

Multimédia e Novos Serviços

Lab 3

João Seixas - up201505648

José Pedro Borges - up201503603

Vicente Espinha - up201503764

**Projeto realizado no âmbito do
Mestrado Integrado em Engenharia Informática e
Computação**

27/03/2018

Índice

Índice	1
1	2
1.1 - Segmentação Simples	2
1.2 - Segmentação Alternativa	8
2 - Adicionar objetos a outras imagens	13

1

1.1 - Segmentação Simples

```
function basic_segment (inputImage)

%imports a coloured image with a blue background and presents
that image on the
%screen
%reads image and converts to rgb if it is in greyscale
image=imread(inputImage);
if size(image,3) ~= 3
    image = cat(3,image,image,image);
end

% get image dimensions: an RGB image has three planes
% reshaping puts the RGB layers next to each other generating
% a two dimensional grayscale image

[height, width, planes] = size(image); %obtain the dimensions of the matrix
%rgb = reshape(image, height, width * planes);

figure(1),imshow(image),title('imagem original');

%separates each RGB component in a different matrix and visualises each one
on the
%screen

%imagesc(rgb);           % visualize RGB planes
%colorbar on             % display colorbar

r = image(:, :, 1);      % red channel
g = image(:, :, 2);      % green channel
b = image(:, :, 3);      % blue channel
```

```

figure(2),imshow(r),title('red');
figure(3),imshow(g),title('green');
figure(4),imshow(b),title('blue');

% write here the code to plot the histogram
figure(5),imhist(b), title(' B channel histogram');
threshold=input('Which threshold?'); % ask the user for a threshold

%threshold = 100;          % threshold value
%imagesc(b < threshold);    % display the binarized image

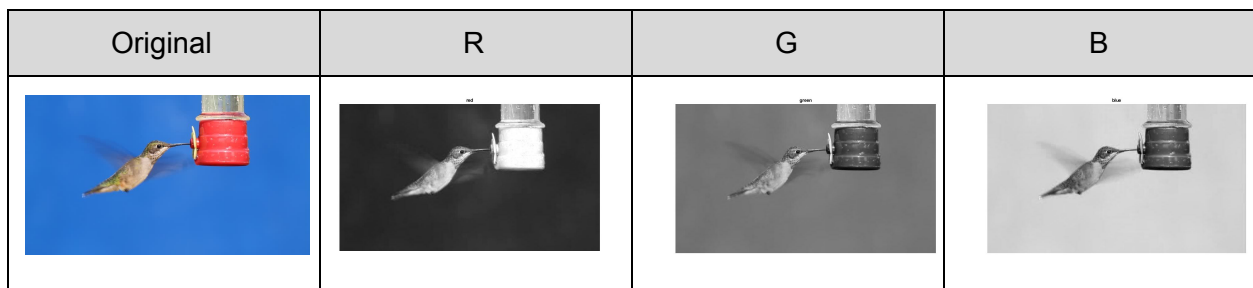
% write here the code to generate a new matrix with dimensions "height" and
% "width" with white pixels in the locations corresponding to the locations
in the B matrix with low values
c=zeros(height, width);
for i=1:height
    for j=1:width
        if(b(i,j)<threshold)
            c(i,j)=255;
        end
    end
end
figure(6), imshow(c),title('Blue Threshold');

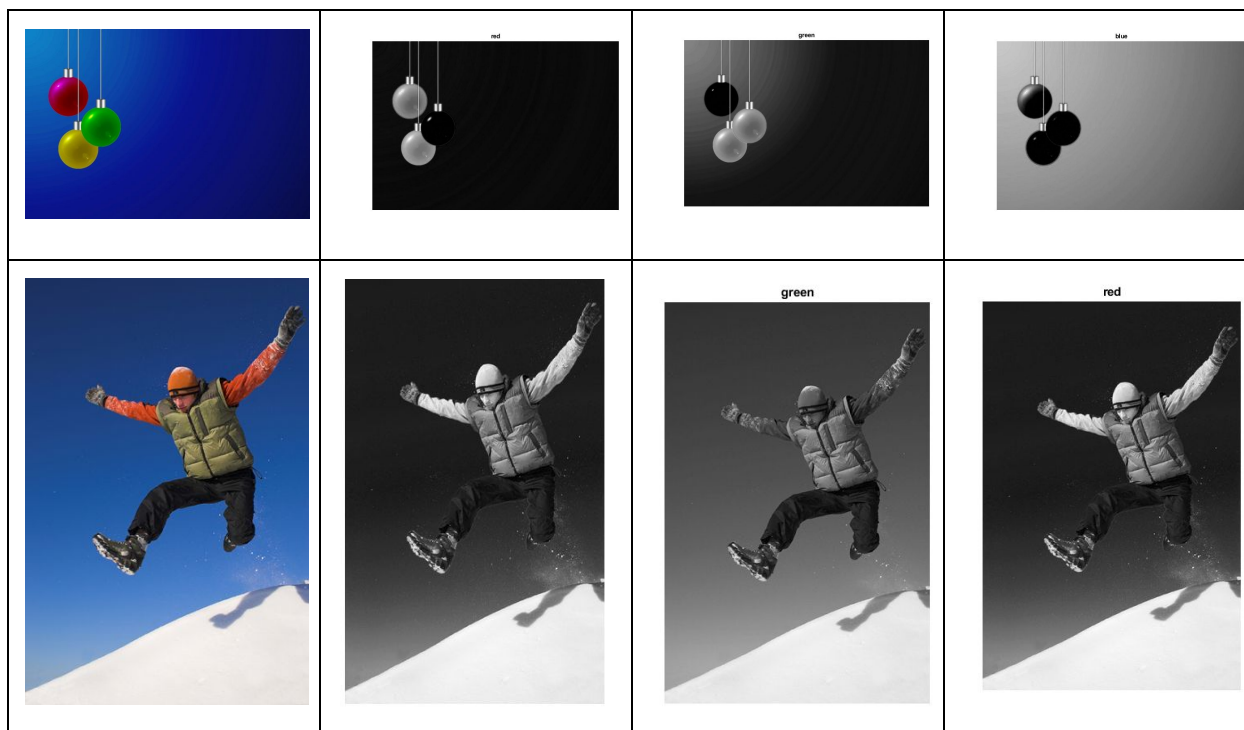
```

