

1º SEMESTRE 2018/2019, MEEC

DETEÇÃO E TRACKING DE OBJETOS EM MOVIMENTO UTILIZANDO CÂMERAS DE PROFUNDIDADE

PROCESSAMENTO DE IMAGEM E VISÃO

RELATÓRIO

Autores:

Nome: Miguel Paulino	Número: 79168
Nome: Diogo Morgado	Número: 84032
Nome: Luís Afonso Lopes	Número: 84116
Nome: Valter Piedade	Número: 84195

Grupo: 30

21 de Dezembro de 2018

1. INTRODUÇÃO

1.1 Descrição do problema

Atualmente, um dos principais e mais relevantes problemas do processamento de imagem é a detecção e rastreamento de objetos em movimento. Este problema consiste em segmentar sequências de imagens, provenientes de uma ou mais câmeras, de modo a conseguir detetar e identificar cada um dos objetos presentes nessas sequências de imagens.

O projeto de Processamento de Imagem e Visão encontra-se dividido em duas partes, sendo que o problema da primeira parte consiste em detetar e rastrear objetos ao longo de uma sequência de imagens provenientes de apenas uma câmera de profundidade (que se encontra estática), enquanto que o problema da segunda parte consiste em detetar e rastrear objetos ao longo de duas sequências de imagens provenientes de duas câmeras de profundidade, ambas estáticas também.

Relativamente ao problema da primeira parte, e como se trata de apenas uma câmera cujos parâmetros intrínsecos e extrínsecos são conhecidos, basta identificar os objetos na respetiva sequência de imagens, representar os seus pontos em 3D e identificar cada um dos objetos detetados de forma a ser possível fazer o seu rastreamento. Em relação ao problema da segunda parte, como existem duas câmeras, definiu-se que a câmera 1 é aquela que define o sistema de coordenadas do mundo. É, portanto, necessário calcular as transformações entre o sistema de coordenadas da câmera 2 e o sistema de coordenadas do mundo (sistema de coordenadas da câmera 1). De seguida, é realizada a detecção e identificação dos objetos (de igual forma à primeira parte) em ambas as sequências de imagens das duas câmeras e, sobre os objetos detetados na câmera 2, são realizadas as respetivas transformações de modo a obter todos os objetos no mesmo sistema de coordenadas do mundo.

1.2 Descrição da câmera de profundidade

A câmera de profundidade utilizada para gerar as sequências de imagens a analisar foi a Microsoft Kinect, desenvolvida para a Xbox 360. Esta câmera possui, como é possível observar na Figura 1, três sensores diferentes, uma câmera RGB, dois sensores de profundidade 3D e um multi-array de microfones. No entanto, para o propósito deste projeto, apenas serão utilizados os sensores de profundidade 3D e a câmera RGB.



FIGURA 1 - REPRESENTAÇÃO DOS SENSORES PRESENTES NA MICROSOFT KINECT.

Tanto a câmara RGB como os sensores de profundidade 3D fornecem imagens com uma resolução de 640x480 pixels, sendo que a primeira fornece uma imagem de cor e a segunda uma imagem de profundidade, representadas na Figura 2. De notar que as câmaras se encontram ligeiramente distanciadas uma da outra, não sendo possível fazer, diretamente, a associação entre pontos da imagem RGB e pontos da imagem de profundidade.

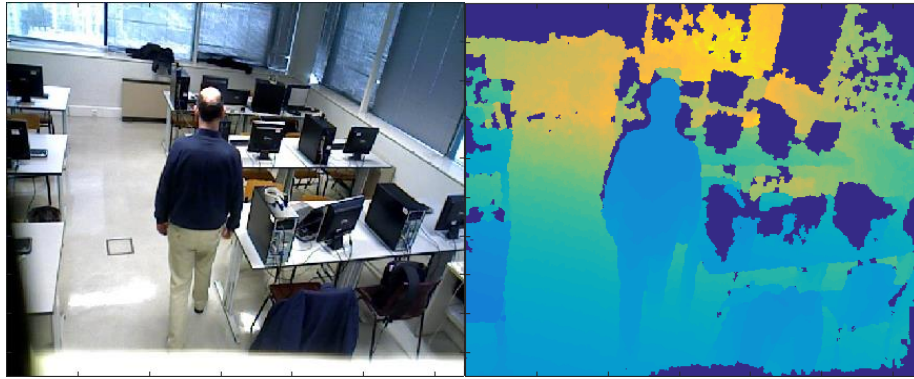


FIGURA 2 - REPRESENTAÇÃO DE UMA IMAGEM RGB (À ESQUERDA) E DA CORRESPONDENTE IMAGEM DE PROFUNDIDADE (À DIREITA).

