

Relatório de Trabalho 2 abnTeX2

Equipe abnTeX2*

Lauro César Araujo[†]

2014, v-1.9.2

Resumo

Conforme a ABNT NBR 6022:2003, o resumo é elemento obrigatório, constituído de uma sequência de frases concisas e objetivas e não de uma simples enumeração de tópicos, não ultrapassando 250 palavras, seguido, logo abaixo, das palavras representativas do conteúdo do trabalho, isto é, palavras-chave e/ou descritores, conforme a NBR 6028. (...) As palavras-chave devem figurar logo abaixo do resumo, antecedidas da expressão Palavras-chave:, separadas entre si por ponto e finalizadas também por ponto.

Palavras-chaves: latex. abntex. editoração de texto.

Introdução

1 Translação, Rotação e Redimensionamento

Translação, Rotação e Redimensionamento são transformações lineares que podem ser aplicadas a um ponto, linha ou qualquer forma representável por um hiperplano. A forma original do objeto é comumente chamada de pré imagem e após as transformações realizadas na forma ou no posicionamento do objeto é simplesmente chamada de imagem. Os tipos de transformações que foram supracitados possuem uma característica entre si. Não importa a ordem de aplicação ou mesmo a intensidade dela, sempre será possível recuperar a forma original a partir da combinação das mesmas três transformações. Os códigos fonte utilizados nesta sessão encontram-se no Apêndice A.

1.1 Translação

A operação de translação move cada ponto da forma em um fator constante em uma direção específica. A forma do objeto não sofre qualquer alteração com essa transformação, apenas o posicionamento no espaço. A translação pode ser vista e interpretada como a adição de vetores constantes a cada ponto da forma ou como o deslocamento do centro de coordenadas. A ideia de se transladar um ponto de coordenadas (x, y) é calcular a nova posição (x', y') onde $(x', y') = (x + x_0, y + y_0)$ onde x_0 e y_0 é a quantidade transladada em cada eixo da coordenada ou simplesmente a nova coordenada da origem. Apesar de

* <<http://abntex2.googlecode.com/>>

[†]laurocesar@laurocesar.com

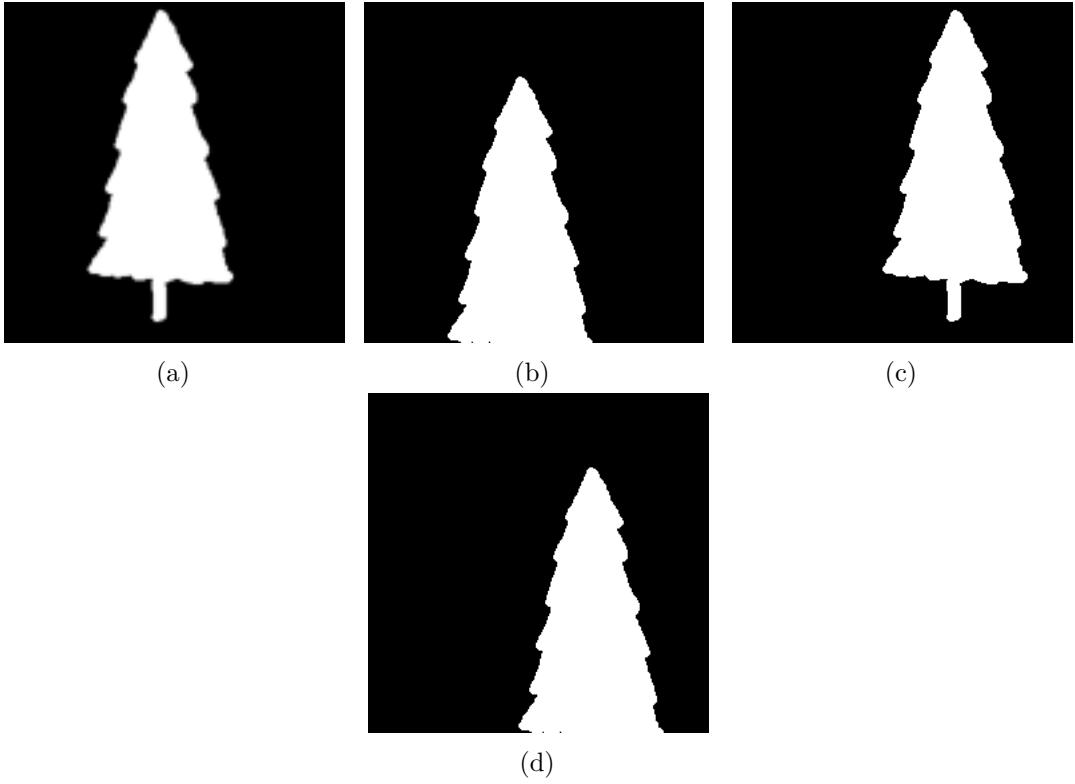


Figura 1 – [1a](#). Image original [1b](#). Imagem com translação vertical de 50 pixels [1c](#). Imagem com translação horizontal de 50 pixels [1d](#). Imagem com translação vertical e horizontal de 50 pixels

ser exibido a translação apenas para duas coordenadas esta operação pode ser feita em qualquer número de dimensões. A translação pode ser expressa de forma matricial. Abaixo temos a representação para a translação de uma forma geométrica em um espaço de três dimensões. De forma generalizada, para qualquer número de dimensões, podemos pensar na matriz de transformação em que todos os elementos da diagonal principal possuem o valor 1 e a última coluna com os valores da nova origem de coordenadas.

