Instituto Superior Técnico - UL

MEEC - Processamento de Imagem e Visão

Projeto - Parte 2

Autores:	*IST ID:*
Gonçalo Mestre	87005
Vicente Pinto	87134
Xavier Dias	87136

Professor João Paulo Costeira

---

22 de Dezembro de 2019

Conteúdo

1 Introdução e Definição do Problema

1.1 Sistema Kinect

A Kinect é um sistema, cuja primeira versão foi lançada em 2010, tendo sido desenvolvido pela *Microsoft* em colaboração com a empresa *Prime Sense*. Este sistema consiste na implementação de diversos sensores, incluído uma câmara RGB e uma câmara de profundidade (utilizadas neste projecto) num único produto, representado na Figura 1.

Este sistema permite aos seus utilizadores interagir com o mesmo, através de gestos sem necessidade de instrumentos ou controladores adicionais, tendo sido utilizado em diversos projectos científicos devido à quantidade e qualidade de informação que consegue obter e ao seu baixo custo e facilidade de utilização.

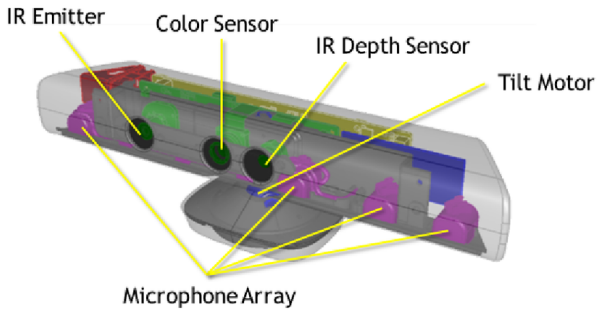


Figura 1: Sensores presentes no sistema Kinect(1ª versão)

Ao permitir recolher imagens de cor(RGB) e imagens de profundidade, como mostra a figura 2, a Kinect permite então construir uma imagem 3D através da combinação destas, sendo possível associar correctamente as coordenadas de cor às coordenadadas de profunidade obtidas pela câmara. De qualquer modo, esta correspondência não é directa, pois as câmaras de profundidade e cor não estão exactamente na mesma posição e não são iguais, logo é necessário considerar os parâmetros extrínsecos (posição relativa das câmaras) e os parâmetros intrínsecos de cada uma destas duas câmaras.

Figura 2: Imagem de cor RGB(à esquerda), imagem em profundidade(à direita)

A imagem 3D obtida através deste processo é uma *point cloud*. Uma *point cloud* consiste na representação de pontos 3D com as coordenadas obtidas através da imagem de profundidade, no ponto de vista da câmara, estando estes pontos munidos com a cor obtida para esse ponto a partir da imagem de cor.

1.2 Definição do Problema

Neste projecto, o objectivo é conseguir a reconstrução de uma cena 3D utilizando imagems de cor e de profundidade obtidas através de uma Kinect em posições diferentes. Para que isto seja possível é necessário obter para cada imagem do respectivo dataset, uma transformação da mesma para o referencial da primeira imagem do dataset. Para resolver este problema, o mesmo foi dividido em várias partes listadas de seguida.

1. Obter as coordenadas de cada ponto em metros, utilizando as imagens de cor e profundidade.
2. Associar as imagens de cor e profundidade, utilizando as coordenadas de cada ponto em metros e a imagem de cor para obter a imagem em rgbd.
3. Associar características (Feature Matching) entre as imagens.
4. Proceder à remoção de outliers de entre as características obtidas no modelo.
5. Obter uma transformação de cada imagem para o referencial da primeira imagem.

2 Resolução do Problema

2.1 Modelo da Câmara e Associação entre imagens de cor e profundidade

Para resolver o problema foi utilizado o modelo da câmara que irá ser explicado de seguida, sendo depois clarificado como é que o mesmo foi aplicado.

Uma câmara pode ser considerada como um modelo, descrito por uma matrix, que mapeia pontos 3D ($\mathbf{X} = \begin{bmatrix} X & Y & Z \end{bmatrix}^T$) para o plano da imagem ($\mathbf{x} = \begin{bmatrix} x & y \end{bmatrix}^T$), como representado na Figura 3.