Para esta pergunta foi pedido aos alunos que adaptassem um *script* fornecido para que:

- Recebesse como argumento uma imagem com fundo azul;
- Separasse as componentes RGB da imagem em três matrizes diferentes;
- Mostrasse um histograma da componente B da imagem;
- Separasse os pixels que possuem componente B abaixo de um certo threshold dos que possuem componente B acima do threshold para uma nova matriz.

No final, este *script* deverá mostrar as imagens das componentes R, G e B separadamente, junto com a imagem a preto e branco. Em baixo, são apresentados alguns resultados obtidos.





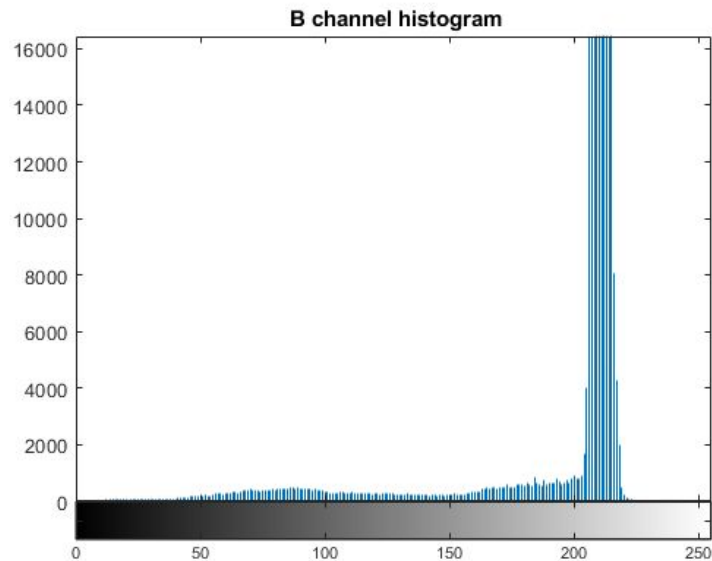
Nas imagens obtidas, as partes mais claras da imagem correspondem às partes em que a componente principal tem um valor elevado na imagem original. Por exemplo, na imagem R da *christmasBB* nota-se que as duas bolas mais à esquerda são as que se apresentam mais claras, devido a serem essas as que possuem um valor R maior.