1.3 Modelo da câmara

O modelo da câmara permite descrever, matematicamente, a relação entre as coordenadas de um ponto em 3D (dado por $\mathbf{X} = [X \ Y \ Z]^T$) e a sua projecção no plano de uma imagem 2D (onde os pontos são dados por $\mathbf{x} = [x \ y]^T$). Este modelo matemático baseia-se no modelo da câmara pinhole, representado na Figura 3.

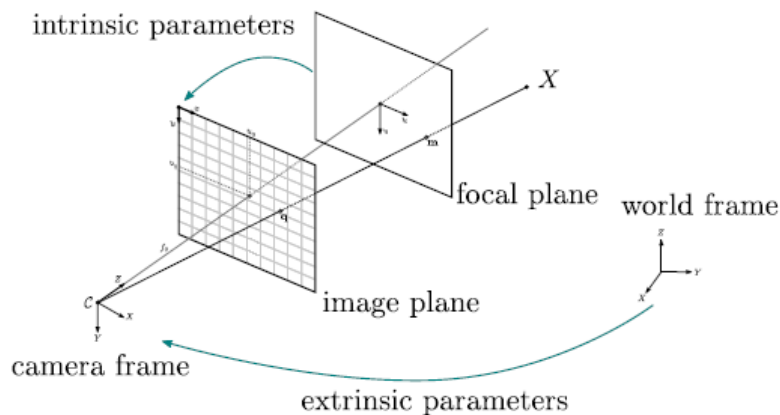


FIGURA 3 – REPRESENTAÇÃO DO MODELO DA CÂMERA PINHOLE.

A projecção em perspetiva de uma câmara normalizada, em que a distância entre o centro de projecção e o ponto de projecção (denominada por distância focal) é $f=1$, é dada por

$$x = \frac{X}{Z} \quad y = \frac{Y}{Z} \quad (1)$$

ou, em coordenadas homogêneas, por

$$\lambda \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \Leftrightarrow \lambda \tilde{x} = [I \ 0] \tilde{X} = X \quad (2)$$

Duas propriedades do modelo da câmera são os parâmetros intrínsecos e extrínsecos. Os primeiros dizem respeito aos parâmetros internos da câmera (que dependem das suas características) e são utilizados para realizar a conversão de coordenadas métricas para pixel através das seguintes equações:

$$\begin{cases} x' = fs_x x + c_x \\ y' = fs_y y + c_y \end{cases} \quad (3)$$

onde:

- $x = [x \ y]^T$ está em metros;
- $x' = [x' \ y']^T$ está em pixels;
- f representa a distância focal;
- c_x e c_y representam, em pixels, as coordenadas do ponto principal;
- s_x e s_y representam o fator de escala nas direções x e y , em pixel/m;

Esta conversão de coordenadas métricas para pixels é dada, em coordenadas homogêneas por

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} fs_x & 0 & c_x \\ 0 & fs_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \Leftrightarrow \tilde{x}' = K \tilde{x} \quad (4)$$

em que K é uma matriz triangular superior que contém os parâmetros intrínsecos acima indicados.

Em relação aos parâmetros extrínsecos da câmera, estes dizem respeito à posição da câmera e são utilizados para realizar a conversão das coordenadas do mundo para as coordenadas da câmera. Esta transformação é expressa por

$$X = RX' + T \quad (5)$$

onde:

- $X = [X \ Y \ Z]^T$ corresponde às coordenadas de um ponto no referencial da câmera;
- $X' = [X' \ Y' \ Z']^T$ corresponde às coordenadas de um ponto no referencial do mundo;
- $R \in \mathbb{R}^{3 \times 3}$ representa a matriz rotação e expressa a rotação entre o referencial do mundo e o da câmera. De notar que R é uma matriz ortogonal, pelo que respeita as seguintes propriedades:

$$\begin{cases} \det R = 1 \\ R^T R = I \end{cases} \quad (6)$$

- $T \in \mathbb{R}^{3 \times 1}$ representa um vetor de translação que expressa a origem do referencial do mundo no referencial da câmera.

Combinando as expressões (5) e (2) vem

$$\lambda \tilde{x} = [R \ T] \tilde{X}' \quad (7)$$

Pelo que o modelo final da câmara é dado por

$$\lambda \tilde{x}' = K[R \ T] \tilde{X}' = P \tilde{X}' \quad (8)$$

em que $P = K[R \ T]$ é uma matriz 3x4 e representa a matriz da câmara.

