$ . // Empilha pixel direito.
6. Pop  $(k,j)$ . Se  $(k,j) \neq (0,0)$  ir para Passo 2. // Desempilha o pixel mais recente.
7. Pop  $(k,j)$ . Retornar. // Restaura endereço da semente

Listagem 8 - Exemplo de Algoritmos de Rotulação

Fonte: ALSINA (2011 p. 10).

A Listagem acima apresenta os algoritmos de rotulação de regiões de uma imagem  $I(k, j)$ . Inicia-se pela busca da semente (pixel de partida), após utiliza-se de uma pilha para empilhamento dos pixels conexos a essas sementes, através do crescimento da região, rotulando os mesmos e por fim repetem-se os passos de busca de sementes e empilhamento dos pixels até rotular toda a imagem.

Neste projeto, o processamento para rotulação dos elementos na imagem iniciou através da varredura de cores dos pixels, identificando em uma matriz o fundo, a pele e os objetos de interesse (nevos). Após serem rotuladas as regiões conexas, cada região foi alterada para uma cor (Listagem 9), cujo fundo e pele foram rotulados como branco (255,255,255) resultando na imagem representada na Figura 36.

```

for(int row=0; row<imageLoaded->Height; row++)
{
    for(int col=0; col<imageLoaded->Width; col++)
    {
        if((x >= 2) && (vetct[x] > 12)) //Objetos (nevos)
        {
            if(x<= 5)
            {
                color2 = image_color32(x*50, 0 , 0);
                *src2 = color2;
            }
            else if((x>5) && (x<=10))
            {
                color2 = image_color32( 0, (x*24)*x , 0);
                *src2 = color2;
            }
            else if((x>10) && (x<=65))
            {
                color2 = image_color32(0, 0, (x*4)-20);
                *src2 = color2;
            }
        }
        else //Fundo ou Pele
        {
            color2 = image_color32( 255 , 255 , 255);
            *src2 = color2;
        }
        src2++;
    }
}

```

**Listagem 9 – Código de rotulação por cores das regiões.**

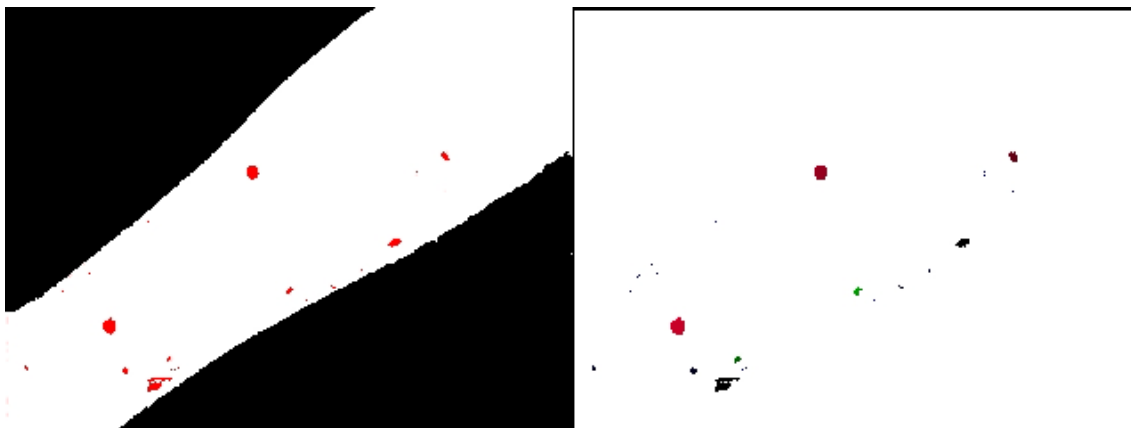


Figura 35 - Resultado de identificação de regiões por cores.

Na sequência foram analisadas as regiões detectadas, e descartadas as regiões com um número de pixels inferior a 12, reduzindo assim o ruído na imagem de resultado (Figura 37). O valor de 12 foi escolhido de forma empírica, após uma análise manual dos nevos presentes e seus tamanhos em pixels.

