



Universidade do Minho
Escola de Engenharia

COMPUTER VISION

TUTORIAL 2

IMAGE RECOGNITION

GRUPO 1



Filipe Guimarães A85308



Hugo Cunha A84656



Joana Afonso Gomes A84912

Braga, 30 de novembro de 2020

Conteúdo

1	INTRODUÇÃO	2
2	RESOLUÇÃO DO PROBLEMA PROPOSTO	3
2.1	BACKGROUND EXTRACTION	4
2.2	CONTROLO MORFOLÓGICO	5
2.3	Feature Extraction	7
3	CONCLUSÃO & ANÁLISE DE RESULTADOS	8

1 INTRODUÇÃO

No âmbito da unidade curricular de **Visão por Computador**, foi-nos proposta a realização deste tutorial, que consiste na resolução de exercícios relacionados com **reconhecimento de imagem**.

O objetivo deste trabalho é, a partir de uma sequência de imagens/vídeo capturadas com uma câmara RGB-D, detetar os pés, particularmente as suas articulações.

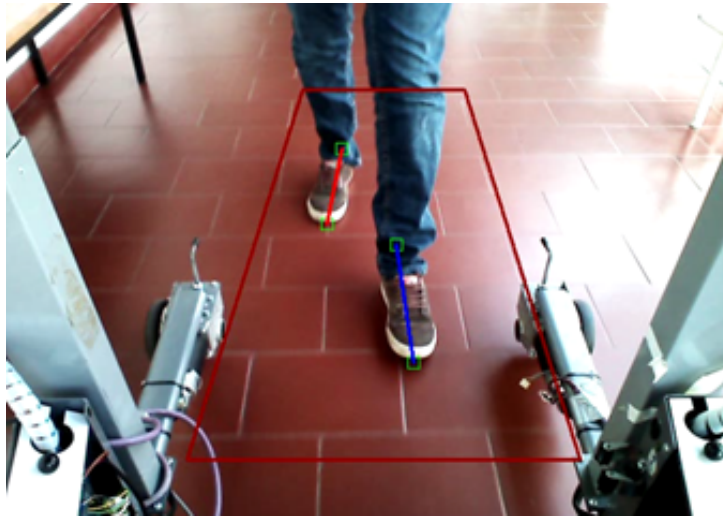


Figura 1: Guia instrucional para o objetivo do projeto

2 RESOLUÇÃO DO PROBLEMA PROPOSTO

Nesta secção do relatório iremos explicar o nosso algoritmo desenvolvido para o problema proposto. Como guia do mesmo fizemos um diagrama que demonstra todos os passos desde a obtenção de cada frame até à escrita da mesma no resultado final.