Assim, podemos afirmar que as partes com maior valor da componente B, para além do fundo, nas respectivas imagens são as seguintes:

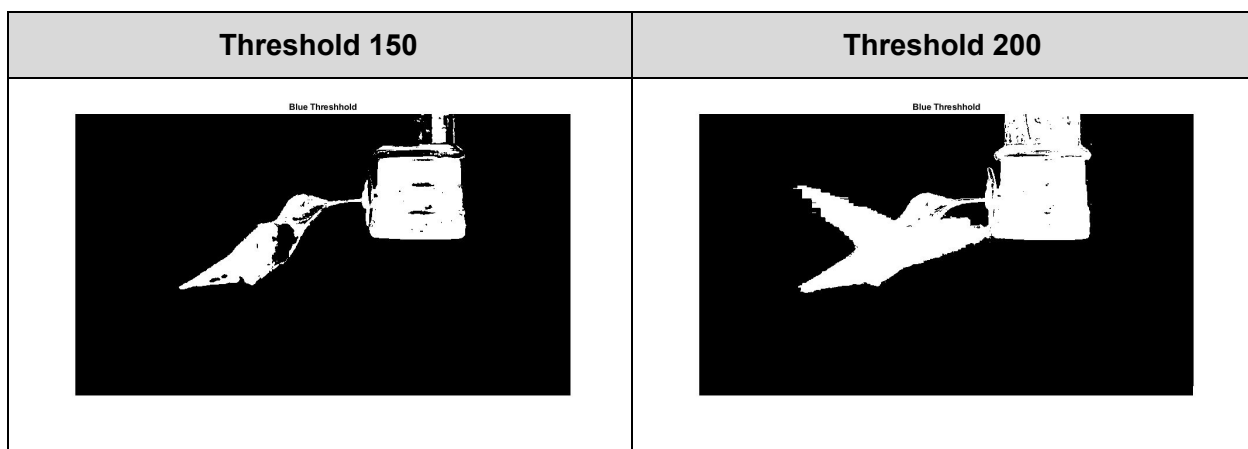
- **birdBB**: Corpo do Pássaro;
- **christmasBB**: Reflexo nas bolas de Natal;
- **jumpBB**: Monte de neve.

Apesar de estes componentes não serem completamente azuis, são partes na imagem que são muito claras ou brancas, o que pode levar a que estas componentes sejam removidas erradamente na fase seguinte do *script*.

Ao correr o *script* com a imagem do pássaro é obtido o seguinte histograma:



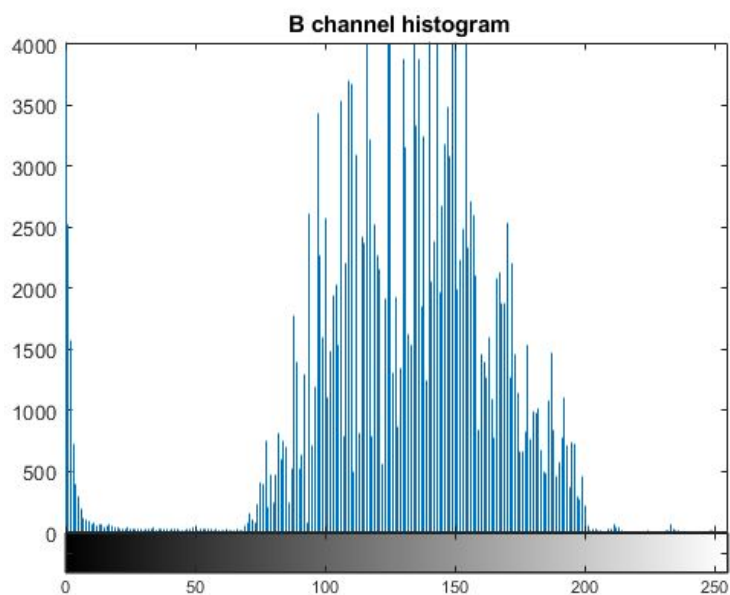
Com base nisto, podemos concluir que a zona com maior densidade de azuis se encontra a rondar os 200 . Assim, decidimos fazer duas experiências diferentes e analisar os seus resultados. Para uma das experiências o threshold escolhido foi de 150 e para outra foi de 200.