Definido este modelo matemático da câmara, é agora possível fazer a associação entre pontos de uma imagem RGB e pontos de uma imagem de profundidade, pelo que será também possível representar toda uma imagem (ou parte dela) em 2D numa point cloud (representação 3D).

2. IMPLEMENTAÇÃO

2.1 Câmera Única Estática

A primeira parte do projeto consiste em detetar e rastrear objetos em movimento, com base numa sequência de imagens (de RGB e profundidade) obtidas através de uma única câmara estática.

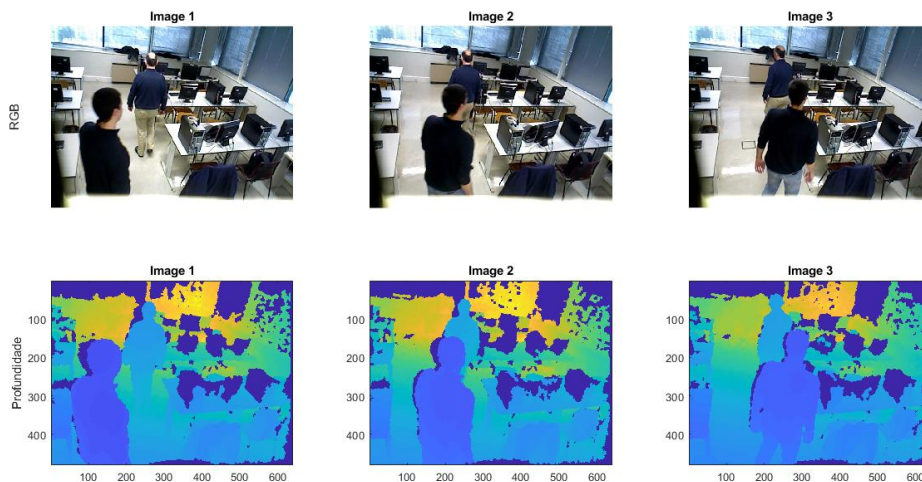


FIGURA 4 – EXEMPLO DE UMA SEQUÊNCIA DE IMAGENS DE RGB E DE PROFUNDIDADE DAS QUAIS SE PREDENTE DETETAR E RASTREAR OBJETOS EM MOVIMENTO (DATASET LAB-FILINHA).

De forma a resolver o problema descrito, utiliza-se a seguinte abordagem:

- i. Detecção dos objetos em movimento:
 - 1) Cálculo e remoção do *background* e obtenção do *foreground*;
 - 2) Filtragem de ruído;
 - 3) Separação de objetos sobrepostos;
 - 4) Identificação dos objetos na *point cloud*.
- ii. Rastreamento dos objetos:
 - 5) Cálculo dos centroides de cada objeto identificado;
 - 6) Cálculo do histograma de cor, usando o matiz (hue), de cada objeto identificado;
 - 7) Comparação com a distância aos centroides dos objetos da imagem anterior.

- 8) Comparação com o histograma de cor dos objetos da imagem anterior.

Cálculo e remoção do *background* e obtenção do *foreground*:

Esta operação pode ser realizada tanto com o conjunto de imagens RGB como com as imagens de profundidade. Optou-se por utilizar as imagens de profundidade por conveniência, uma vez que nos próximos passos os cálculos serão realizados utilizando os dados da profundidade e porque, caso se utilizasse o conjunto de imagens RGB, não seria possível passar corretamente do *background* obtido nesse caso para a profundidade.

De entre os vários métodos para obter o *background*, testou-se o uso da média e da mediana. Verifica-se nas Figuras 5(a) e 5(b) que é gerado um *background* mais rigoroso utilizando a mediana, uma vez que esta atribui a cada pixel o valor que lhe ocorre mais vezes (suposto *background*), enquanto que a média conta com todos os valores que ocorrem nos pixéis (incluindo assim também *outliers*), resultando num *background* com maior ruído.

Também se verificou que a mediana, apesar de ser uma boa solução, possui certas limitações, como se observa no *dataset* lab-filinha, em que existe uma região do *background* onde se obteve um erro (Figuras 5 (c) e 5 (d)) porque na maioria das imagens existiam objetos em movimento nessa região.