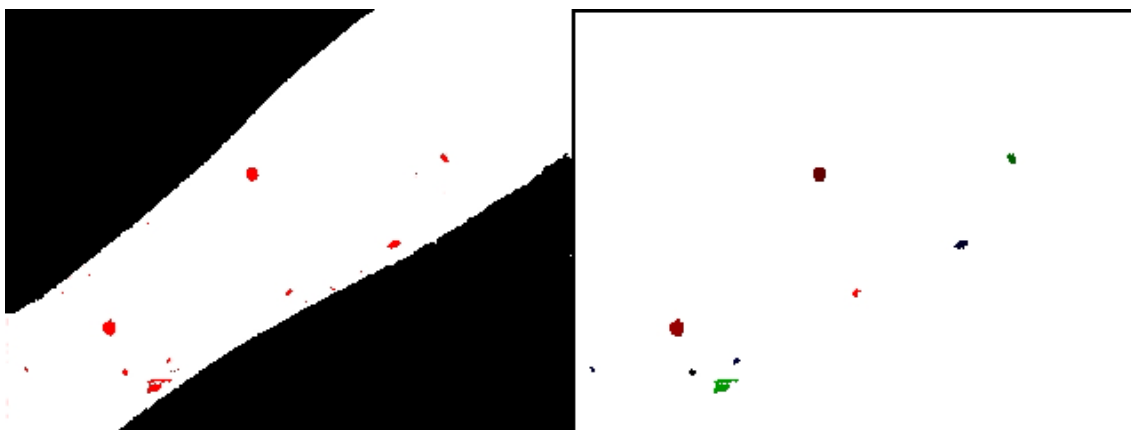


Figura 36 – Resultado de regiões com número de pixels maior que 12.

A partir dos resultados obtidos na rotulação dos nevos presentes na imagem, foram extraídas as características de cada região, como número de pixels, média e desvio padrão para cada um dos canais de cores (R,G,B).

## 5.5. BANCO DE DADOS

Para a implementação do banco de dados foi escolhida a criação de um banco de dados local, através do software EasyPHP, o qual proporciona o gerenciamento do banco de dados MySQL, com objetivo de realização de testes de interação entre os códigos de processamento e o mesmo.

Criou-se as tabelas a partir de códigos SQL conforme mostram as listagens e seus resultados representados na interface gráfica *PHPMyAdmin* conforme as figuras 38 a 42.

```

1 CREATE TABLE Pessoa(
2 IDPessoa integer,
3 Nome Varchar(60),
4 Nascimento date,
5 CorPele Varchar(30)
6 PRIMARY KEY (IDPessoa)
7 );

```

Listagem 10 – Código SQL de criação da tabela Pessoa

The screenshot shows the phpMyAdmin interface for a database named 'tccdatabase'. The 'Pessoa' table is selected, and its structure is displayed in a table format. The columns are: IDPessoa (int(100)), Nome (varchar(60)), Nascimento (date), and CorPele (varchar(30)). The interface also shows a tree view of databases and tables on the left, and a table with row statistics and space usage on the right.

#	Nome	Tipo	Agrupamento (Collation)	Atributos	Nulo	Predefinido	Extra
1	IDPessoa	int(100)			Não	0	
2	Nome	varchar(60)	utf8_unicode_ci		Sim	NULL	
3	Nascimento	date			Sim	NULL	
4	CorPele	varchar(30)	utf8_unicode_ci		Não	None	

Espaço ocupado		Row statistics	
Dados	0 Bytes	Formato	dinâmico
Índice	1 KB	Agrupamento (Collation)	utf8_unicode_ci
Total	1 KB	Registos	0
		Criação	09-Jun-2015 às 13:56
		Última actualização	09-Jun-2015 às 13:56
		Última Verificação	09-Jun-2015 às 13:56

Figura 37 – Resultado Tabela Pessoa

```

1 CREATE TABLE Imagem(
2 IDImagem integer,
3 IDPessoa integer,
4 PartedoCorpo Varchar(30),
5 DataAquisicao date,
6 QntdRegioes integer,
7 Diretorio varchar(80),
8 NomeArquivo varchar(30),
9 PRIMARY KEY (IDImagem),
10 FOREIGN KEY (IDPessoa) REFERENCES Pessoa (IDPessoa)
11 );

```

Listagem 11 – Código SQL de criação da tabela Imagem

phpMyAdmin

Servidor: 127.0.0.1 » Base de Dados: tccdatabase » Tabela: imagem

Procurar Estrutura SQL Pesquisar Inserir Exportar

#	Nome	Tipo	Agrupamento (Collation)	Atributos	Nulo	Predefinido	Extra
1	IDImagem	int(100)			Não	0	
2	IDPessoa	int(100)			Sim	NULL	
3	PartedoCorpo	varchar(30)	utf8_unicode_ci		Sim	NULL	
4	DataAquisicao	date			Sim	NULL	
5	QntdRegioes	int(65)			Sim	NULL	
6	Diretorio	varchar(80)	utf8_unicode_ci		Sim	NULL	
7	NomeArquivo	varchar(30)	utf8_unicode_ci		Sim	NULL	

↑ Todos Com os seleccionados: Procurar Muda Elimina Primár

Information

Espaço ocupado		Row statistics	
Dados	0 Bytes	Formato	dinâmico
Índice	1 KB	Agrupamento (Collation)	utf8_unicode_ci
Total	1 KB	Registos	0
		Criação	09-Jun-2015 às 10:48
		Última actualização	09-Jun-2015 às 10:48
		Última Verificação	09-Jun-2015 às 10:48

Figura 38 – Resultado Tabela Imagem

```

1 CREATE TABLE Regiao(
2 IDRegiao integer,
3 IDImagem integer,
4 NumeroPixels integer,
5 Media integer,
6 DesvioPadrao double,
7 MinCoordX integer,
8 MaxCoordX integer,
9 MinCoordY integer,
10 MaxCoordY integer,
11 PRIMARY KEY (IDRegiao),
12 FOREIGN KEY (IDImagem) REFERENCES Imagem (IDImagem)
13 );