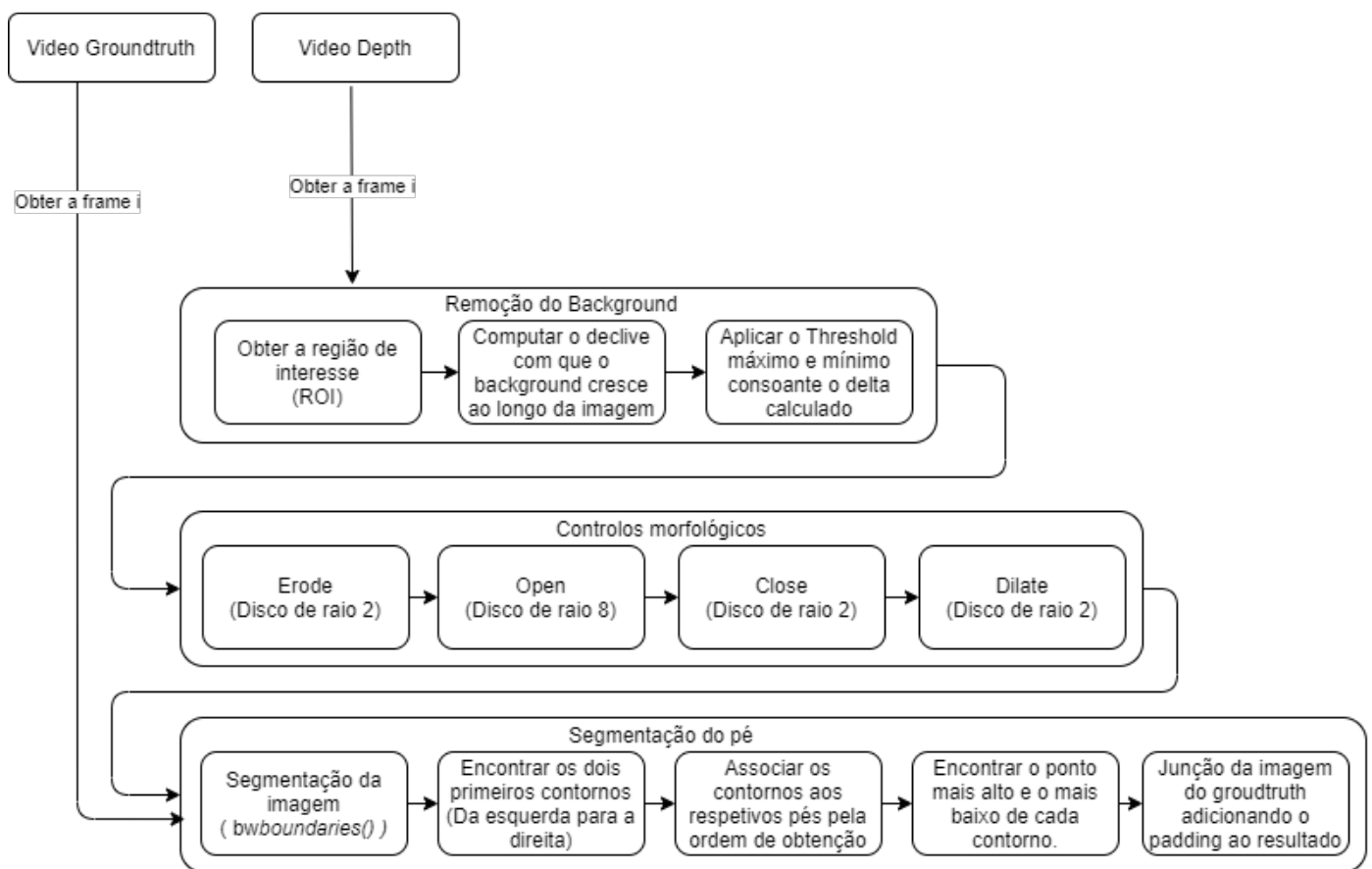


Figura 2: Diagrama final do algoritmo utilizado

As seguintes secções iram explicar ao detalhe cada uma das três grandes partes deste algoritmo.

2.1 BACKGROUND EXTRACTION

Para remover o *background*, primeiramente cortamos a imagem, de forma a apenas tratarmos a nossa região de interesse. Para isto cortamos cerca de 1/4 das margens verticais e 3/8 das margens horizontais.

```
1 depthCrop = depth(imgSize(1)/4 :imgSize(1)*3/4,imgSize(2)
   *3/8 :imgSize(2)*5/8);
```

Depois de remover as margens, podemos passar à detecção do *background*. Como sabemos que a câmara de profundidade aumenta os valores de intensidade dos pixels linearmente com a altura da imagem obtida, podemos extrair o declive desta função ao observar a diferença de intensidade entre os pixels de linhas adjacentes. Fazemos a média das primeiras 20 linhas da imagem para apagar o erro do sensor e calculamos as diferenças entre elas das quais acabamos por fazer uma média obtendo a nossa estimativa do declive.

```
1 [sizeCropX,sizeCropY] = size(depthCrop);
2 for i = 0:20
3     delta(i+1) = mean(depthCrop(sizeCropX-(i+1),:))
4                 -mean(depthCrop(sizeCropX-i,:));
5     deltaD = mean(delta);
6 end
```

Com esta estimativa criamos dois valores de máximo e de mínimo em cada linha para fazer um *double thresholding* manual onde procuramos e colocamos a 0 todos os pixels que tenham até -200 de intensidade do valor esperado para o *background* por linha.

Isto permite alguma robustez contra a posição da câmara e até algumas oscilações que podem surgir da interação com o utilizador.

```
1 for i=1:sizeCropX
2     k = sizeCropX - i +1;
3     dmax = (i)*deltaD + mean(depthCrop(sizeCropX,:));
4     dmin = dmax - 200;
5     for j = 1:sizeCropY
6         if(newImage(k,j)>=dmax-10 || newImage(k,j)<dmin )
7             newImage(k,j)=0;
8         end
```

```

9     end
10    deltaD=deltaD + 0.003;
11 end

```

No final obtemos uma imagem onde o background e as pernas foram removidos da imagem.