$$Im' = T * Im \quad (1)$$

$$Im' = \begin{bmatrix} 1 & 0 & 0 & x_0 \\ 0 & 1 & 0 & y_0 \\ 0 & 0 & 1 & z_0 \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \quad (2)$$

Na figura 1 temos o resultado da translação na imagem.

1.2 Rotação

A operação de rotação é uma transformação geométrica que mapeia a posição de uma forma em um espaço levando em consideração o ângulo desejado na transformação. Todos os pontos da forma são rotacionados a partir de um ângulo constante em relação ao ponto de rotação, para rotação no \mathbb{R}^2 , reta no \mathbb{R}^3 ou hiperplano para o R^n . No R^2 os

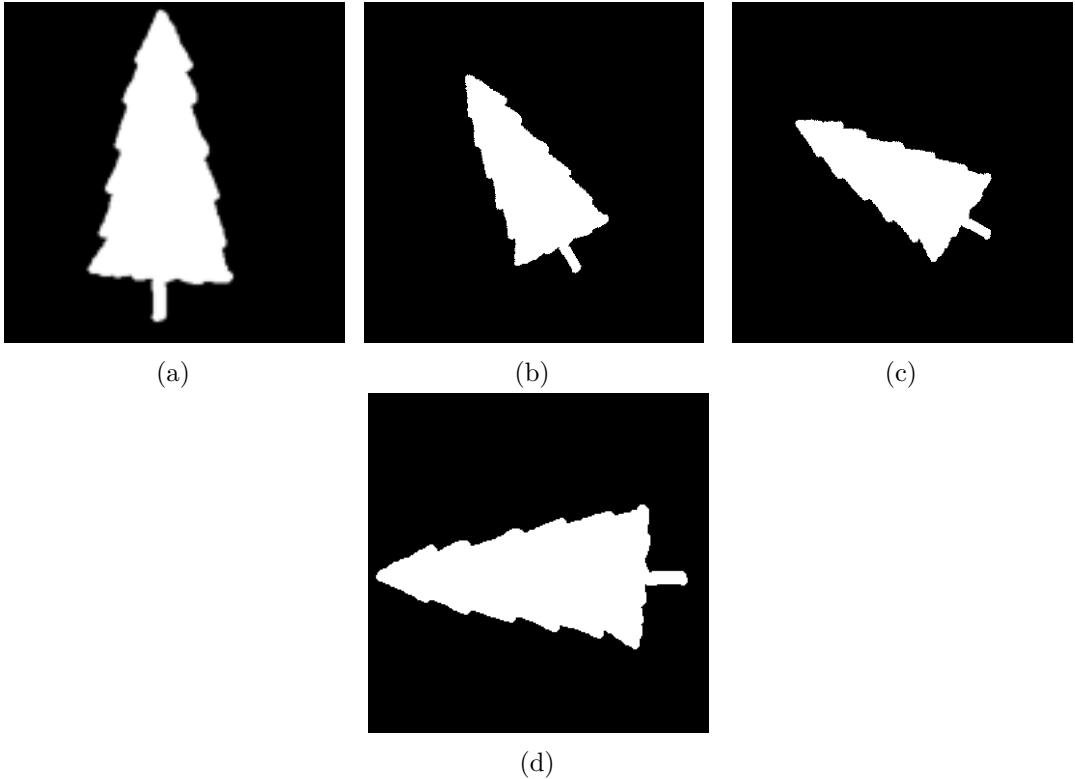


Figura 2 – [2a](#). Image original [2b](#). Imagem com rotação de 30° [2c](#). Imagem com rotação de 60° [2d](#). Imagem com rotação de 90°

pontos de destino da forma podem ser calculados através das seguintes equações:

$$x' = \cos(\theta) * (x_1 - x_0) - \sin(\theta) * (y_1 - y_0) + x_0 \quad (3)$$

$$y' = \sin(\theta) * (x_1 - x_0) + \cos(\theta) * (y_1 - y_0) + y_0 \quad (4)$$

(5)

ou em termos matriciais.

$$Im' = T * Im \quad (6)$$

$$Im' = \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix} * \begin{bmatrix} x \\ y \end{bmatrix} \quad (7)$$

Ao fazer uso da forma matricial deve-se perceber que a rotação ocorrerá somente sobre a origem. Para contornar esta característica faz-se necessário o uso do operador de translação, deslocando assim o ponto desejado para o centro das coordenadas para só então realizar a rotação. Na figura [2](#) podemos ver o resultado da rotação.