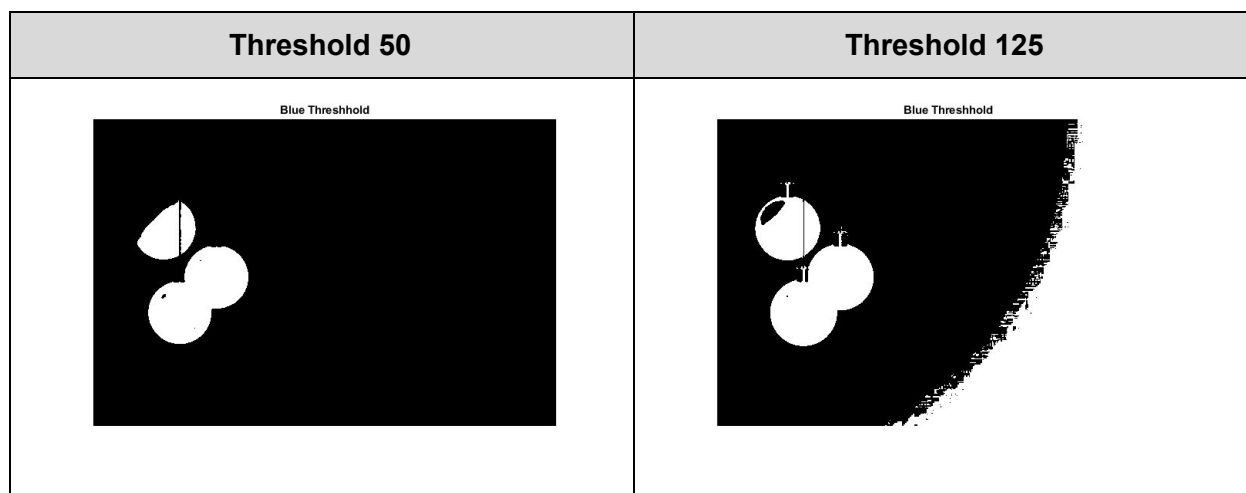
Como seria de esperar, em ambas as imagens o fundo azul foi substituído por um fundo preto, visto que estes valores se encontram acima dos thresholds escolhidos. Podemos também notar algumas diferenças entre as imagens. A primeira imagem, tendo um threshold mais baixo que o imediatamente antes da zona mais densa, acaba por captar alguns valores a mais como sendo azuis. Por causa disto, as azas e certas zonas do corpo do pássaro são assinaladas como sendo acima do threshold e são desenhadas a preto na imagem resultante. O mesmo não acontece na segunda imagem, visto que o threshold já não apanha os valores previamente mencionados. Nesta imagem tanto as asas como a maior parte do pássaro já são detetadas

como sendo abaixo do threshold e desenhadas a branco. Este resultado já é mais próximo do pretendido.

Relativamente à imagem das bolas de Natal, foi obtido o seguinte histograma:



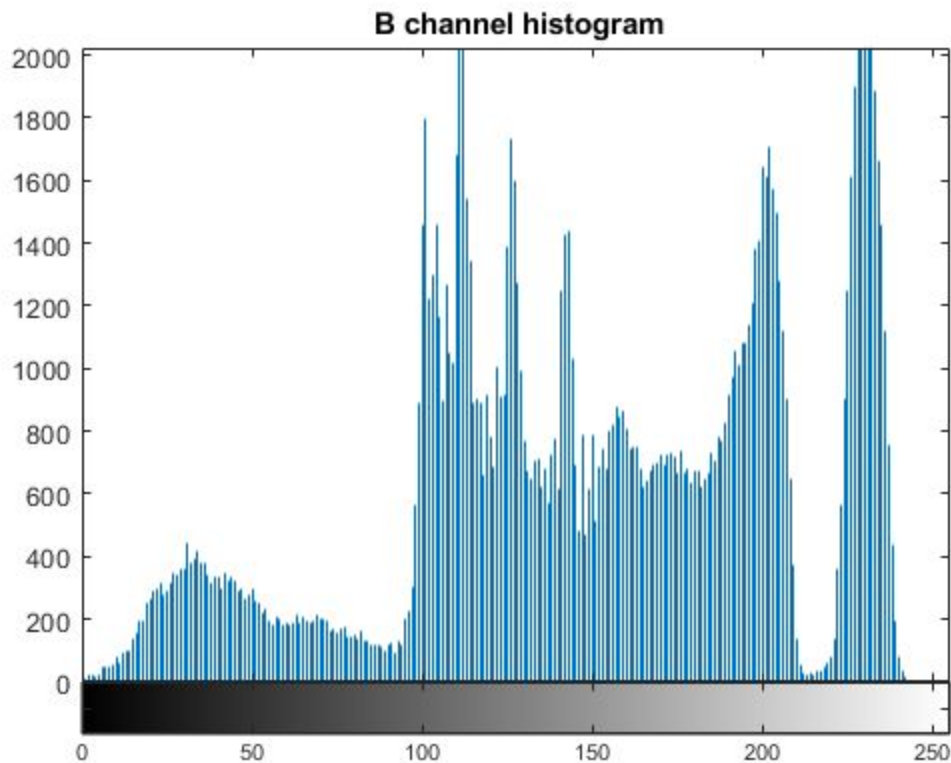
Neste histograma observa-se que a zona de maior densidade de azuis se encontra, aproximadamente, entre os 75 e os 200 Hz. Assim, foi escolhido um threshold de 50 por se encontrar antes da zona de maior densidade. Foi também escolhido um threshold de 125 por ser um valor em que a imagem obtida é a mais próxima do resultado pretendido.