```

Listagem 12 – Código SQL de criação da tabela Região



phpMyAdmin

Servidor: 127.0.0.1 » Base de Dados: tccdatabase » Tabela: regiao

Procurar Estrutura SQL Pesquisar Inserir Exportar

#	Nome	Tipo	Agrupamento (Collation)	Atributos	Nulo	Predefinido	Extra
<input type="checkbox"/>	1 IDRegiao	int(65)			Sim	NULL	
<input type="checkbox"/>	2 IDImagem	int(100)			Sim	NULL	
<input type="checkbox"/>	3 NumeroPixels	int(255)			Sim	NULL	
<input type="checkbox"/>	4 Media	int(11)			Sim	NULL	
<input type="checkbox"/>	5 DesvioPadrao	double			Sim	NULL	
<input type="checkbox"/>	6 MinCoordX	int(11)			Sim	NULL	
<input type="checkbox"/>	7 MaxCoordX	int(11)			Sim	NULL	
<input type="checkbox"/>	8 MinCoordY	int(11)			Sim	NULL	
<input type="checkbox"/>	9 MaxCoordY	int(11)			Sim	NULL	

Com os seleccionados: Procurar Muda Elimina Pr

Information

Espaço ocupado		Row statistics	
Dados	0 Bytes	Formato	static
Índice	1 KB	Agrupamento (Collation)	utf8_unicode_ci
Total	1 KB	Registos	0
		Criação	09-Jun-2015 às 10:44
		Última actualização	09-Jun-2015 às 10:44

Figura 39 – Resultado Tabela Região

1. CREATE TABLE Marcador(
2. IDImagem integer,
3. Cor varchar(30),
4. MinCoordX integer,
5. MaxCoordX integer,
6. MinCoordY integer,
7. MaxCoordY integer,
8. FOREIGN KEY (IDImagem) REFERENCES CUSTOMER (IDImagem)
9. );

Listagem 13 – Código SQL de criação da tabela

phpMyAdmin

Servidor: 127.0.0.1 » Base de Dados: tccdatabase » Tabela: marcador

Procurar Estrutura SQL Pesquisar Inserir Exportar

#	Nome	Tipo	Agrupamento (Collation)	Atributos	Nulo	Predefinido	Extra
<input type="checkbox"/>	1 IDImagem	int(11)			Sim	NULL	
<input type="checkbox"/>	2 Cor	varchar(30)	utf8_unicode_ci		Sim	NULL	
<input type="checkbox"/>	3 MinCoordX	int(11)			Sim	NULL	
<input type="checkbox"/>	4 MaxCoordX	int(11)			Sim	NULL	
<input type="checkbox"/>	5 MinCoordY	int(11)			Sim	NULL	
<input type="checkbox"/>	6 MaxCoordY	int(11)			Sim	NULL	

Com os seleccionados: Procurar Muda Elimina Pr

Information

Espaço ocupado		Row statistics	
Dados	0 Bytes	Formato	dinâmico
Índice	1 KB	Agrupamento (Collation)	utf8_unicode_ci
Total	1 KB	Registos	0
		Criação	13-Jun-2015 às 14:21
		Última actualização	13-Jun-2015 às 14:21

Figura 40 – Resultado Tabela Marcador

```

1 CREATE TABLE Laudo(
2 IDLaudo integer,
3 IDRegiao integer,
4 Medico varchar(60),
5 Laudo varchar(200),
6 FOREIGN KEY (IDRegiao) REFERENCES Regiao (IDRegiao)
7 );

```

Listagem 14 – Código SQL de criação da tabela Laudo

The screenshot displays the phpMyAdmin interface for a MySQL database named 'tccdatabase'. The 'laudo' table is selected, and its structure is shown in a table format:

#	Nome	Tipo	Agrupamento (Collation)	Atributos	Nulo	Predefinido	Extra
1	IDLaudo	int(100)			Sim	NULL	
2	IDImagem	int(100)			Sim	NULL	
3	Medico	varchar(60)	utf8_unicode_ci		Sim	NULL	
4	Laudo	varchar(200)	utf8_unicode_ci		Sim	NULL	

Below the table structure, the 'Information' tab is active, showing the following statistics:

Espaço ocupado		Row statistics	
Dados	0 Bytes	Formato	dinâmico
Índice	1 KB	Agrupamento (Collation)	utf8_unicode_ci
Total	1 KB	Registos	0
		Criação	09-Jun-2015 às 10:47
		Última actualização	09-Jun-2015 às 10:47
		Última Verificação	09-Jun-2015 às 10:47