1.3 Redimensionamento

O redimensionamento é uma transformação linear que aumenta ou diminui o tamanho de uma forma em um fator constante em todas as direções da forma. Normalmente este aumento ou redução é feita de forma uniforme, mas pode ser feita de forma diferente em cada eixo da representação da forma. Um fator de redimensionamento maior que 1 reflete em um aumento na imagem, menor que 1 em uma redução e um fator igual a

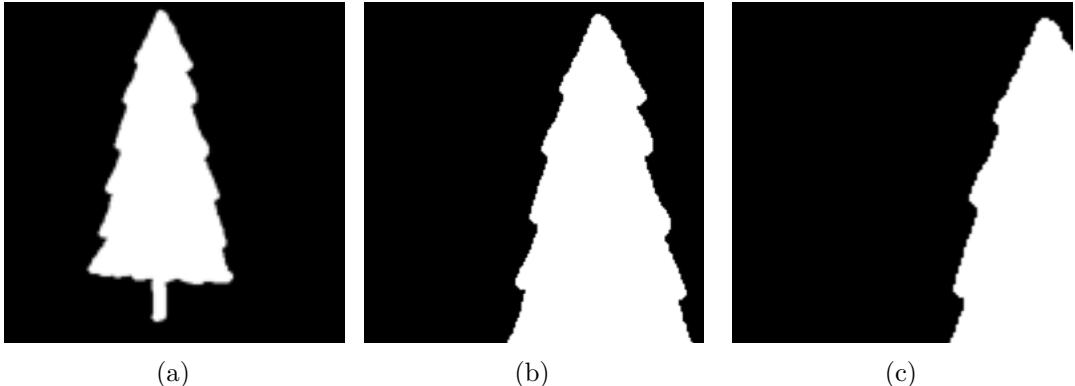


Figura 3 – **3a.** Image original **3b.** Redimensionamento com fator 1.5 **3c.** Redimensionamento com fator 2

1 não é feita nenhuma transformação. É importante ressaltar que valores menor que 0 não são válidos. Bem como as operações de translação e rotação o redimensionamento também pode ser representado por um produto matricial. O redimensionamento de um vetor $\vec{v} = (x, y, z)$ por um fator f resulta em um vetor $\vec{w} = (fx, fy, fz)$, representado matricialmente por:

$$Im' = T * Im \quad (8)$$

$$Im' = \begin{bmatrix} f_x & 0 & 0 \\ 0 & f_y & 0 \\ 0 & 0 & f_z \end{bmatrix} * \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad (9)$$

f_x, f_y, f_z são os fatores de redimensionamento para cada dimensão do vetor.

2 Contraste, Brilho e Gama

As técnicas de modificação de histograma são conhecidas como técnicas ponto-a-ponto, uma vez que o valor de tom de cinza de um certo pixel após o processamento depende apenas de seu valor original. Dentre as principais técnicas temos o ajuste de brilho, contraste e gama.

2.1 Brilho

O Brilho é um atributo de percepção visual no qual determina a intensidade de energia emitida ou refletida por uma fonte. O brilho é uma característica importante pois é ela que determina se a quantidade de luz é perceptível pelos sensores ou pela aplicação. O ajuste de intensidade do brilho é feito a partir da seguinte equação:

$$Im'(x, y) = Im(x, y) + b \quad (10)$$

onde b é o valor de ajuste do brilho.

2.2 Contraste

O contraste é um atributo que permite que objetos sejam discriminados dentro de um mesmo campo visual. Dois objetos com intensidades semelhantes são dificilmente

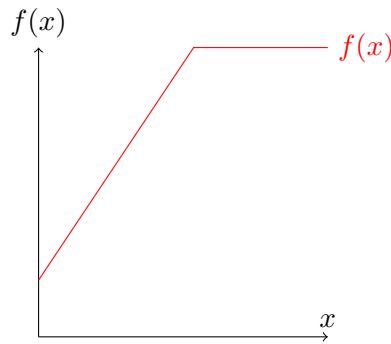


Figura 4 – Função de mapeamento com brilho=50 e contraste=1.5

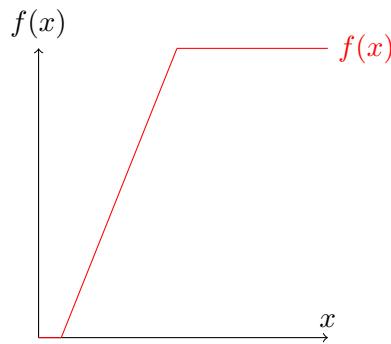


Figura 5 – Função de mapeamento com brilho=-50 e contraste=2.5

distinguidos por sensores como o olho humano. O ajuste no contraste tem como princípio aumentar a diferença de intensidade entre os objetos.

$$Im'(x, y) = c * Im(x, y) \quad (11)$$

onde c é o valor de ajuste do contraste.