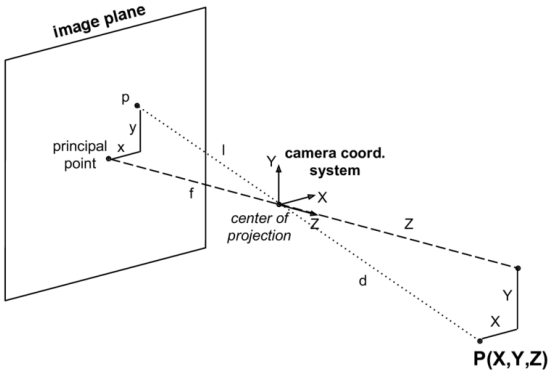


Figura 3: Modelo da Câmara

Considerando a câmara normalizada, ou seja, com a distância focal (distância entre o centro óptico da câmara e o plano da imagem) igual a 1, a projecção em prespectiva é dada pela equação 1.

$$x = \frac{X}{Z}, y = \frac{Y}{Z} \tag{1}$$

Ou em coordenadas homogéneas pela equação 2.

$$\lambda \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{2}$$

O modelo da câmara completo é obtido tendo em conta dois tipos de parâmetros:

- parâmetros extrínsecos: a posição da câmara em relação ao referencial do mundo
- parâmetros intrínsecos: distância focal, factores de escala e o ponto principal da câmara

Para o modelo interno da câmara é definido um novo sistema de coordenadas, como se pode ver na figura 4, usando uma transformação 2D em que a conversão é feita de metros para pixels, utilizando o sistema de equações dado por (3) e (4).

$$x' = f s_x x + c_x \tag{3}$$

$$y' = f s_y y + c_y \tag{4}$$

Em coordenadas homogéneas a transformação é dada pelo sistema matricial representado em (5).

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} fs_x & 0 & c_x \\ 0 & fs_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{5}$$

Onde:

- $\mathbf{x} = \begin{bmatrix} x & y \end{bmatrix}^T$ está em metros
- $\mathbf{x'} = \begin{bmatrix} x' & y' \end{bmatrix}^T$ está em pixels
- f é a distância focal
- c_x e c_y são as coordenadas do ponto principal em pixels
- s_x e s_y são os factores de escala nas direções x e y, respectivamente, em pixels/metro

A figura4 representa o modelo internto.

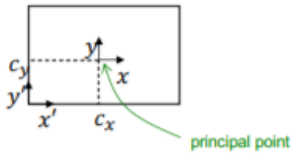


Figura 4: Modelo Interno

Para o modelo externo é definido um novo sistema de coordenadas (o sistema de coordenadas do mundo) por uma transformação de coordenadas 3D dada pela equação (6). Este sistema de coordenadas também pode ser observado na Figura 5.

$$\mathbf{X} = R\mathbf{X'} + \mathbf{T} \tag{6}$$

A transformação também se tem em coordenadas homogéneas dada pela equação (7).

$$\tilde{\mathbf{X}} = \begin{bmatrix} R & \mathbf{T} \\ \mathbf{0}^T & 1 \end{bmatrix} \tilde{\mathbf{X}}' \tag{7}$$

Onde:

- $\mathbf{X}$ são as coordenadas no referencial da câmara
- $\mathbf{X}$' são as coordenadas no referencial do mundo
- R $\in \mathbb{R}^{3x3}$ é uma matriz de rotação 3D que expressa a relação entre as coordenadas do mundo e da câmara

- $\mathbf{T} \in \mathbb{R}^3$ é um vector de translação 3D que representa a origem do sistema de coordenadas do mundo expressa nas coordenadas da câmara

A figura 5 representa o modelo externo.

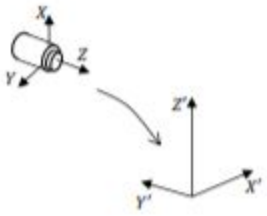


Figura 5: Modelo Externo

Juntando estes dois modelos obtidos tem-se o modelo completo da câmara dado pela equação (8).

$$\lambda \tilde{\mathbf{x}}' = K \begin{bmatrix} R & \mathbf{T} \end{bmatrix} \tilde{\mathbf{X}} \tag{8}$$

Através disto é possível associar a imagem em profundidade à imagem de cor, começando por usar os parâmetros intrínsecos da câmara de profundidade e os valores de profundidade para criar um espaço 3D, fazendo o cálculo inverso da equação (5) e da equação (1).