Relativamente à imagem da esquerda, podemos observar que o threshold escolhido faz com que a bola mais acima seja parcialmente removida. A zona que está a ser removida corresponde ao reflexo da luz na bola, tornando-a roxa e, assim, aproximando-se mais dos valores dos azuis. Nas outras bolas também é removida uma pequena parte, pertencente ao reflexo da luz.

Na imagem original, é de notar que o fundo não tem sempre o mesmo tonalidade de azul, sendo que no canto superior esquerdo é onde o azul é mais claro, por estar mais perto da fonte de luz, e vai escurecendo à medida que se afasta para a direita e para baixo. Assim, na imagem da direita, é possível observar que os azuis mais escuros não são interpretados como pertencendo ao fundo, não sendo removidos. No entanto, nota-se que se obtém um contorno mais realista das bolas de Natal, nomeadamente na bola mais acima.

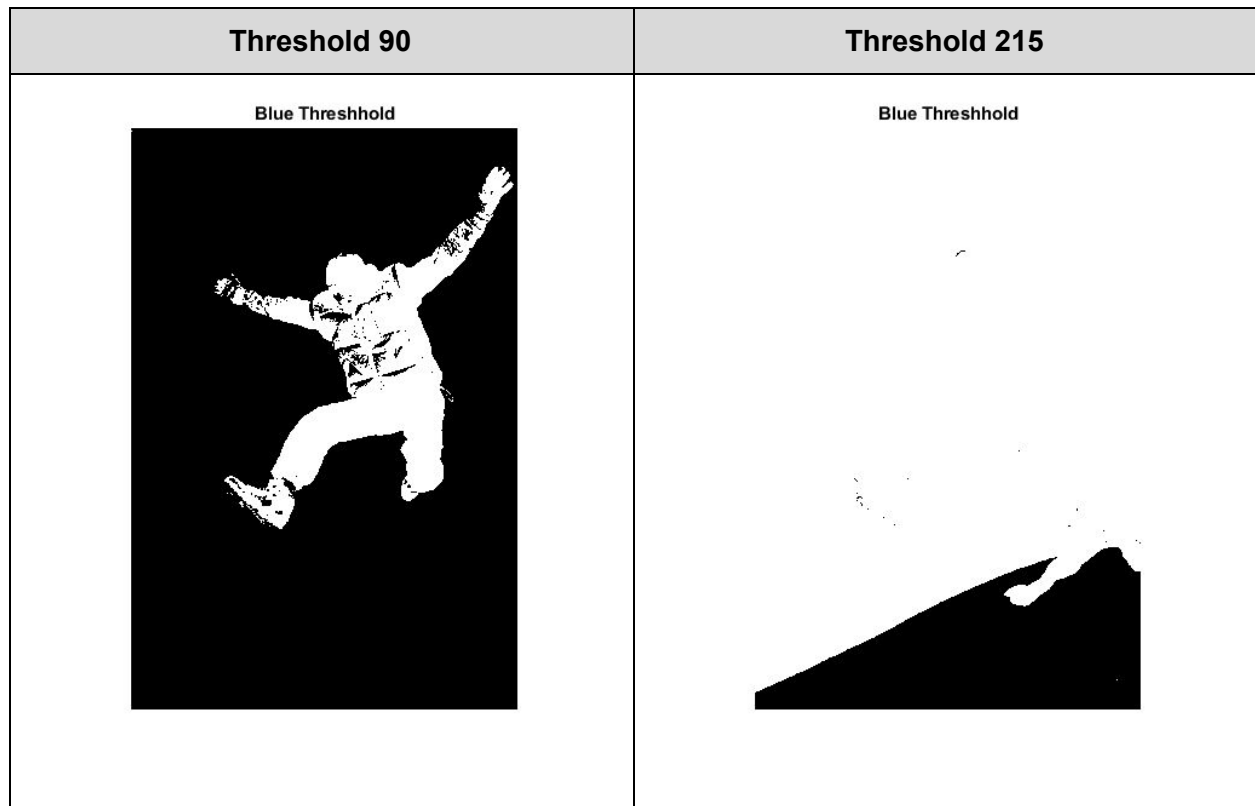
Relativamente à imagem “jumpBB.jpg”, foi obtido o seguinte histograma:



Neste histograma existe uma zona de densidade de azuis entre os 100 e os 200 Hz e novamente entre os ~220 a 240. Logo, foram escolhidos como threshold valores imediatamente antes destas zonas.

Existem duas zonas de elevada densidade porque, para além da imagem conter um fundo azul, contém também um monte de neve, de tonalidade bastante clara. Isto pode causar efeitos indesejáveis no resultado final, por causa de haver risco de o monte de neve ser também removido juntamente com o fundo azul.