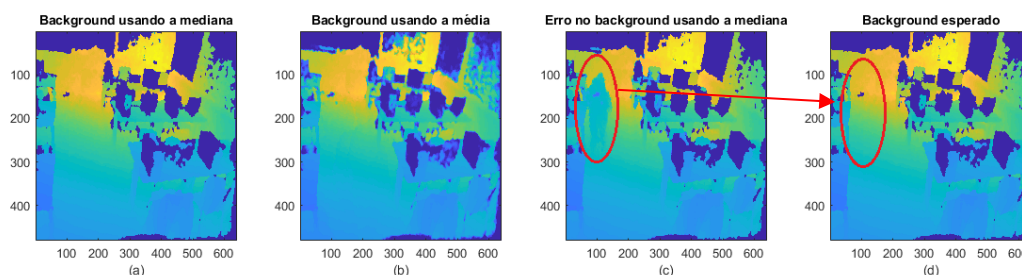


FIGURA 5 – BACKGROUND OBTIDO PARA O CONJUNTO DE IMAGENS DE PROFUNDIDADE USANDO A MEDIANA (A) E A MÉDIA (B). ERRO OBTIDO UTILIZANDO A MEDIANA NO DATASET LAB-FILINHA (C) EM COMPARAÇÃO COM O SEU RESULTADO ESPERADO (D), OBTIDO USANDO APENAS AS PRIMEIRAS 30 IMAGENS DO DATASET.

Após ter sido adquirido o *background* em profundidade, procede-se à subtração de cada imagem pelo *background* calculado, obtendo-se assim o *foreground* para cada uma destas que corresponde aos objetos em movimento que se pretendem detetar.

Filtragem de ruído:

Os *foreground*, obtidos por subtração de cada imagem pelo *background*, correspondem a uma imagem binária que possui bastante ruído (Figura 6 (a)). É, portanto, necessário eliminar o máximo de ruído possível recorrendo, para isso, a um filtro morfológico.

Para aplicar o filtro, primeiramente efetua uma erosão da imagem, onde é eliminado tanto o ruído como partes dos objetos (Figura 6 (b)). De seguida, efetua-se uma dilatação da imagem, o que devolve aos objetos a sua forma original que tinha sido, em parte, perdida durante a erosão. A vantagem deste procedimento é que como o ruído é eliminado durante a erosão, este não vai ser dilatado, obtendo-se assim o *foreground* livre de grande parte do ruído (Figura 6 (c)).

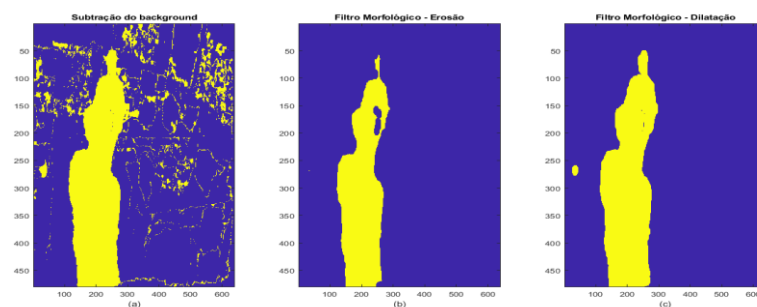


FIGURA 6 – RESULTADO DA SUBTRAÇÃO DO BACKGROUND (A) E DA APLICAÇÃO DO FILTRO MORFOLÓGICO (EROSÃO (B) E DILATAÇÃO (C)).

Separação de objetos sobrepostos:

Depois de removido o ruído, adquire-se uma imagem binária onde se observa, claramente, quais os pontos da imagem correspondentes aos objetos em movimento (Figura 7 (a)). No entanto, levanta-se outro problema, que corresponde em saber quantos objetos estão presentes na imagem, e quais os limites de cada um.

De forma a resolver esta situação, calcula-se o gradiente (em x, Figura 7 (b) e em y, Figura 7 (c)) da profundidade para cada ponto pertencente ao *foreground* e averigua-se entre que pixéis existem grandes variações do mesmo. Essas variações podem corresponder ou a transições entre objetos, ou ainda a algum ruído que possa existir na imagem. Os pixéis onde se verificar esta situação serão, na imagem binária, colocados a 0, o que permite individualizar cada objeto diferente detetado, como se verifica na Figura 7 (d).