Estando-se então no espaço 3D da imagem de profundidade, de seguida faz-se uma transformação para o referencial 3D da imagem RGB usando os parâmetros extrínsecos da câmara Kinect. Neste espaço, as coordenadas X, Y e Z mapeiam directamente para os indíces (píxeis) da imagem RGB. Um esquema a explicar as operações feitas, pode ser observado na figura 6.

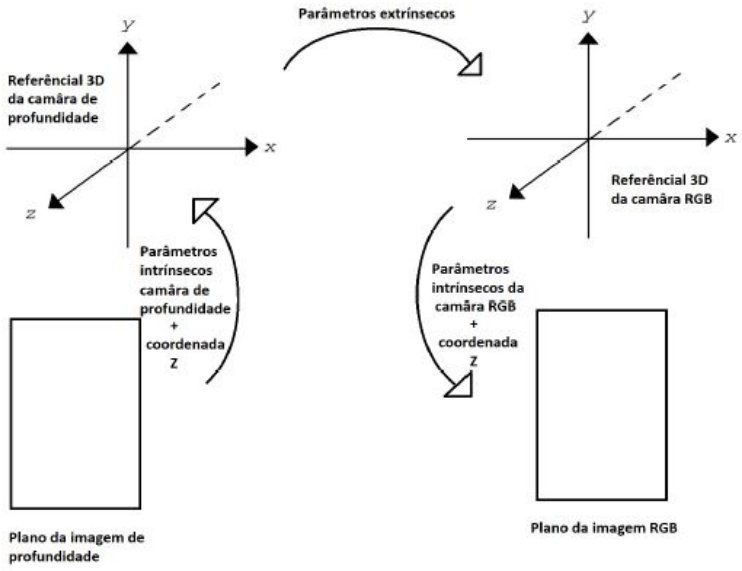


Figura 6: Mudança de coordenadas do eixo da câmara de profundidade para o eixo da câmara RGB

2.2 Associação de Características

Depois de ser obtida a associação entre as imagens de cor e as imagens de profundidade, é necessário fazer a associação de características entre imagens para que se possa obter uma transformação entre estas.

Para que isto fosse feito, para cada imagem N faz-se a associação de características com as imagens de 1 a N-1 utilizando o algoritmo SURF implementado pelo Matlab.

Esta associação é feita entre as imagens 1 a N-1 pois é sempre mais vantajoso obter uma transformação directa da imagem N para a imagem 1, de modo a evitar a propagação do erro. Mas isto nem sempre é possível, sendo nos datasets utilizados apenas garantidas características suficientes para associação entre a imagem N e a imagem N-1.

Assim, é feita a associação de características da imagem N para as restantes imagens, sendo no fim depois da obtenção da transformação comparadas as transformações de modo a escolher a melhor opção de entre as obtidas.

2.3 Extração de Outliers e Random Sample Consensus

Depois de ser feita a associação de características é necessário eliminar associações que tenham sido estabelecidas incorrectamente, sendo estas denominadas de outliers.

Para fazer esta extração de outliers e se obter apenas associações de características correctas é utilizado um algoritmo adequado, sendo neste caso utilizado o Random Sample Consensus (RANSAC).

O RANSAC é um método iterativo para estimar os parâmetros de um modelo matemático através de um conjunto de observações que contém outliers (observações que não deveriam influenciar os valores do modelo estimado).

O algoritmo funciona seguindo os referidos passos:

1. Escolher o modelo usado para estimar os dados.

   Para este caso foi utilizado o modelo de um corpo rígido: $p_1 = Rp_2 + T$.

2. Aleatoriamente escolher o número de pontos necessários para estimar os parâmetros do modelo de entre os pontos obtidos na associação de características.

   Para este caso são utilizados 4 pontos, pois o modelo de um corpo rígido tem 12 parâmetros a estimar e cada ponto gera 3 equações, o que explica a necessidade destes 4 pontos.

3. Estimar os parâmetros do modelo, usando os 4 pontos gerados aleatoriamente.

   Para este caso estes parâmetros são estimados resolvendo o problema de *Procrustes*.

4. Verificar que elementos dos pontos obtidos na associação de características são consistentes com o modelo estimado. Um ponto é considerado inconsistente com o modelo estimado se não corresponder ao modelo dentro de um limite máximo de erro definido. Este limite máximo de erro corresponde ao desvio máximo atribuído ao efeito do ruído.