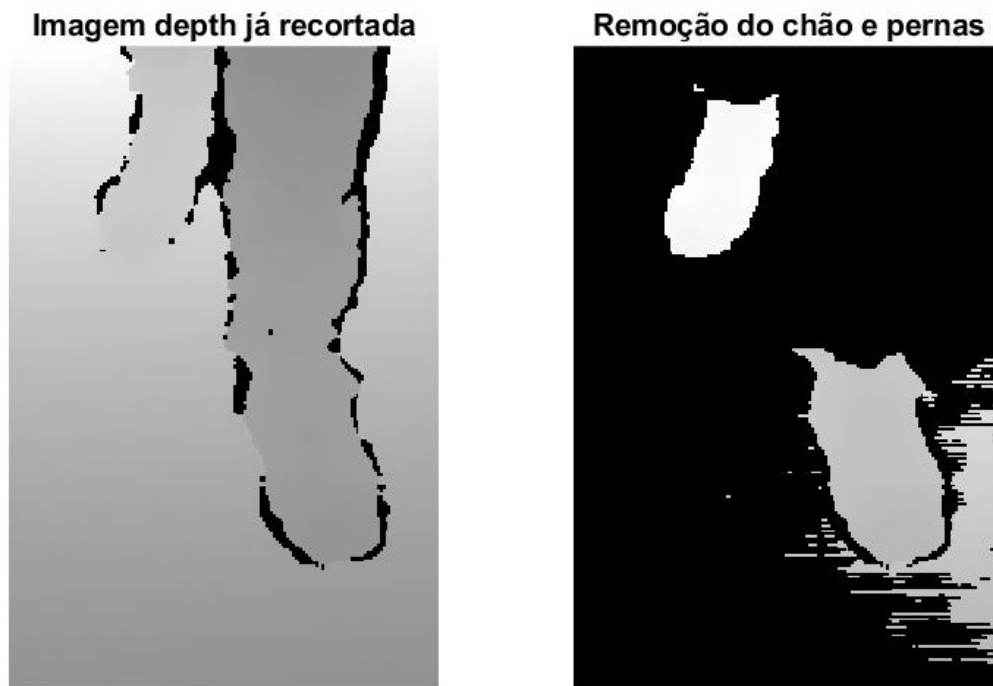


Figura 3: Remoção do background

2.2 CONTROLO MORFOLÓGICO

Após obtermos uma imagem essencialmente com os objetos de interesse binarizados, vamos aplicar operações morfológicas para eliminar cantos e imperfeições indesejadas das nossas áreas de interesse, bem como reduzir o tamanho de alguns erros na deteção que poderiam, numa próxima fase, serem identificados como objetos.

Para isto começamos por aplicar um *erode* com um disco de raio para diminuir as imperfeições

```

1 se=strel('disk',2,6);
2 newImage = imerode(newImage, se);

```

Depois aplicamos um open com um disco de raio 8 para eliminar os pequenos erros.

```

1 se=strel('disk',8,0);
2 newImage=imopen(newImage,se);

```

De seguida, um *close* para preencher possíveis zonas a preto dentro dos objetos.

```

1 se=strel('disk',2,6);
2 newImage=imclose(newImage,se);

```

Acabamos a fazer um *dilate* para recuperar o tamanho das zonas

```

1 se=strel('disk',2,6);
2 newImage=imdilate(newImage,se);

```

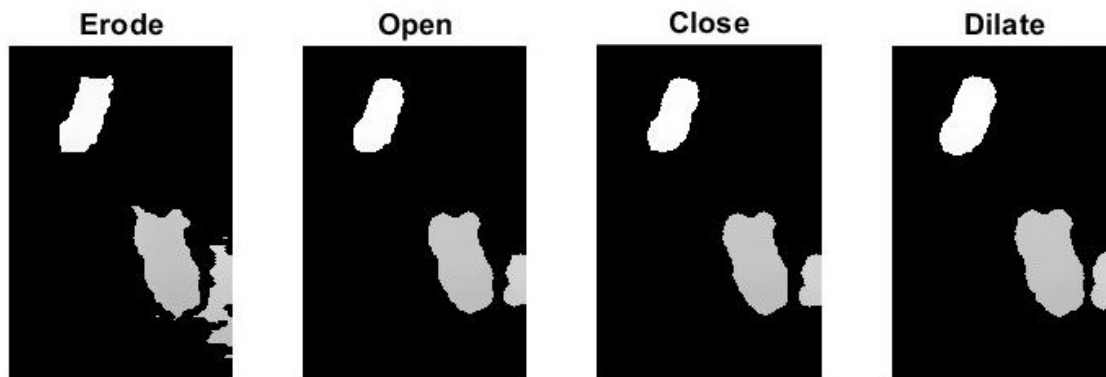


Figura 4: Controlo Morfológico

2.3 Feature Extraction

Para extrairmos os objetos usamos a função `bwboundaries`. Esta função seleciona as manchas de intensidade > 0 , agrupando os contornos das manchas separadas em *arrays*. De seguida, procuramos nestes *arrays* os pontos que se encontram mais acima e mais abaixo do contorno. O ponto mais acima é considerado o tornozelo e o ponto mais abaixo será a ponta do pé.

```
1 [~,indice] = min(left(:,1));  
2 ponta_esquerda = left(indice,:);  
3 plot(ponta_esquerda(2),ponta_esquerda(1),'s');  
4  
5 [~,indice] = max(left(:,1));  
6 tornozelo_esquerdo = left(indice,:);  
7 plot(tornozelo_esquerdo(2),tornozelo_esquerdo(1),'s');
```


3 CONCLUSÃO & ANÁLISE DE RESULTADOS

A realização deste tutorial permitiu ao grupo aprofundar os conhecimentos adquiridos nas aulas, no que toca ao reconhecimento em imagem.

Junto com este relatório, além do código em *Matlab*, é enviado também o resultado em vídeo da aplicação do nosso algoritmo às imagens *depth* sobreposto ao vídeo de *groundtruth*. Ao analisar este vídeo ficamos satisfeitos com os resultados, visto que os pontos ficam sobrepostos ou muito perto dos pontos fornecidos.



Figura 5: Alguns exemplos de frames do resultado final