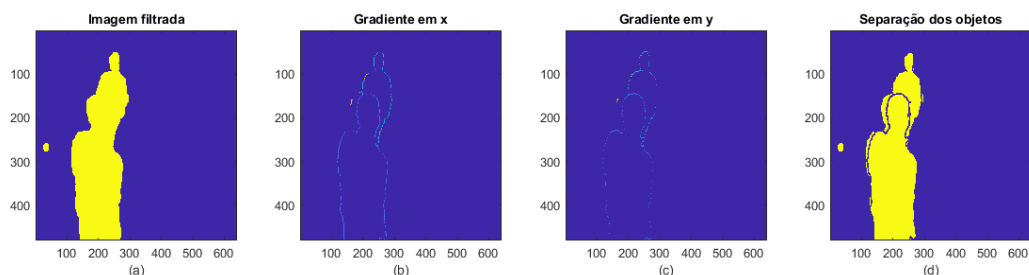


FIGURA 7 – IMAGEM FILTRADA MORFOLOGICAMENTE (A) E RESPECTIVOS GRADIENTES DE PROFUNDIDADE EM X (B) E Y (C) PARA CADA ÍNDICE PERTENCENTE AO FOREGROUND. IMAGEM RESULTANTE DA SEPARAÇÃO DOS OBJETOS RECORRENDO AO GRADIENTE (D).

Identificação dos objetos na *point cloud*:

De forma a obter as 8 coordenadas (x,y,z) que definam uma caixa que engloba cada objeto detetado, procedeu-se à identificação dos mesmos objetos de forma a que cada um deles possuísse um índice diferente (figura 8 (a)) para melhor distinção entre os mesmos. De seguida, para cada objeto, procede-se à transformação dos valores de profundidade correspondentes aos índices identificados como pertencentes ao objeto (2D) para uma *point cloud* em 3D (Figuras 8 (b) e 8 (c)). Como se está a realizar a *point cloud* com cada objeto separadamente, e estes já não possuem ruído, para adquirir os pontos (x,y,z) que formam uma caixa em torno do objeto apenas é necessário calcular os máximos e mínimos da *point cloud* para cada um dos eixos. As caixas obtidas podem ser observadas novamente nas Figuras 8 (b) e 8 (c).

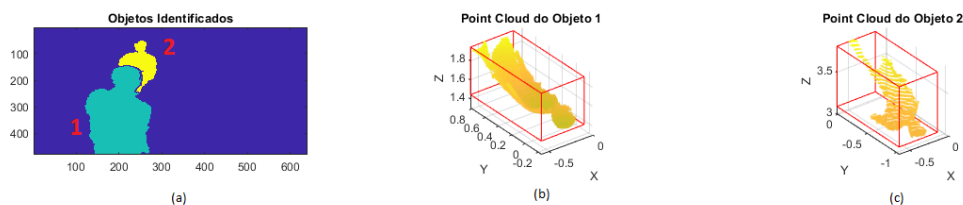


FIGURA 8 – OBJETOS DETETADOS IDENTIFICADOS COM INDICES DIFERENTES (A). POINT CLOUD DO OBJETOS 1 (B) E DO OBJETO 2 (C).

Cálculo dos centroides de cada objeto identificado e comparação com a distância aos centroides dos objetos da imagem anterior:

Tendo já sido detetados os objetos, e com fim a rastrear o seu movimento ao longo de uma sequência de imagens, optou-se por utilizar a distância entre os centroides das caixas de cada objeto da imagem em questão e os centroides das caixas dos objetos da imagem anterior.

Os resultados obtidos ao testar o *dataset* lab-filinha utilizando apenas este método foram bastante variados. Verificou-se que se obtiveram bons resultados quando o objeto em questão se movimenta lentamente (o que permite a escolha de um *threshold* menor para o valor da diferença de distância entre as caixas, reduzindo o risco de associar o objeto erradamente), quando o número de objetos é reduzido ou quando têm distancias consideravelmente grandes entre eles. Obtiveram-se maus resultados principalmente nas situações em que os objetos se cruzam, sendo, nesse caso, as distâncias entre objetos diferentes reduzidas, induzindo este método em erro. O mesmo acontece quando os objetos se movimentam muito rápido, o que faz com que ou se aumente o *threshold* e se corra o risco de detetar também outros objetos como possíveis caminhos, ou que não se aumente o *threshold* e se perca o rasto do objeto.

Cálculo dos histogramas de cor de cada objeto identificado e comparação com os histogramas de cor dos objetos da imagem anterior:

Tendo já sido detetados os objetos, e com fim a rastrear o seu movimento ao longo de uma sequência de imagens e tentando melhorar os resultados obtidos utilizando apenas o método das distâncias entre os centroides, também se optou por utilizar o histograma de cor, obtido a partir do matiz (*hue*), de cada objeto da imagem em questão e os histogramas de cor dos objetos da imagem anterior.