Para este caso o erro é calculado fazendo a norma do valor real do ponto menos o valor estimado pelo modelo. Se o valor for superior ao limite definido, o ponto será considerado *outlier*.

5. Determinar o número de pontos que são consistentes com o modelo estimado.

6. Repetir os passos 2, 3, 4 e 5 para o número de iterações definido.

   Para este caso foram utilizadas 200 iterações.

7. O modelo estimado é razoavelmente bom se forem classificados suficientes pontos como inliers. Assim é escolhida a iteração do ransac em que foi obtido um maior número de inliers e são utilizados esses inliers para estimar o modelo.

2.4 Problema de Procrustes

O problema de procrustes é um metodo que permite obter a rotação óptima para a superimposição de Procrustes de um objecto em relação a outro. Este algoritmo assume que há dois conjuntos de pontos correspondentes p_1 e p_2, para as imagens 1 e 2, respectivamente. Estes pontos são associados através da transformação da equação 9.

$$p_1 = Rp_2 + T \tag{9}$$

Este problema foi resolvido em 1964 numa tese da autoria de Peter Schönemann e a solução obtida utiliza a decomposição em valores singulares (SVD) para chegar à solução do problema que resulta na estimativa de mínimos quadrados do erro.

Assumindo que a translação entre cada par de pontos correspondente é igual à translação entre os centróides dos conjuntos, é possível escrever a equação (10).

$$T = centroide_1 - R * centroide_2 \tag{10}$$

Sendo definidos os centroides 1 e 2 pelas equações (11) e (12).

$$centroide_1 = \frac{1}{N}\sum_{i=1}^{N} p_{1i} \tag{11}$$

$$centroide_2 = \frac{1}{N}\sum_{i=1}^{N} p_{2i} \tag{12}$$

Assim, tendo-se os pontos e a matriz de rotação é possível descobrir o vector de translação. Logo para descobrir a matriz de rotação é resolvido o problema escrevendo a equação (13) que representa os pontos na imagem 1 retirando-lhes a coordenada do centroide, sendo este o centro do referencial. Esta equação é de seguida simplificada, resolvendo-se ao obter a equação (14).

$$p_1 - centroid_1 = R * p_2 + T - R * centroid_2 - T \tag{13}$$

$$p_1 - centroid_1 = R * (p_2 - centroid_2) \tag{14}$$

Assim, tendo-se N pontos, e representando as matrizes A e B os pontos na imagem 1 e 2, respectivamente, já subtraido o valor do respectivo centróide, é possível escrever de um modo mais simples a equação (14) na equação (15)

$$A = RB \tag{15}$$

Logo o problema que se pretende resolver agora será escolher a matriz R que minimiza o erro dado por (16).

$$||A - R * B||^2 \tag{16}$$

É então para resolver esta parte do problema que se usa a decomposição em valores singulares, sendo a mesma descrita na equação (17).

$$\begin{bmatrix} U & \Sigma & V \end{bmatrix} = svd(A * B^T) \tag{17}$$

Depois de feita a decomposição em valores singulares, já é possível obter a matriz de rotação R através da equação 18 e depois desta matriz, o vector de translação T, utilizando a equação (10).

$$R = U * V^T \tag{18}$$

Assim, está resolvido o problema de Procrustes, mas é de notar um factor bastante importante, ou seja, que para resolver este problema é necessário pelo menos 4 pontos de correspondência entre as duas imagens. Estes pontos, não podem ser quaisquer pontos, tendo estes que ser 4 pontos não coplanares.

Para resolver os casos em que apenas são obtidos pontos coplanares, pode-se assumir por exemplo a coordenada z igual a 0, logo escolhendo este plano, obtêm-se as duas primeiras colunas da matriz de rotação através do mesmo calculo efectuado na equação 18. Para a terceira coluna é necessária uma coluna que faça a transformação entre o eixo z da imagem 2 e o eixo z da imagem 1. Assim esta coluna será dada pelo produto externo das duas primeiras colunas, pois assim já se tem a componente perpendicular ao plano, correspondendo esta ao eixo z.

Depois de obtida a matriz de rotação, o vector de translação será obtido da mesma forma.

É de notar que a deteção da presença de apenas pontos coplanares dentro dos pontos utilizados, pode ser feita verificando o último valor próprio da matriz Σ obtida na equação 17, sendo que quando este apresenta um valor bastante reduzido, se está na presença de pontos coplanares.