Foram obtidas as seguintes imagens, após inserção de threshold.



Na imagem à esquerda foi removido o fundo azul, junto com o monte de neve, mantendo o contorno do homem. Na imagem da direita foram removidos o fundo azul e a sombra do homem no monte de neve. Isto confirma que a zona do histograma entre os 100 e os 200 Hz pertence ao fundo azul e a zona entre os 220 e 240 Hz pertence ao monte de neve.

Esta técnica não é recomendada devido à ineficiência a distinguir o fundo azul de tonalidades claras, provocando erros na imagem final.

1.2 - Segmentação Alternativa

Ao contrário da experiência anterior, é utilizado um método de segmentação que usa a componente *blueness*, que indica a pureza de azul que o pixel tem. Para calcular a componente de blueness foi utilizada a seguinte fórmula :

$$\text{Blueness} = \text{Blue} - \text{MAX}(\text{Red}, \text{Green})$$

Para a realização desta experiência, foi escrito o seguinte Matlab script:

```

function alternative_segment (inputImage)

%imports a coloured image with a blue background and presents
that image on the
%screen
%reads image and converts to rgb if it is in greyscale
image=imread(inputImage);
if size(image,3) ~= 3
    image = cat(3,image,image,image);
end

% get image dimensions: an RGB image has three planes
% reshaping puts the RGB layers next to each other generating
% a two dimensional grayscale image

[height, width, planes] = size(image); %obtain the dimensions of the matrix
%rgb = reshape(image, height, width * planes);

figure(1),imshow(image),title('imagem original');

%separates each RGB component in a different matrix and visualises each one
on the
%screen

%imagesc(rgb);                % visualize RGB planes
%colorbar on                  % display colorbar

r = image(:, :, 1);          % red channel
g = image(:, :, 2);          % green channel
b = image(:, :, 3);          % blue channel

figure(2),imshow(r),title('red');
figure(3),imshow(g),title('green');
figure(4),imshow(b),title('blue');

%computes the blueness of a pixel using the
% equation blueness = B?max(R,G).
b_aux = b;
for i=1:height
    for j=1:width
        b_aux(i,j) = b(i,j) - max(r(i,j),g(i,j));
    end
end

% write here the code to plot the histogram

```

```

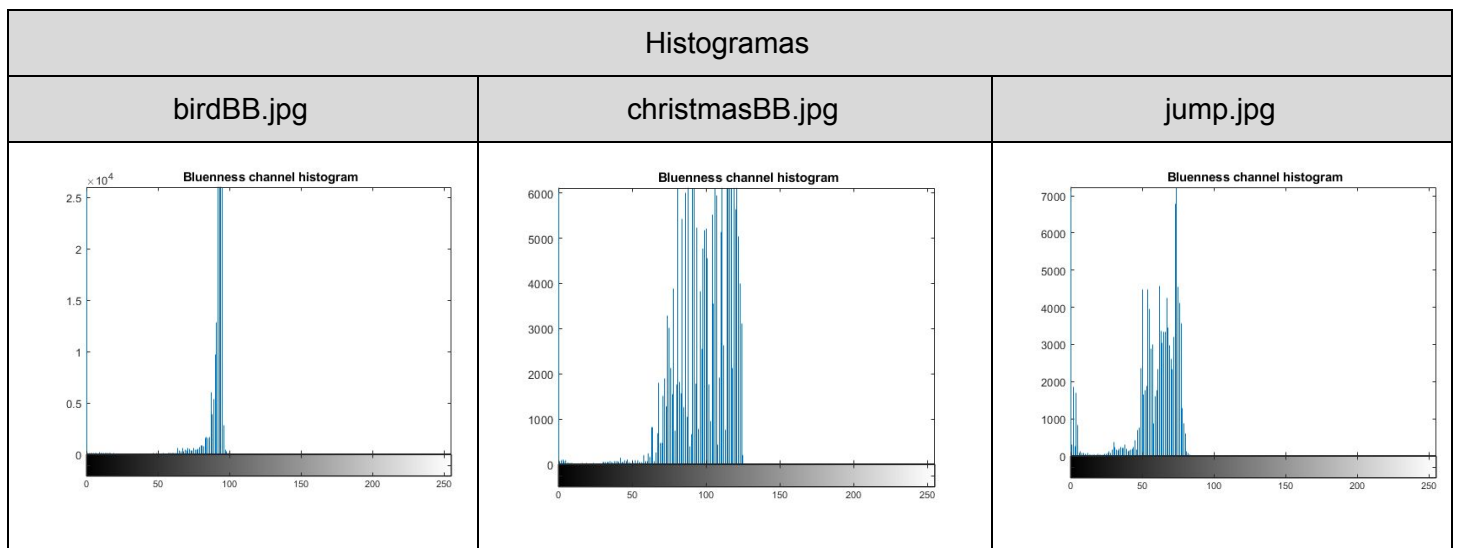
figure(5),imhist(b_aux), title(' Blueness channel histogram');
threshold=input('Which threshold?'); % ask the user for a threshold


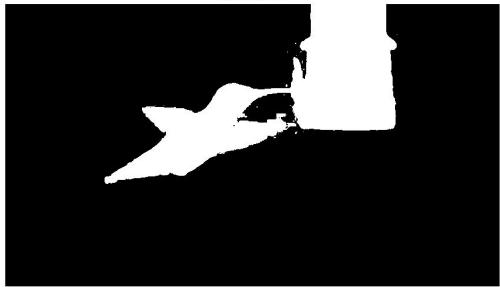
%threshold = 100;           % threshold value
%imagesc(b < threshold);    % display the binarized image



% write here the code to generate a new matrix with dimensions "height" and
% "width" with white pixels in the locations corresponding to the locations
in the B matrix with low values
c=zeros(height, width);
for i=1:height
    for j=1:width
        if(b_aux(i,j)<threshold)
            c(i,j)=255;
        end
    end
end
figure(6), imshow(c),title('Blue Threshold');