De modo a medir o grau de semelhança entre os histogramas de cor foi definida uma expressão que determina uma medida de erro entre os histogramas de cor a serem contemplados. Após esse cálculo esse erro entre os histogramas é passado por uma função exponencial negativa e se o resultado final ficar acima de um *threshold* significa que as objetos que estão a ser comparados são objectos semelhantes em relação à sua matiz (*hue*). O *threshold* escolhido foi 0.2 e foi definido após vários resultados experimentais.

Os resultados obtidos ao testar o *dataset* lad-filinha utilizando também este método foram bastante semelhantes aos resultados obtidos utilizando apenas o método da distância entre os centroides pois o valor definido para o *threshold* na comparação entre histogramas de cor foi definido de forma a que apenas os objetos com cores fortemente diferentes fossem separados. Isto foi definido desta maneira pois como cada objeto em

movimento no *dataset* considerado tem várias cores bastante diferentes (diferença de cor entre pele e roupa) e as cores são bastante semelhantes entre os vários objectos (mais do que uma pessoa a usar roupa azul, cor de pele semelhantes entre pessoas). Obtiveram-se maus resultados ao aumentar o *threshold* (grau de semelhança entre histogramas de cor dos objetos teria de ser mais elevado para serem considerados o mesmo objeto) pois em certos casos o este método dividiu os objetos (pessoas) nos seus elementos (pele, roupa), perdendo-se o rasto do objeto inicial pretendido.

2.2 Duas Câmeras Estáticas:

A segunda parte do projeto consiste novamente em detetar e rastrear objetos em movimento, desta vez com base em sequências de imagens (de RGB e profundidade) provenientes de duas câmeras estáticas.

A abordagem a este novo problema é a seguinte:

- i. Obter matrizes de rotação e translação que transformam as coordenadas da imagem 2 nas coordenadas da imagem 1 (coordenadas do mundo):
 - 1) Obter os melhores pontos de cada imagem;
 - 2) Realizar correspondências entre os pontos das duas imagens;
 - 3) Identificar as melhores correspondências;
 - 4) Calcular as matrizes R e T usando as melhores correspondências entre as 2 imagens.
- ii. Detecção e rastreamento dos objetos em movimento (tal como na parte 1) para a imagem 1 e para a imagem 2 (nas coordenadas da imagem 1).

Obter os melhores pontos de cada imagem e realizar as suas correspondências:

A fim de determinar as melhores correspondências entre as duas imagens, procede-se, inicialmente, à escolha de bons pontos em cada imagem (pontos únicos que possuam características peculiares como por exemplo cantos ou texturas). Para isso utilizou-se a *scale-invariant feature transform* (SIFT), que determina os melhores pontos com base em características como as variações de iluminação, o ponto de vista da imagem, entre outros (figuras 9 (a) e (b)). A vantagem de utilizar o SIFT em vez de outro detetor como por exemplo o *Harris Corner Detector* é que o SIFT apresenta bons resultados mesmo com diferenças de escala, rotação, iluminação e ponto de vista, enquanto que o *Harris Corner Detector* apresenta bons resultados para casos em que se mantém a escala e orientação da imagem.

Após a obtenção dos melhores pontos, realizam-se as correspondências entre os mesmos nas duas imagens tendo por base as características determinadas anteriormente (variações de iluminação, ponto de vista da imagem, etc), utilizando a imagem que tiver produzido um maior número de correspondências (figura 9 (c)).

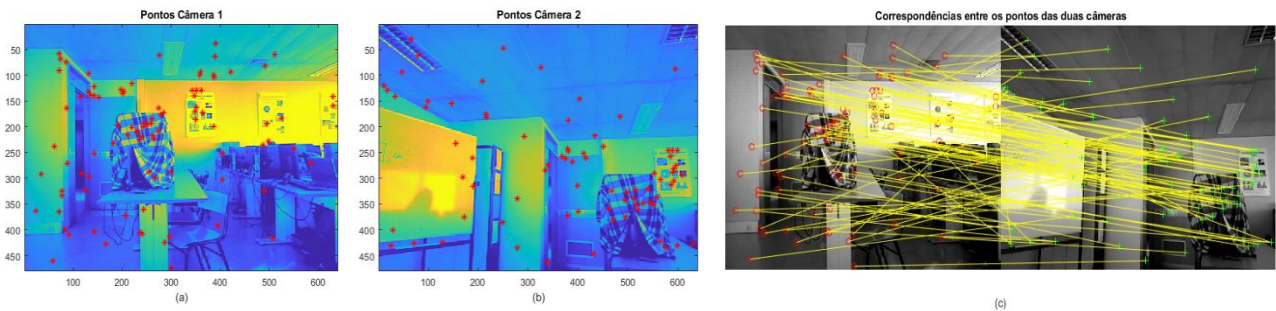


FIGURA 9 – MELHORES PONTOS OBTIDOS NA CÂMERA 1 (A) E NA CÂMERA 2 (B) E RESPECTIVAS CORRESPONDÊNCIAS (C).