Um outro caso que poderia gerar problemas é quando a matriz R obtida em vez de ser uma matriz de Rotação é uma matriz de Reflexão, pois a solução deste problema Procrustes nalguns casos gera uma e noutros gera outra.

Como uma matriz de Rotação não é mais que uma transformação de eixos em que transforma x_2 em x_1, y_2 em y_1 e z_2 em z_1, uma matriz de Reflexão não é muito diferente, mudando que em vez de trasnformar z_2 em z_1 transforma em $-z_1$, isto assumindo que o eixo reflectido é o eixo z.

Assim para resolver este problema, é verificado o determinante, que caso seja negativo, aproximadamente -1, implica que se está perante uma matriz de reflexão. Após esta verificação, estando-se perante uma matriz de reflexão apenas é necessário reflectir a última coluna, ou seja, multiplicar por -1 as entradas da última coluna matriz obtida.

2.5 Extração da Transformação

Nesta subsecção pretende-se mostrar apenas como é obtida a transformação da imagem N para o referencial da imagem 1, quando a correspondência das características é feita entre a imagem N e a imagem m, sendo $1 < m < N$ e conhecendo-se a transformação da imagem m para o referencial da imagem 1.

O problema é resolvido utilizando as equações (19) e (20), obtendo-se como resultado a equação (21).

$$X_1 = R_{1m}X_m + T_{1m} \tag{19}$$

$$X_m = R_{mN}X_N + T_{mN} \tag{20}$$

$$X_1 = R_{1m}R_{mN}X_N + R_{1m}T_{mN} + T_{1m} \tag{21}$$

Assim para resolver o problema no caso referido obtém-se a transformação da imagem N para o referencial da imagem m e de seguida utiliza-se essa transformação e a transformação já conhecida da imagem m para o referencial da imagem 1, para no fim se obter a transformação da imagem N para o referencial da imagem 1. Logo a resolução do problema o se obter a matriz de rotação da imagem N para a imagem 1 e o vector de translação da imagem N para a imagem 1 é dada pelas equações (22) e (23).

$$R = R_{1m}R_{mN} \tag{22}$$

$$T = R_{1m}T_{mN} + T_{1_m} \tag{23}$$

2.6 Escolha da Transformação

Tal como explicado na subsecção 2.2, é feito o cálculo da transformação da imagem N para a imagem 1 fazendo a associação de características entre a imagem N e as imagens 1 a N-1.

Assim, no fim de calculada a transformação utilizando cada uma das diferentes associações de características, é escolhida a primeira transformação das obtidas que mantenha uma diferença não demasiado grande para a transformação que se obtém utilizando a associação de características entre as imagens N e N-1.

Isto permite garantir duas questões importantes para o problema. A primeira é que se tenta utilizar a primeira imagem que permita obter inliers suficientes, para fazer associação de características com a imagem N. A segunda é que se garante que a transformação utilizada não será demasiado diferente da obtida utilizando imagens consecutiva. Isto é relevante, pois apenas é garantido haver características suficientes para fazer associação entre imagens consecutivas, não sendo garantido para os restantes casos.