```

As imagens utilizadas nesta experiência foram as mesmas que na anterior, e, obteve-se os seguintes resultados:



birdBB.jpg	
Threshold 60	Threshold 80
<p>Blue Threshold</p> 	<p>Blue Threshold</p> 

christmasBB.jpg	
Threshold 50	Threshold 60
<p>Blue Threshold</p> 	<p>Blue Threshold</p> 

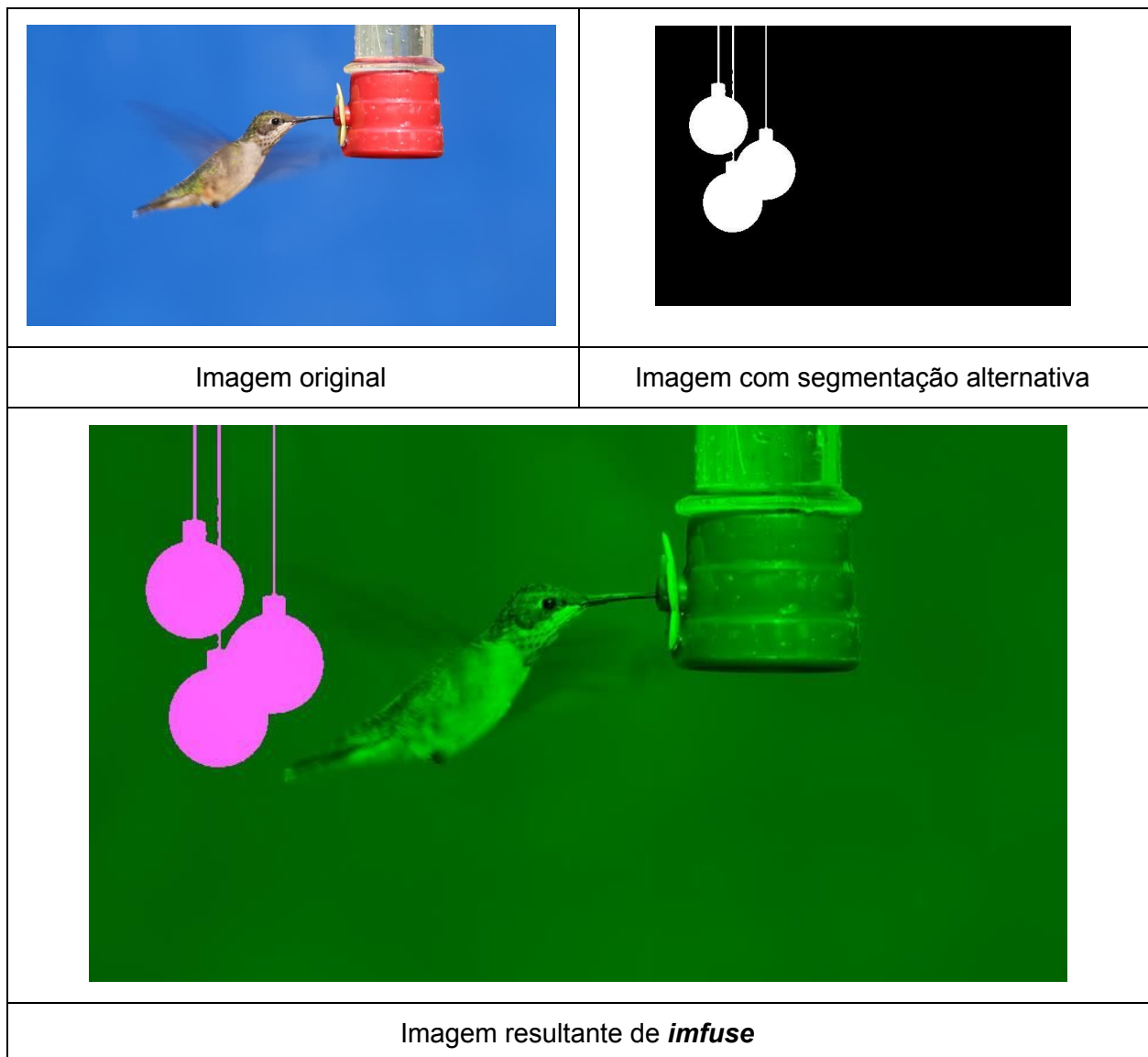
jump.jpg	
Threshold 25	Threshold 40
<p>Blue Threshold</p> 	<p>Blue Threshold</p> 