2.3 Gama

O gama, assim como o ajuste de contraste é utilizado para aumentar a diferença de intensidade entre os objetos porem de forma não linear. A regulação do gama pode ser feita com base na seguinte função.

$$Im'(x, y) = c * Im(x, y)^\gamma \quad (12)$$

onde c é um fator de correção do gama e γ é o fator de gama. O fator de correção é importante pois geralmente o função descrita acima retorna valores muito altos.

2.4 Resultado Obtidos

Na figura 8 temos os resultados obtidos após a realização de uma série de ajustes na figura 7. Podemos perceber através dos histogramas que a alteração do contraste

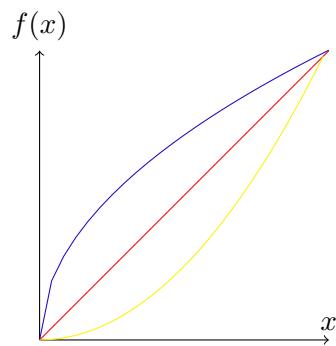


Figura 6 – Função de mapeamento do gama. vermelho, azul e amarelo possuem 1, 0,5 e 2 valores de gama respectivamente



Figura 7 – Imagem utilizada como teste para as operações de brilho, contraste e gama

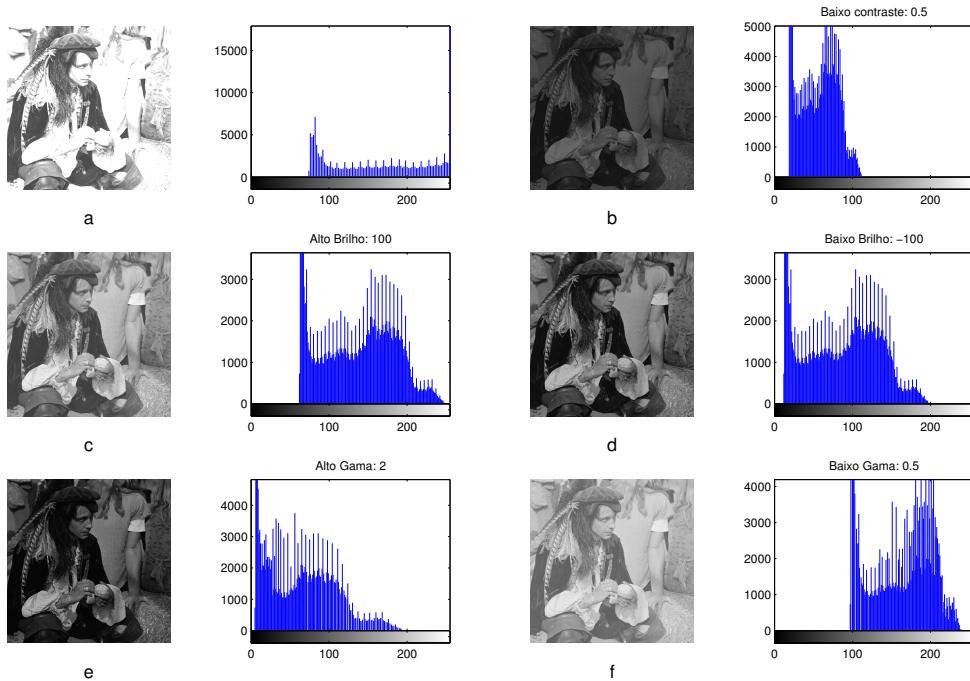


Figura 8 – Resultado da aplicação das técnicas de ajuste de histograma. a)Contraste aplicado 2 b)Contraste aplicado 0.5 c)Brilho aplicado 25 d)Brilho aplicado -25 e)Gama aplicado 2 f)Gama aplicado 0.5

a,b, faz com que o hisograma seja alongado ou comprimido ao longo dos possíveis valores da imagem. Diferentemente do ajuste de contraste, o brilho apenas desloca o histograma, figura 8 c,d. Caso o valor do brilho seja positivo o histograma é deslocado para a região mais clara, caso negativo para a região mais escura. O ajuste de gama,figura 8 e,f, é um pouco mais peculiar que os dois anteriores. Por ser uma operação exponencial, caso o valor seja maior que 1 a imagem como um todo terá a intensidade de seus pixels reduzidas, porém os pixels com valores de intensidade menor serão ainda mais afetados pelo operador. caso o valor seja menor que 1 a intensidade da imagem aumentará e da mesma forma os pixels com valores de intensidade menor sofrerão maiores mudanças. O código fonte deste exemplo pode ser visualizado no Apêndice B

3 Filtragem no domínio da frequência

A transformada de Fourier é uma transformação matemática que permite que funções possam ser expressas em componentes senoidais. O somatório destas componentes tem como resultado a função original. A informação de frequencia destas componentes são muito importantes tendo em vista que com elas podemos identificar e filtrar ruídos que atuam em frequencias conhecidas. Também pode-se fazer a filtragem de componentes de maiores ou menores frequencias a partir de um limiar, filtros passa-baixa e filtros passa-alta respectivamente.