Identificar as melhores correspondências e calcular as matrizes R e T entre as 2 imagens:

Depois de realizada a correspondência entre os melhores pontos de cada imagem, é necessário saber se todas as correspondências encontradas estão corretas, ou seja, é necessário garantir que a não existência de *outliers*. Para isso é utilizado o *RANdom Sample Consensus* (RANSAC) que seleciona aleatoriamente 4 pontos e, consoante um *threshold* pré-definido, classifica os pontos em *inliers* e *outliers*. Este processo é repetido diversas vezes com o objetivo de determinar os quatro pontos que obtêm um maior número de *inliers*. Para isso, utiliza-se o *procrustes* para calcular as matrizes R e T que transformam as coordenadas da imagem 2 nas coordenadas da imagem 1 e calculam-se as diferenças entre os pontos correspondidos da imagem 1 e 2. Caso essa diferença seja inferior ao *threshold* esses pontos são *inliers* e correspondem a uma correspondência correta. Pretende-se o maior número de *inliers* possível uma vez que as matrizes R e T que produzirem mais *inliers* são aquelas que transformaram da melhor forma as coordenadas da imagem 2 nas coordenadas da imagem 1.

Aplicando as matrizes R e T obtidas a uma imagem da câmera 2 (figura 10 (b)), verifica-se que se obtêm umas novas coordenadas para esta imagem (figura 10(c)), que correspondem às coordenadas da câmera 1 (figura 10(a)).

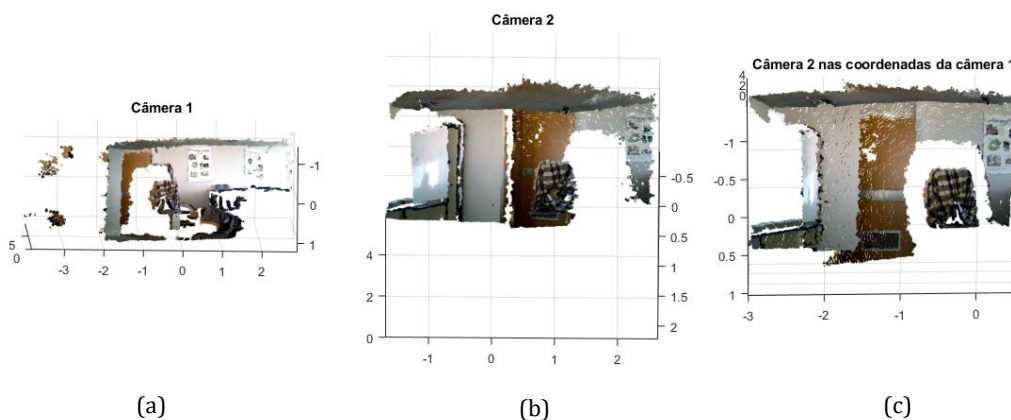


FIGURA 10 – IMAGEM DA CÂMERA 1 (A), DA CÂMERA 2 (B) E DAS COORDENADAS DA CÂMERA 2 NAS COORDENADAS DA CÂMERA 1

Rastreamento de Objectos:

De seguida é efectuado o rastreamento dos objectos para as imagens produzidas por cada uma das câmeras, tal como descrito na anteriormente. No entanto, é agora necessário verificar as correspondências entre os objectos identificados em cada uma das câmeras. Para isso, para cada um dos objectos identificados em cada uma das câmeras compararam-se os *frames* em que os objectos foram identificados. Em caso afirmativo, foram então calculados e comparados os centroides para ambos os objectos e, de acordo com um *threshold* definido, os objectos foram identificados como sendo o mesmo (norma entre centroides menor que o *threshold* definido) ou um objecto diferente (norma entre centroides maior que o *threshold* definido). Para o caso de não haver correspondência entre os *frames* onde os objectos foram identificados estes são considerados objectos novos.

3. CONCLUSÃO

O problema de detecção e rastreamento de objetos em movimento possui inúmeras vertentes de estudo, com os objetos a detetar a poderem variar bastante consoante o caso em questão. Deste modo, é extremamente complicado e complexo determinar uma solução geral para resolver toda, ou grande parte, da gama de problemas existente em relação a este tema.