Uma comparação dos histogramas por segmentação alternativa com os de segmentação simples permite concluir que os primeiros têm uma gama de frequências menor, como por exemplo, enquanto que o histograma por segmentação simples da imagem 'jump.jpg' tem uma gama dos 0 aos 250, o histograma da mesma imagem por segmentação alternativa apresenta uma gama dos 50 aos 70, tendo em conta que os valores de *blueness* são inferiores aos de *blue*, pois a esta componente é subtraída a componente *green* ou *red*. Isto facilita o isolamento do fundo azul porque pixels que tenham uma grande componente *blue* sem serem azuis expressam-se muito menos no histograma do que pixels que tenham tons de azul.

Conclui-se assim que, apesar de o segundo método necessitar de um maior número de cálculos, produz resultados superiores para todos os casos testados, já que apenas afeta secções de imagem puramente azuis, e não zonas com componente azul elevada que não pertençam a esta componente.

2 - Adicionar objetos a outras imagens

Nesta experiência foi usada uma função embutida no matlab, ***imfuse***, de forma a juntar uma imagem segmentada (resultante da experiência anterior) a uma imagem original. O método *default* usado na função é o “*falsecolor*” que cria uma imagem RGB composta mostrando as duas imagens sobrepostas em bandas de cores diferentes. As regiões cinzas na imagem composta mostram onde as duas imagens têm as mesmas intensidades. As regiões magenta e verde mostram onde as intensidades são diferentes. É possível verificar isto nas imagens abaixo:



	
<p>Imagem original</p>	<p>Imagem com segmentação alternativa</p>
	
<p>Imagem resultante de <i>imfuse</i></p>	

Na segunda parte desta experiência foi modificado o script de forma a gerar uma imagem segmentada a cores. Este bloco de código foi adicionado ao script anterior:

```
r1=zeros(height, width);
g1=zeros(height, width);
b1=zeros(height, width);
```

```

for i=1:height
    for j=1:width
        if(b_aux(i,j)<threshold)
            r1(i,j)=r(i,j);
            g1(i,j)=g(i,j);
            b1(i,j)=b(i,j);
        end
    end
end
d=cat(3,uint8(r1),uint8(g1),uint8(b1));
figure(6), imshow(d),title('Segmented Image with Color');

```

Depois de geradas as imagens segmentadas a cor, foi efetuado o mesmo processo anteriormente utilizado com o ***imfuse***, em que foram obtidas as imagens nas tabelas seguintes:


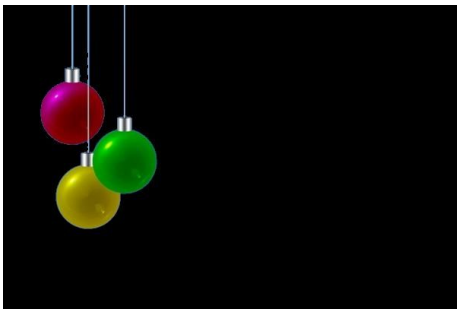

	
<p>Imagem original</p>	<p>Imagem com segmentação alternativa a cores</p>
	

Imagem resultante de *imfuse*



Imagem original



Imagem com segmentação alternativa a cores



Imagem resultante de *imfuse*

Nota-se que, usando uma imagem segmentada com cores, apresenta alguma transparência no *imfuse*, uma vez que a intensidade das cores já são mais baixas comparativamente com o branco utilizado anteriormente.