Para um sinal de uma dimensão temos a seguinte equação utilizada para calcular a transformada de fourier de $s(t)$:

$$S(f) = \int_{-\infty}^{\infty} s(t) \cdot e^{-i2\pi ft} dt. \quad (13)$$

Para duas dimensões temos a seguinte equação:

$$\hat{f}(\xi_x, \xi_y) = \iint f(x, y) e^{-2\pi i(\xi_x x + \xi_y y)} dx dy \quad (14)$$

ξ_x e ξ_y são as frequências calculadas sobre a imagem. À medida que nos aproximamos da origem da transformada, imagem calculada a partir da equação anterior, as baixas frequências correspondem aos componentes de intensidade de variação lenta em uma imagem como mudanças suaves de intensidade na parede ou em outras regiões uniformes da imagem. À medida que nos distanciamos da origem da transformada, as frequências mais altas começam a corresponder a variações de intensidade cada vez mais rápidas como bordas de objetos e outros elementos, figura 9. De acordo com as características citadas anteriormente podemos construir as matrizes para a filtragem no domínio da frequência, figura 13. Uma das características do domínio da frequência é que a filtragem espacial pode ser realizada utilizando apenas um produto de matrizes. O resultado da filtragem pode ser visualizado na figura. O código fonte utilizado encontra-se no Apêndice C.

4 Segmentação Baseada em cores

4.1 Espaço de cores RGB

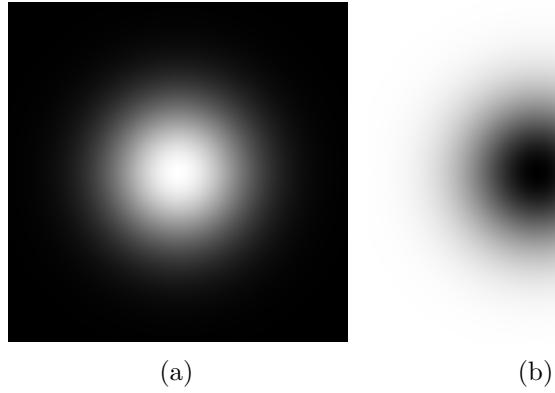
O espaço de cores RGB é formado por todas as combinações possíveis das cores R(Red), G(Green) e B(Blue), vermelho, verde e azul respectivamente. Esse espaço de cores é normalmente representado por um cubo onde cada eixo representa uma das componentes, ou



(a)

(b)

Figura 9 – 9a. Image original 9b. Espectro de Fourier da imagem.



(a)

(b)

Figura 10 – 11c. Filtro passa baixa. 11d. Filtro passa alta.

canais, de cor, a intensidade de cada componente indica a cor resultante, figura 13a. Em imagens coloridas baseadas nesta representação o valor máximo de cada componente é 255. Na figura 12 temos o resultado de algumas combinações deste espaço de cores. Neste espaço de cores, vale ressaltar que a cor branca é formada pela combinação da máxima intensidade de todas as três componentes, a cor preta pela combinação da intensidade mínima das três componentes.

4.2 Espaço de cores HSV

O espaço de cores HSV também conhecido como HSB, assim como o RGB possue 3 componentes para representar as cores. As três componentes estão organizadas de forma que suas componentes armazenem informação da matiz, saturação e brilho separadamente. O lugar geométrico deste espaço de cores é um cilindro, as vezes também representado por um cone, onde a altura é o nível de brilho, o raio a intensidade da cor e o ângulo a cor no espectro luminoso, matiz. A matiz, componente H, descreve a cor encontrada no espectro eletromagnético, é dessa componente que descrevemos as cores eg. vermelho, verde, laranja. A saturação, componente S, descreve a pureza da cor em relação a cor branca. Uma cor totalmente impura, valor mínimo desta componente, tem como cor resultante a cor branca. A última componente V ou B, o brilho, descreve quanta luz é emitida do objeto. A conversão entre os espaços de cores HSV e RGB pode ser feita facilmente com os seguintes comandos no Matlab.