Figura 41 – Resultado Tabela Laudo

### 5.5.1. Armazenamento

O armazenamento das informações no banco de dados foi feito em linguagem C através da conexão com o Banco de Dados MySQL, a qual foi inserida o conjunto de bibliotecas do *Mysql Connector/C*, o qual proporcionou a conexão e comunicação entre o banco e o código (CONNECTOR, 2015) como mostra a Listagem 15.

```

1 MYSQL conexao;
2 mysql_init(&conexao);
3 if(mysql_real_connect(&conexao,"127.0.0.1","root", NULL,"tccdatabase",
4 0,NULL,0))
5 {
6 printf("conectado com sucesso!\n");
7 mysql_close(&conexao);
8 }
9 else
10 {
11 printf("Falha de conexao\n");
12 printf("Erro %d : %s\n", mysql_errno(&conexao), mysql_error(&conexao));

```

12 }

**Listagem 15 – Código de criação e inicialização da variável MYSQL.**

Inicializou-se uma estrutura do tipo MYSQL através dos comandos das linhas 1 e 2 da Listagem 15. Após a inicialização da variável de MYSQL para a comunicação com o Banco de Dados foi realizada então a conexão com o banco através da função *mysql\_real\_connect* na linha 3.

Esta função recebe como parâmetros a estrutura *MySQL* criada (&conexao), o endereço de conexão ("127.0.0.1"), o usuário ("root"), a senha (não definida), a base de dados ("tccdatabase"), a porta usada (0), socket (*NULL*) e *flags* do cliente (0) respectivamente. Os últimos três parâmetros foram atribuídos para que seus valores sejam "automáticos", de forma que o sistema possa escolher a melhor opção de valores para os mesmos.

Para a interação utiliza-se a função *mysql\_query* a qual recebe de parâmetros a estrutura *MYSQL* e a query no formato *String*. Seu retorno é uma variável de tipo Inteiro. Foram criadas funções de Inserção, remoção e alteração de informações das tabelas do Banco de Dados conforme Listagem 16.

```

1  int inseredb(MYSQL *conexao, char query[] )
2  {
3      int res = mysql_query(conexao,query);
4      if (!res) printf("Registros inseridos %d\n",
mysql_affected_rows(conexao));
5      else printf("Erro na inserção %d : %s\n", mysql_errno(conexao),
mysql_error(conexao));
6      return(res);
7  }

8  int removedb(MYSQL *conexao, char query[] )
9  {
10     int res = mysql_query(conexao,query);
11     if (!res) printf("Registro removido %d\n",
mysql_affected_rows(conexao));
12     else printf("Erro na remocao %d : %s\n", mysql_errno(conexao),
mysql_error(conexao));
13     return(res);
14 }

15 int alteradb(MYSQL *conexao, char query[] )
16 {
17     int res = mysql_query(conexao,query);
18     if (!res) printf("Registro alterado %d\n",
mysql_affected_rows(conexao));
19     else printf("Erro na alteracao %d : %s\n", mysql_errno(conexao),
mysql_error(conexao));
20     return(res);
21 }

```

**Listagem 16 – Código de criação e inicialização da variável MYSQL.**

As funções de inserção, remoção e alteração das informações retornam o valor de retorno (res) da função *mysql\_query*, executam a mesma e verificam se o processo foi efetuado com sucesso ou não.

### 5.5.2. Recuperação

A recuperação das imagens e suas informações associadas, presentes no banco de dados, é importante para a realização da comparação das mesmas através da realização do registro de imagens, em que uma imagem é transformada para seus elementos coincidirem.

Para a realização a recuperação das imagens manteve-se os métodos de armazenamento, porém foram alteradas a função (selectdb) (Listagem 17) e as *queries* de comandos em SQL, para realizar o armazenamento e tratamento das características a serem retornadas.

```

1. void selectdb(MYSQL *conexao, char query[] )
2. {
3.     MYSQL_RES *resp;
4.     MYSQL_ROW linhas;
5.     MYSQL_FIELD *campos;
6.     int conta; //Contador comum
7.     if (mysql_query(conexao,query))
8.         printf("Erro: %s\n",mysql_error(conexao));
9.     else
10.    {
11.        resp = mysql_store_result(conexao);//recebe a consulta
12.        if (resp) //se houver consulta
13.            {
14.                campos = mysql_fetch_fields(resp);
15.                for (conta=0; conta<mysql_num_fields(resp); conta++)
16.                    {
17.                        printf("%s", (campos[conta]).name);
18.                        if (mysql_num_fields(resp)>1)
19.                            printf("\t");
20.                    }
21.                printf("\n");
22.                while ((linhas=mysql_fetch_row(resp)) != NULL)
23.                    {
24.                        for (conta=0; conta<mysql_num_fields(resp); conta++)
25.                            printf("%s\t",linhas[conta]);
26.                        printf("\n");
27.                    }
28.            }
29.        mysql_free_result(resp);//limpa a variável do resultado: resp
30.    }
31. }

```

**Listagem 17 – Função de retorno de informações do Banco de Dados.**

## 5.6. RESULTADOS E DISCUSSÕES

Ao decorrer do desenvolvimento dos trabalhos, através do processamento de imagens de teste, foram obtidos resultados das características das imagens de entrada. Essas informações de características foram armazenadas no banco de dados.