3 Resultados Experimentais

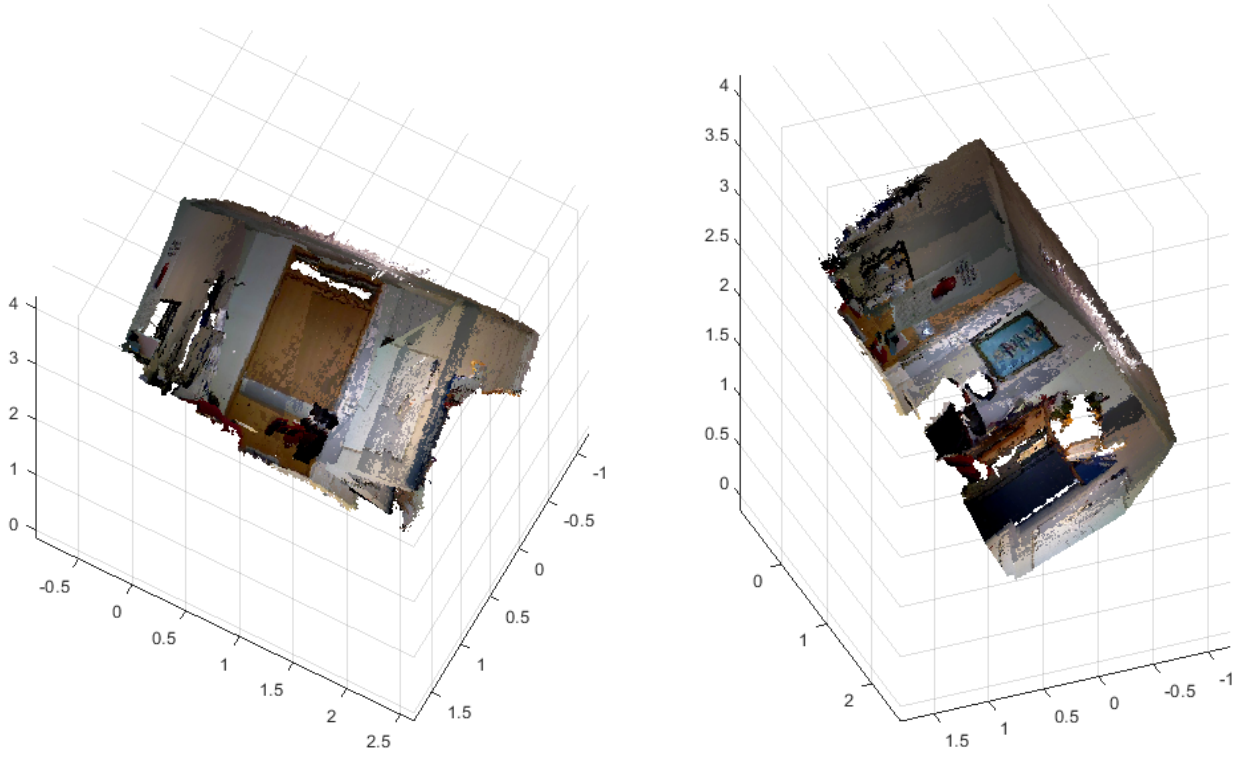
Com vários conjuntos de imagens de cor e profundidade de teste fornecidos, conseguiu-se visualizar a aplicação do algoritmo. No final de todas os ajustes possíveis, obtiveram-se boas e más reconstruções, pois uma reconstrução depende do que se é dado. Nesta secção ir-se-á demonstrar as reconstruções para ambos os casos e comentá-los.

3.1 Dataset `room1`

Este dataset foi dos mais simples para se fazer a reconstrução, tendo sido aquele com que o algoritmo foi inicialmente testado e com que foram corrigidos os primeiros erros de implementação.

O dataset não levantou problemas, tendo sido originada uma reconstrução bastante semelhante à cena real, apenas com um erro pequeno na posição de uma ou duas pointClouds, por exemplo um desalinhamento muito pequeno na porta, como é possível observar na Figura 9a.

Isto está dentro do esperado, pois ao calcular as transformações há sempre um ligeiro erro presente e a propagação do mesmo nas últimas imagens de cada dataset é invevitável, tendo sido mantido ao mínimo com os procedimentos explicados nas secções 2.2 e 2.6.



(a) (b)

Figura 7: Reconstrução 3D de uma sala com recurso a imagens de `room1`

3.2 Dataset `board1`

Este dataset é referente a um quadro que como se nota pelas imagens tem as imagens tiradas todas referentes a um mesmo plano. Isto acrescenta uma dificuldade adicional como explicado na secção 2.4.

Numa fase mais embrionária do projecto antes de se resolver o problema de Procrustes considerando este caso era obtida uma pointcloud como a demonstrada na Figura 8.