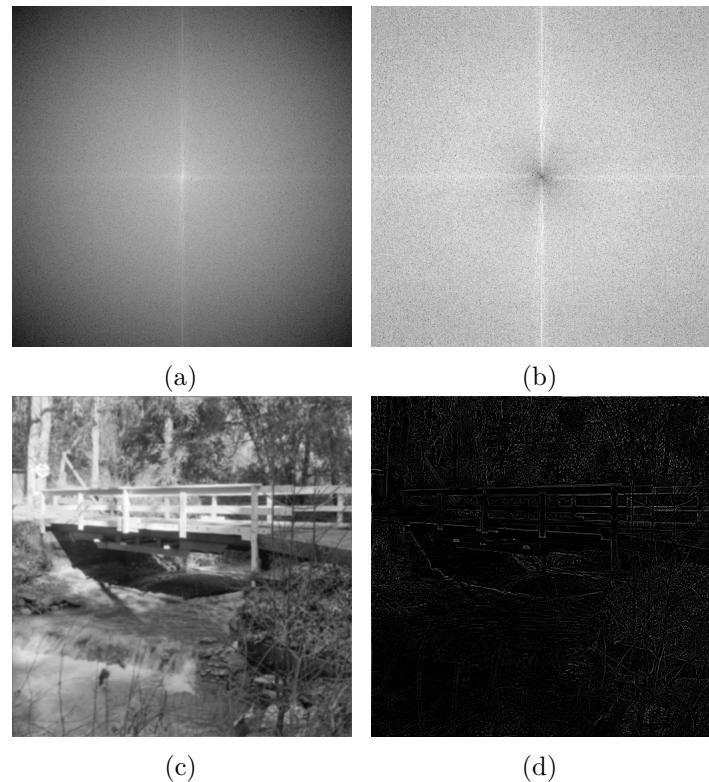


Figura 11 – [11c](#). Filtro passa baixa. [11d](#). Filtro passa alta.

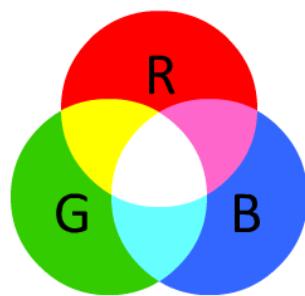


Figura 12 – Algumas combinações possíveis do espaço de cores RGB. Perceba que na interseção das cores são produzidas cores diferentes.

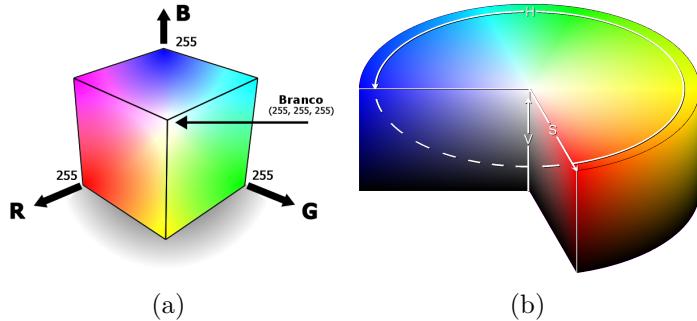


Figura 13 – 13a. Representação do espaço de cores RGB 13b. Representação do espaço de cores HSV.

```

1 % Converter RGB para HSV
2 hsv_image = rgb2hsv(rgb_image)
3 % Converter HSV para RGB
4 rgb_image = hsv2rgb(hsv_image)

```

No Matlab, por padrão as imagens coloridas são alocadas em matrizes onde cada canal representa uma das componentes do RGB. Cada elemento da matriz representa uma componente de um pixel da imagem com valores variando entre [0, 255]. Após a conversão da imagem para o formato HSV as matrizes de cada componente têm seus valores variando entre [0, 1].

4.3 Segmentação baseada na cor da pele

A definição do limiar de segmentação baseado em cores é feita utilizando o espaço de cores HSV. Como dito anteriormente este espaço de cores possui como característica a separação da informação de cor de outras informações como luminosidade e pureza. A figura 15a foi utilizada para a definição dos limiares. Foi verificado que os valores H que melhores descrevem os tons da pele estão no intervalo [0 , 0.1], intervalo este que representa 10% da faixa de valores possíveis do canal H. Foi verificado também que para o canal S os valores que melhor representam o tom da pele estão na faixa de [0.4 1 0.8]. O resultado da combinação dos dois limiares em S e H com a imagem de entrada pode ser visualizado na imagem 15d. O código fonte utilizado encontra-se no Apêndice D.