Elaborou-se o processamento de duas imagens (Figura 43), as quais foram capturadas em tempos diferentes, efetuando a extração das características das mesmas conforme as Tabelas 2 e 3.

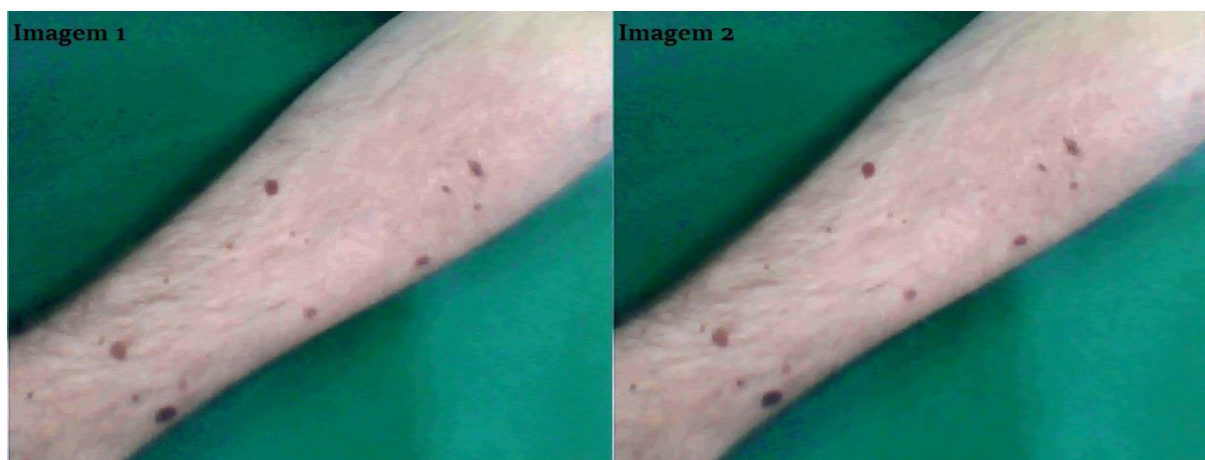


Figura 42 – Imagens de entrada para processamento.

O resultado visual do processamento das imagens pode ser visto na Figura 44, cujos nevos foram identificados com cores aleatórias devido a aleatoriedade no processo de rotulação das regiões.

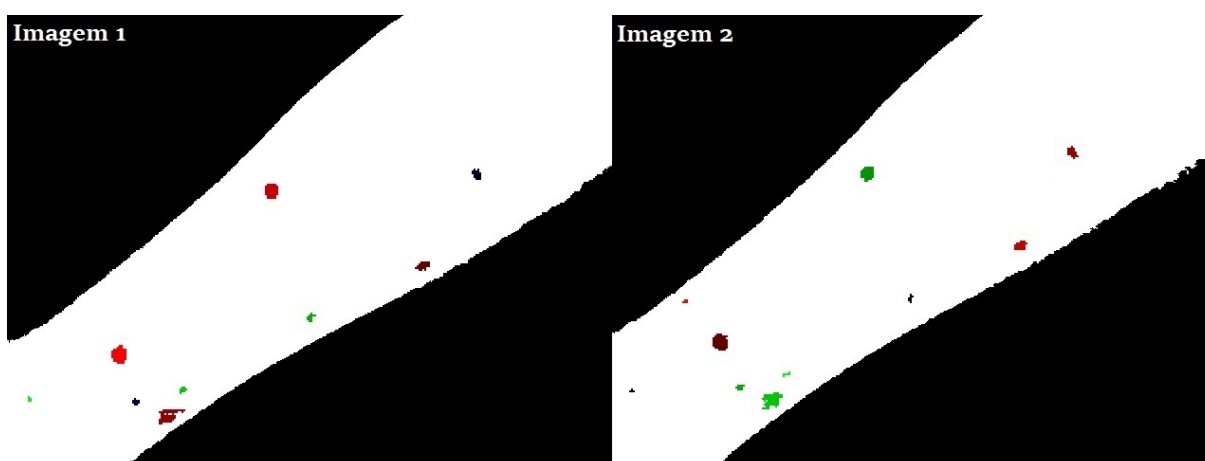


Figura 43 – Imagens pós-processamento.

Tabela 2 – Resultados da extração de características Imagem 1.