Tendo isto em conta, optou-se por apenas testar e otimizar o projeto realizado para casos com pessoas em movimento, como é o caso dos datasets lab-um, lab-filinha, entre outros, que foram os utilizados na realização deste projeto.

Mesmo tendo apenas um tipo específico de objetos a identificar (pessoas), verificou-se que continua a ser bastante difícil obter um método de detecção e rastreamento que resulte na perfeição para todos os casos. Isto deve-se, por exemplo, às câmaras poderem estar mais ou menos distantes dos objetos, fazendo com que estes sejam constituídos por um maior ou menor número de pixels. Outra componente que pode variar são as velocidades de movimento das pessoas. Nestas situações, a obtenção de um melhor ou pior resultado depende dos parâmetros definidos, tais como o *threshold* do valor do gradiente ou a distância máxima aceitável para a distância entre os centroides, uma vez que o processo geral de remover o background, de seguida filtrar e separar os objetos e finalmente a representação de cada um numa *point cloud* se mostrou eficiente na resolução do problema quando aliado a valores razoáveis para os parâmetros utilizados.

Em relação à utilização de apenas uma câmara ou de duas câmaras, verificou-se que apesar da utilização de duas câmaras implicar o passo extra de se calcular a matriz de rotação e translação entre as duas imagens, esta irá proporcionar um melhor rastreamento dos objetos. Ao utilizar as duas câmaras em posições distintas, são obtidas duas perspetivas da mesma situação. Desta forma, mesmo que as trajetórias de alguns objetos se cruzem ou existam objetos sobrepostos numa das imagens, pode-se recorrer à outra para a detecção e rastreamento, existindo assim um menor erro associado ao rastreamento do objeto.

4. ANEXO

De seguida serão demonstrados alguns resultados, correspondentes a casos específicos obtidos para certos *datasets*.

Dataset lab-filinha:

As imagens do *dataset* correspondem a um conjunto de pessoas a entrar e sair de uma sala em fila. Verificou-se que, a certo ponto, quando as pessoas estão muito juntas, a pessoa mais próxima da câmara se sobrepõe, não totalmente, à que está à sua frente (mais distante da câmara), o que faz com que essa pessoa fique definida por dois objetos diferentes.

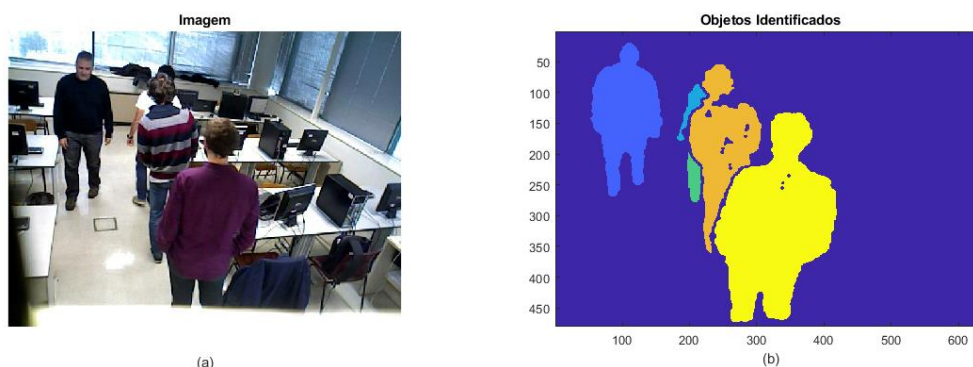


FIGURA 11 – IMAGEM OBTIDA DA CÂMERA (A) E RESPECTIVOS OBJETOS IDENTIFICADOS (B). VERIFICA-SE QUE UMA DAS PESSOAS TEM A MAIORIA DO CORPO TAPADO POR OUTRA, SENDO POR ISSO DIVIDIDA EM 2 OBJETOS DIFERENTES, O SEU BRAÇO E PERNA.

Esta situação provoca um erro quando se está a realizar o rastreamento uma vez que se irão construir duas *point clouds* distintas para a mesma pessoa. Ao rastrear em comparação com a imagem anterior, como não ocorreu a divisão da pessoa nessa imagem, as duas partes da pessoa serão considerados como novos objetos. Obtém-se, portanto, a partir de um objeto, três objetos considerados distintos (quando na verdade são o mesmo), que resulta de um erro que é independente da resolução efetuada para o problema.