4.4 Rastreamento baseado em cor

Considerações finais

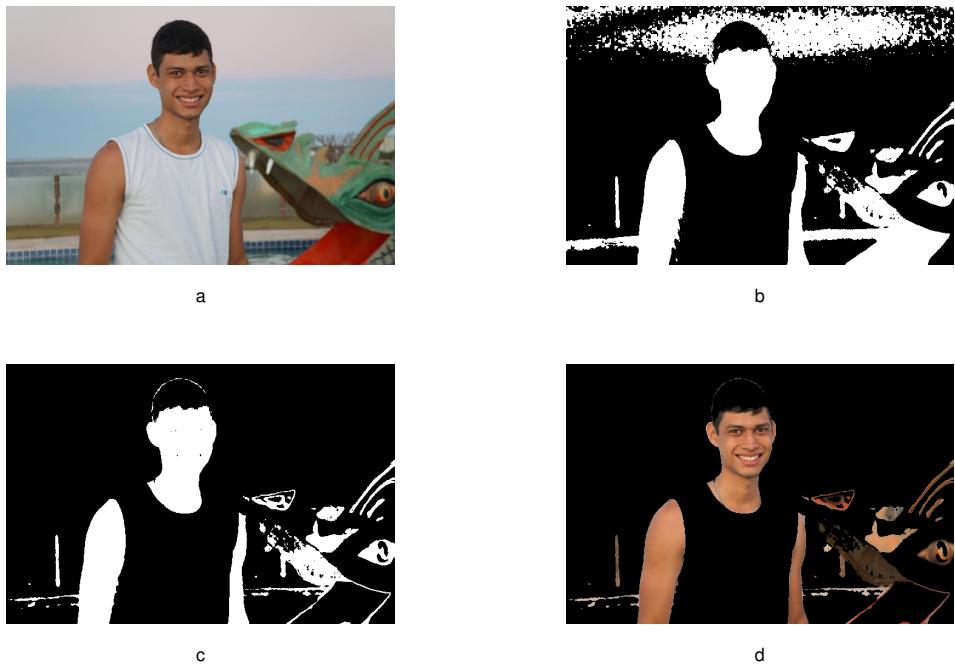


Figura 14 – Resultado da aplicação das técnicas de segmentação. a)Imagem Original b)Filtro aplicado ao canal H. c)Filtro aplicado ao canal H e S d)Imagen resultante após os filtros.



Figura 15 – Resultado da aplicação das técnicas de segmentação. a)Imagem Original b)Filtro aplicado ao canal H. c)Filtro aplicado ao canal H e S d)Imagen resultante após os filtros.

Referências

APÊNDICE A – Código fonte: Translação, Rotação e Redimensionamento

```
1 clear;
2 clc;
3 image = imread('../bd/tree.gif');
4 [m n] = size(image);
5 imshow(image);
6
7 %%
8 %Operao de rotao
9 image30 = imrotate(image,30);
10 image60 = imrotate(image,60);
11 image90 = imrotate(image,90);
12
13
14 imwrite(image30,'image30.png');
15 imwrite(image60,'image60.png');
16 imwrite(image90,'image90.png');
17
18 %%
19 %Operao de redimensionamento
20 imageScale1 = imresize(image, 0.5);
21 imageScale2 = imresize(image, 1.5);
22 imageScale3 = imresize(image, 2);
23
24 imwrite(imageScale1,'imageScale1.png');
25 imwrite(imageScale2(1:m,1:n),'imageScale2.png');
26 imwrite(imageScale3(1:m,1:n),'imageScale3.png');
27
28 %%
29 %Operao de translao
30 imageT1 = imtranslate( image, [50, 0] );
31 imageT2 = imtranslate( image, [0 50] );
32 imageT3 = imtranslate( image, [50 50] );
33
34 imwrite(imageT1,'imageT1.png');
35 imwrite(imageT2,'imageT2.png');
36 imwrite(imageT3,'imageT3.png');
```

APÊNDICE B – Código fonte: Ajuste de Brilho, Contraste e Gama

```
1 function res = ajustaBrilho(image, b)
2     res = image + b;
3 end
```

```
1 function res = ajustaContraste(image, c)
2     res = image*c;
3 end
```

```
1
2 function res = ajustaGama(image,g)
3     res = uint8(255*((double(image)/255).^g));
4 end
```

```
1 close all;clear;
2 %Carregando e exibindo Imagem Original
3 image = imread('../bd/pirate.tif');
4 figure
5 subplot(1,2,1);
6 imshow(image);
7 subplot(1,2,2);
8 imhist(image);
9
10 figure;
11 res = ajustaContraste(image,2);
12 subplot(3,4,1);
13 imshow(res);
14 xlabel('a', 'FontSize', 15)
15 subplot(3,4,2);
16 imhist(res);
17 imwrite(res, 'altoContraste.png');
18
19
20 res = ajustaContraste(image,0.5);
21 subplot(3,4,3);
22 imshow(res);
23 xlabel('b', 'FontSize', 15)
24 subplot(3,4,4);
25 imhist(res);
26 title('Baixo contraste: 0.5')
27 imwrite(res, 'baixoContraste.png');
28
29
30
31 res = ajustaBrilho(image,25);
32 subplot(3,4,5);
33 imshow(res);
34 xlabel('c', 'FontSize', 15)
35 subplot(3,4,6);
36 imhist(res);
37 title('Alto Brilho: 100')
38 imwrite(res, 'altoBrilho.png');
39
40
41 res = ajustaBrilho(image,-25);
42 subplot(3,4,7);
43 imshow(res);
```

```

44 xlabel('d', 'FontSize', 15)
45 subplot(3,4,8);
46 imhist(res);
47 title('Baixo Brilho: -100')
48 imwrite(res, 'baixoBrilho.png');
49
50 res = ajustaGama(image,2);
51 subplot(3,4,9);
52 imshow(res);
53 xlabel('e', 'FontSize', 15)
54 subplot(3,4,10);
55 imhist(res);
56 title('Alto Gama: 2')
57 imwrite(res, 'altoGama.png');
58
59 res = ajustaGama(image,0.5);
60 subplot(3,4,11);
61 imshow(res);
62 xlabel('f', 'FontSize', 15)
63 subplot(3,4,12);
64 imhist(res);
65 title('Baixo Gama: 0.5')
66 imwrite(res, 'baixoGama.png');