Imagem 1						
(R,G,B)	(Pixels)	(R,G,B)	(R,G,B)	(X,Y)	(X,Y)	

Id	Cor	Nº	Média	Desvio Padrão	Min	Max
1	(250,0,0)	240	(245,111,111)	(15.6, 10.5, 10.5)	(110, 348)	(126, 368)
2	(0,0,28)	88	(237,155,154)	(15.3, 12.3, 12.3)	(490, 162)	(500, 174)
3	(100,0,0)	110	(242, 95, 100)	(15.5, 10.8, 11.1)	(431, 262)	(446, 270)
4	(200,0,0)	200	(243, 92,91)	(15.5, 9.5, 9.5)	(272, 178)	(286, 194)
5	(0,0,64)	44	(250,140,141)	(15.5, 11.8, 11.8)	(133, 405)	(141, 84)
6	(150,0,0)	244	(238, 145, 145)	(15.3, 12, 12)	(160, 416)	(188, 432)
7	(0,200,0)	40	(238,185,185)	(15.2, 13.5, 13.5)	(182, 392)	(190, 400)
8	(0,175,0)	52	(242,187,187)	(15.4, 13.5, 13.5)	(316, 314)	(326, 324)
9	(0,250,0)	20	(234,182,181)	(15, 13.2, 13.2)	(22, 402)	(26, 408)

**Tabela 3 – Resultados da extração de características Imagem 2.**

Imagem 2						
	(R,G,B)	(Pixels)	(R,G,B)	(R,G,B)	(X,Y)	(X,Y)
Id	Cor	Nº	Média	Desvio Padrão	Min	Max
1	(100,0,0)	240	(245, 98, 98)	(15.5, 9.8, 9.8)	(106, 337)	(122, 354)
2	(150,0,0)	96	(243,168,168)	(15.4, 12.9, 12.9)	(480, 138)	(492, 152)
3	(200,0,0)	108	(242,126,126)	(15.4, 11.1, 11.1)	(424, 238)	(438, 248)
4	(0,150,0)	180	(246,99,100)	(15.6, 9.9, 9.9)	(262, 160)	(276, 176)
5	(0,175,0)	44	(249,132,132)	(15.6, 11.4, 11.4)	(130, 390)	(140, 396)
6	(0,200,0)	244	(246,137,137)	(15.6, 11.6, 11.6)	(158, 398)	(180, 416)
7	(0,250,0)	24	(243,206,206)	(15.4, 14.2, 14.2)	(180, 376)	(188, 382)
8	(0,0,32)	32	(241,211,211)	(15.3, 14.4, 14.4)	(312, 294)	(318, 304)
9	(0,0,40)	16	(243,203,203)	(15.2, 14, 14)	(18, 394)	(24, 398)
10	(250,0,0)	20	(249,123,123)	(15.5, 10.9, 10.9)	(74, 300)	(80, 304)

As tabelas de características extraídas do processamento das imagens, foram criadas manualmente a partir de análises visuais das imagens e dos resultados de informações obtidas.

As características presentes nas tabelas têm como objetivo a detecção de variações entre as duas imagens, a coluna com atributo “Id” correlaciona as tabelas, cujas regiões consideradas como nevos são equivalentes, considerando as proximidades de posição, observadas visualmente pelo autor. Cada tabela possui como informação, a cor de identificação da região (nevo) na imagem de resultado, o número de pixels, a média e o desvio padrão de cada canal de cor (R,G,B) e as coordenadas Mínimas e Máximas de X e Y (x,y) de cada região identificada.

Após realizar análise dos resultados obtidos pela extração das características, notou-se o surgimento de uma nova região no processamento da “imagem 2” ante o processamento da “imagem 1”.

Um dos nevos aumentou ligeiramente de tamanho (2) enquanto outros reduziram de tamanho (3, 4, 7 e 8) e os demais permaneceram inalterados. Essa

informação pode nortear o especialista na seleção de nevos suspeitos para uma análise aprofundada.

## 6. CONCLUSÃO

Com o aumento da incidência de cânceres de peles no país, se torna importante a realização da prevenção médica dos mesmos. Uma forma de prevenção é a identificação computacional e histórico de nevos presentes na pele, através de imagens digitais.

Neste trabalho foram realizados testes de algoritmos de processamento de imagens, para estudo e análise de possibilidade de identificação de nevos presentes na pele humana. As imagens foram capturadas por uma câmera digital, processadas através de software em linguagem C, utilizando uma biblioteca de filtros de domínio público.

Com a implementação dos algoritmos de processamento propostos, avaliou-se a possibilidade de métodos de segmentação de nevos presentes na pele. Foram realizados testes utilizando os métodos de processamento de imagens por cor e por escala de cinza com bordas e observou-se a necessidade de um estudo aprofundado do tema.

O processamento por cor é uma alternativa dependente da iluminação e dos limiares de segmentação adotados. As imagens utilizadas nos testes foram capturadas do mesmo indivíduo e sob condições de iluminação similares. Entretanto, para que o processo de segmentação possa ser utilizado para imagens provenientes de diferentes indivíduos, acredita-se que uma abordagem híbrida, combinando segmentação por bordas e por cor, possa ser a melhor solução. As informações da cor média e do desvio padrão dos nevos são extraídas a partir dos componentes de cor da região, portanto, a informação de cor não pode ser descartada.