```

APÊNDICE C – Código fonte: Filtragem no domínio da frequência

```

1 close all;clear
2
3 %Carregando imagem
4 image = imread('../bd/walkbridge.tif');
5 image = image(:,:,1);
6
7 subplot(4,2,1);
8 imshow(image);
9 title('Original');
10
11 %Aplicando transformada de Fourier
12 spec = fft2(image);
13
14 %Exibindo Spectro de Fourier
15 ;
16 subplot(4,2,2)
17 surf(log(abs(spec)), 'EdgeColor', 'None')
18 imwrite(mat2gray(log(abs(spec))), 'espectroOriginal.png')
19 title('Espectro');
20
21
22 %% Passa Baixa
23
24 var = 3;
25 d = 20/length(spec(1,:));
26
27 %Calculando filtro gaussiano
28 filtro = gaussmf(-10:d:10,[var 0])' * gaussmf(-10:d:10,[var 0]);
29 filtro = filtro( 1:length(spec(1,:)), 1:length(spec(1,:)));
30 imwrite(mat2gray(filtro), 'passaBaixa.png');

```

```

31 %filtro = double(filtro > 0.5);
32 ;
33 subplot(4,2,3)
34 surf(filtro,'EdgeColor','None')
35 title('Filtro Passa Baixa');
36
37 %Filtrando
38 filtrada = filtro .* fftshift(spec);
39 ;
40 subplot(4,2,5)
41 surf(log(abs(filtrada)), 'EdgeColor', 'None')
42 imwrite(mat2gray(log(abs(filtrada))), 'filtradaPBspec.png');
43 title('Filtrada Baixa');
44
45 %Exibindo imagem resultante
46 espaco = uint8(ifft2(fftshift(filtrada)));
47
48 imwrite(espaco, 'filtradaPB.png')
49 subplot(4,2,7)
50 imshow(espaco);
51 title('Resultado Passa Baixa');
52
53
54
55
56
57
58
59 %% Passa Alta
60
61 var = 3;
62 d = 20/length(spec(1,:));
63
64 %Calculando filtro gaussiano invertido
65 filtro = gaussmf(-10:d:10,[var 0])' * gaussmf(-10:d:10,[var 0]);
66 filtro = filtro( 1:length(spec(1,:)), 1:length(spec(1,:)));
67 filtro = 1 - filtro;
68 imwrite(mat2gray(filtro), 'passaAlta.png');
69
70 %filtro = double(filtro < 0.5);
71 ;
72 subplot(4,2,4)
73 surf(filtro,'EdgeColor','None')
74 title('Filtro Passa Alta');
75
76 %Filtrando
77 filtrada = filtro .* fftshift(spec);
78 ;
79 imwrite(mat2gray(log(abs(filtrada+1))), 'filtradaPASpec.png');
80 subplot(4,2,6)
81 surf(log(abs(filtrada)), 'EdgeColor', 'None')
82 title('Filtrada Passa Alta');
83
84 %Exibindo imagem resultante
85 espaco = mat2gray(uint8(ifft2(fftshift(filtrada))));
```

```

91 | imwrite(espaco,'filtradaPA.png')
92 | subplot(4,2,8)
93 | imshow(espaco);
94 | title('Resultado Passa Alta');

```

APÊNDICE D – Código fonte: Segmentação de Pele

```

1 close all;
2 clc;
3 %Carregando imagem
4 image = imread('../bd/eu.JPG');
5
6 subplot(2,2,1)
7 imshow(image)
8 xlabel('a', 'FontSize', 15)
9
10 %Separando canais HSV
11 hsv_image = rgb2hsv(image) ;
12 h = hsv_image(:,:,1);
13 s = hsv_image(:,:,2);
14 v = hsv_image(:,:,3);
15
16 %Filtrando tons da pele
17 h = h<0.1 & h>0;
18 s = s> 0.4 & s < 0.8;
19 skinMask = double(s & h);
20
21 %Canal H filtrado
22 subplot(2,2,2)
23 imshow(h)
24 xlabel('b', 'FontSize', 15)
25
26 %Canal S filtrado combinado com H filtrado
27 subplot(2,2,3)
28 imshow(skinMask)
29 xlabel('c', 'FontSize', 15)
30
31 %Imagem resultante
32 res = []
33 res(:,:1) = double(image(:,:1)) .* skinMask;
34 res(:,:2) = double(image(:,:2)) .* skinMask;
35 res(:,:3) = double(image(:,:3)) .* skinMask;
36
37 subplot(2,2,4)
38 imshow(mat2gray(res))
39 xlabel('d', 'FontSize', 15)

```