Observou-se que a partir deste trabalho podem ser propostos, através da visão computacional, métodos de segmentação de nevos presentes na pele humana e problemas similares a este.

O armazenamento das informações e características presentes nas imagens e das imagens em si, foi efetuado para proporcionar a comparação entre imagens capturadas em momentos diferentes, possibilitando a identificação das mudanças entre as imagens que constituem o histórico do indivíduo.



Pode-se implementar futuramente o processo de registro entre duas imagens, para a realização da sobreposição das mesmas, facilitando assim o trabalho de identificação de forma automática de relação dos nevos presentes nas imagens.

Como trabalho futuro poderá ser desenvolvida uma aplicação integrada para a aquisição, processamento, armazenamento e recuperação das imagens.

Também poderão ser extraídas futuramente da imagem as características de formas dos nevos, através da aplicação de operadores morfológicos, os quais podem proporcionar a identificação do formato dos nevos presentes.

Outro futuro desdobramento é a incorporar a covariância de cada região como uma das características da imagem, possibilitando uma melhor comparação e identificação entre as imagens e suas características.

## REFERÊNCIAS

AIRES, João Paulo. **Uma arquitetura distribuída para o modelo de metadados metamídia**. Universidade Federal do Paraná-UFPR, Curitiba-PR, 2005. Disponível em: <<http://dspace.c3sl.ufpr.br:8080/dspace/bitstream/handle/1884/2860/DISSERTA%C3%87%C3%83O%20-%20UMA%20ARQUITETURA%20DISTRIBU%C3%8DDA%20PARA%20O%20MODELO%20DE%20M.pdf?sequence=1&isAllowed=y>> acesso em: 12 dez. 2014.

ALSINA, Pablo Javier. **Introdução À Visão Robótica**. Universidade Federal do Rio Grande do Norte-UFRN, Natal-RN, 2011. Disponível em: <<http://www.dca.ufrn.br/~pablo/FTP/robotica/cap7.pdf>> acesso em: 08 jun. 2015.

APACHE HTTP SERVER, Project. Disponível em: <<http://httpd.apache.org/>> acesso em: 17 nov. 2014.

BITTENCOURT, Rogério Gonçalves. **Aspectos Básicos de Banco de Dados**. Florianópolis, 2004. Disponível em: <<https://www.marilia.unesp.br/Home/Instituicao/Docentes/EdbertoFerneda/BD%20-%20Aspectos%20Basicos.pdf>> acesso em: 22 jun. 2015.

BOTELHO, Glenda M.. **Processamento de imagens no auxílio à detecção de câncer de pele utilizando redes neurais artificiais**. Universidade Federal de Goiás, Câmpus Catalão, 2007. Disponível em: <<http://dcc.catalao.ufg.br/up/498/o/Glenda2007.pdf> > acesso em 26 set. 2014.

BOUDREAUX, Ryan. **CSS3 filter effects: grayscale, invert, sepia, and more**. 2012. Disponível em: <<http://www.techrepublic.com/blog/web-designer/css3-filter-effects-grayscale-invert-sepia-and-more/>> acesso em 12 dez. 2014.

CANNY, John. **A Computational Approach to Edge Detection**. 1986. Disponível em: <[http://www.cim.mcgill.ca/~image529/TA529/Image529\\_99/assignments/edge\\_detection/references/canny-pami.pdf](http://www.cim.mcgill.ca/~image529/TA529/Image529_99/assignments/edge_detection/references/canny-pami.pdf)> acesso em 12 dez. 2014.

CONNECTOR, MySQL. **Connector/C 6.1.6**. 2015. Disponível em: <<https://dev.mysql.com/downloads/connector/c/>> acesso em 22 jun. 2015.

DATE, C. J. **Introdução a Sistemas de Bancos de Dados**. Rio de Janeiro: Elsevier Editora Ltda. 2004.

FILTERSDLL, **FiltersDLL-3.3-2009\_12**. Disponível em: <<http://filters.sourceforge.net/>> acesso em: 02 nov. 2014.

GARCIA, Simone. **Metadados para documentação e recuperação de imagens**. Instituto Militar De Engenharia – IME, Rio de Janeiro-RJ, 1999. Disponível em: <[http://www2.comp.ime.eb.br/dissertacoes/1999-Simone\\_Garcia.pdf](http://www2.comp.ime.eb.br/dissertacoes/1999-Simone_Garcia.pdf)> acesso em: 12 dez. 2014.

GOMES, Alessandra; COSTA, Aline. **JPEG – Formato de arquivo de imagens**. Escola de Ciência da Informação – UFMG, Belo Horizonte – MG, 2005. Disponível em: <<http://www2.dcc.ufmg.br/disciplinas/ii/ii05-1/seminario/jpeg.pdf>> acesso em: 06 dez. 2014.

GONZALEZ, Rafael C.; WOODS, Richard E. **Processamento de imagens digitais**. São Paulo: Editora Edgard Blücher LTDA, 2000.

HOSPITAL A.C.CAMARGO. **Câncer: conhecer para prevenir**. São Paulo: 2014. Disponível em: <<http://www.accamargo.org.br/files/pdf/materiais-de-prevencao/cartilha-conhecer-para-prevenir.pdf>> acesso em: 23 set. 2014

HYPertext PREPROCESSOR, PHP. Disponível em: <<http://php.net/>> acesso em: 18 nov. 2014.

KIOSKEA, **Bancos de dados – Introdução**. 2014. Disponível em: <<http://pt.kioskea.net/contents/65-bancos-de-dados-introducao>> acesso em: 10 jun. 2015.

MACHADO, Miriam Schacker. **Modelo de metadados para armazenamento e recuperação de imagens estáticas no formato DICOM**. Universidade Federal do

Rio Grande Do Sul – PPGC da UFRGS, 2002. Disponível em: <<http://www.lume.ufrgs.br/bitstream/handle/10183/2987/000329954.pdf?sequence=1>> acesso em: 12 dez. 2014.

MATLAB, Matrix Laboratory. Disponível em: <<http://www.mathworks.com/products/matlab/>> acesso em: 04 dez. 2014.

MINISTERIO DA SAÚDE. INCA. **Estimativa 2014: Incidência de câncer no Brasil**. Rio de Janeiro: INCA, 2014. Disponível em: <<http://www.inca.gov.br/estimativa/2014/estimativa-24042014.pdf> > acesso em: 22 set. 2014.

MORAES DE OLIVEIRA, Daniel Cardoso. **Introdução ao uso de imagens no MySQL**. PESC - COPPE/UFRJ, Rio de Janeiro, 2013. Disponível em: <<http://www.devmedia.com.br/introducao-ao-uso-de-imagens-no-mysql/10041>> acesso em: 17 nov. 2014.

MYSQL. Disponível em: <<http://www.mysql.com/>> acesso em: 17 nov. 2014.

PEREIRA, Gustavo A. **Nevos melanocíticos (pintas ou sinais)**. 2012. Disponível em: <<http://www.cancerdepele.net.br/nevos> > acesso em: 22 set. 2014.

PHPMYADMIN. Disponível em: <[http://www.phpmyadmin.net/home\\_page/index.php](http://www.phpmyadmin.net/home_page/index.php)> acesso em: 18 nov. 2014.

PIRES DE CASTRO FILHO, Carlos Alberto; LOURENÇO, Leonardo Arthur Esteves. **Banco de dados de imagens: características e aplicações**. Instituto Nacional de

Pesquisas Espaciais-INPE, 2010. Disponível em: <[http://wiki.dpi.inpe.br/lib/exe/fetch.php?media=cap378:cap378\\_bd\\_imagens\\_lourenco\\_pires.pdf](http://wiki.dpi.inpe.br/lib/exe/fetch.php?media=cap378:cap378_bd_imagens_lourenco_pires.pdf)> acesso em: 12 dez. 2014.

RESOURCES NETWORK, C++. 2013. Disponível em: <<http://www.cplusplus.com/>> acesso em: 03 dez. 2014

ROCHA, Anderson; SANTANCHÈ, André. **Bancos de Dados: Teoria e Prática**. Universidade Estadual de Campinas – UNICAMP, Campinas-SP, Instituto de Computação – IC, 2012. Disponível em: <<http://www.ic.unicamp.br/~rocha/teaching/2012s2/mc536/aulas/aula-04.pdf>> acesso em: 12 dez. 2014.

SCURI, Antonio Escaño. **Fundamentos da Imagem Digital**. Tecgraf/PUC-Rio de Janeiro-RJ, 2002. Disponível em: <<http://webserver2.tecgraf.puc-rio.br/~scuri/download/fid.pdf>> acesso em: 12 dez. 2014.

SHARMA, Garima; THÉ, Andy. **Automating Image Registration with MATLAB**. MathWorks, 2013. Disponível em: <[http://www.mathworks.com/tagteam/76973\\_92121v00\\_automating-image-registration-with-matlab.pdf](http://www.mathworks.com/tagteam/76973_92121v00_automating-image-registration-with-matlab.pdf)> acesso em: 10 jun. 2015.

SILVA, Antônio Machado e. **Curso Processamento digital de imagens de satélite**. Centro de Eventos da PUCRS - outubro de 2001. Porto Alegre - RS. Disponível em: <[www.cartografia.org.br](http://www.cartografia.org.br)> acesso em: 12 jun. 2015.

SOARES, Fabrízio A. A. M. N. **Processamento digital de imagens**. Universidade Federal de Goiás, Instituto de Informática, 2012. Disponível em: <<http://www.inf.ufg.br/~fabrizzio/mestrado/pdi/aulas/aula2.pdf>> acesso em: 26 set. 2014.

SPRING. **Integrating remote sensing and GIS by object-oriented data modelling**. Camera G, Souza RCM, FreitasUM, Garrido J Computers & Graphics, 20: (3) 395-403, May-Jun 1996.