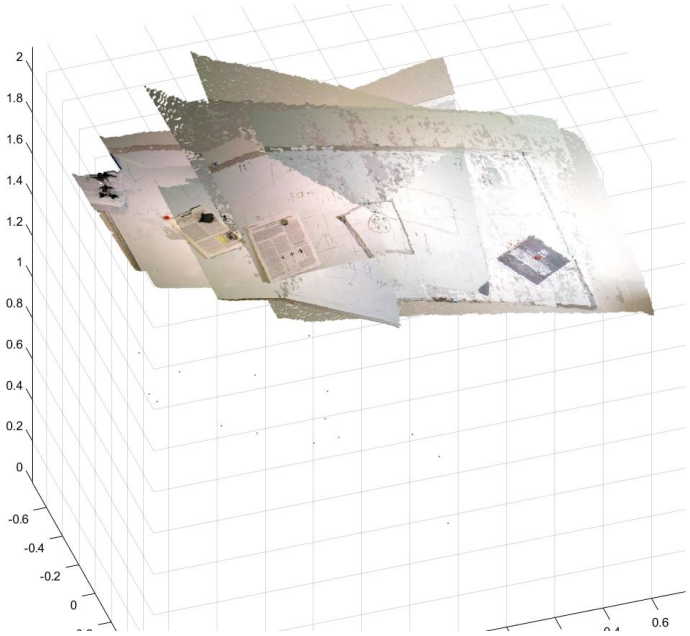
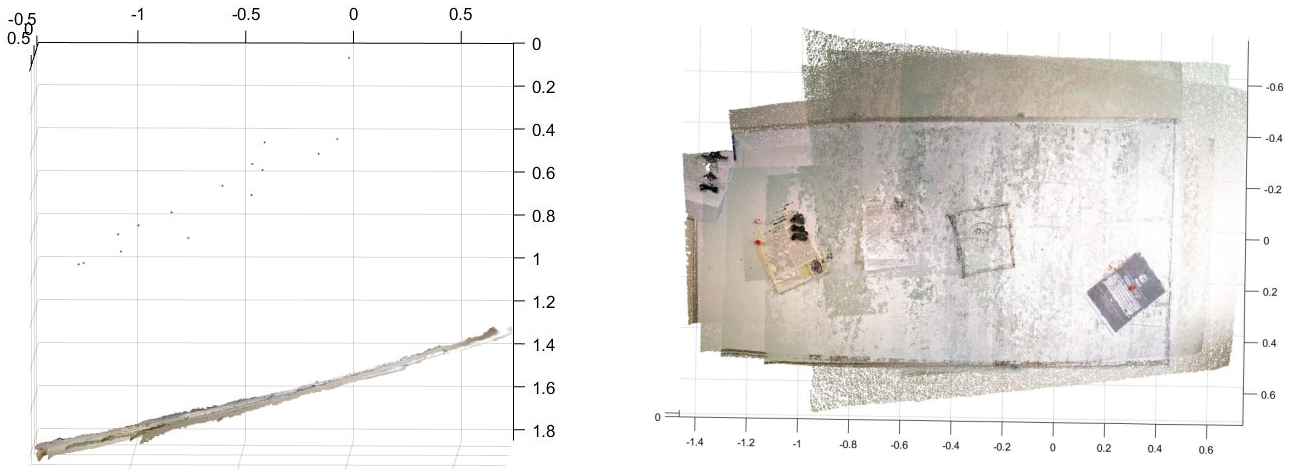


Figura 8: Reconstrução 3D de um quadro recorrendo a imagens do conjunto `board1`, antes da correção para pontos coplanares

Após ter sido resolvido o problema como explicado na secção 2.4 e se ter passado a calcular a matriz de rotação do modo correcto para estes cojuntos de pontos é obtida a pointcloud representada na Figura 9.



(a) Perspectiva de Cima

(b) Perspectiva de Frente

Figura 9: Reconstrução 3D de um quadro com recurso a imagens de `board1`

3.3 Dataset `labpiv`

Ao analisar este dataset, é possível reparar que o mesmo é bastante extenso, apresentado 23 imagens. Este dataset apresenta um conjunto de dificuldades diferente dos anteriores, pois ao testar o mesmo, existiam imagens em que a associação de características era apenas feita com pontos coplanares (obrigando a resolver este caso) e porque apresenta uma extensão considerável.

Esta extensão considerável torna bastante grande a fonte de erro, pois quando se está numa imagem muito distante da 1, já não há características suficientes para a associação ser feita com uma das primeiras imagens. Isto gera uma propagação de erro grande devido a ter que se usar transformações anteriores para calcular a transformação das últimas imagens para o referencial da primeira. Algo que poderia diminuir este erro seria por exemplo calcular a transformação de cada imagem não para a primeira, mas para uma das do meio.

A reconstrução 3D está apresentada na Figura 10 e é possível de notar que apesar de estar bastante boa para o dataset em questão, apresenta algum erro, sendo possível notar por exemplo monitores um pouco deslocados da posição correcta, ou ligeiramente tortos.

De qualquer modo, considera-se esta reconstrução bastante boa, tendo em conta a dificuldade acrescida deste dataset.

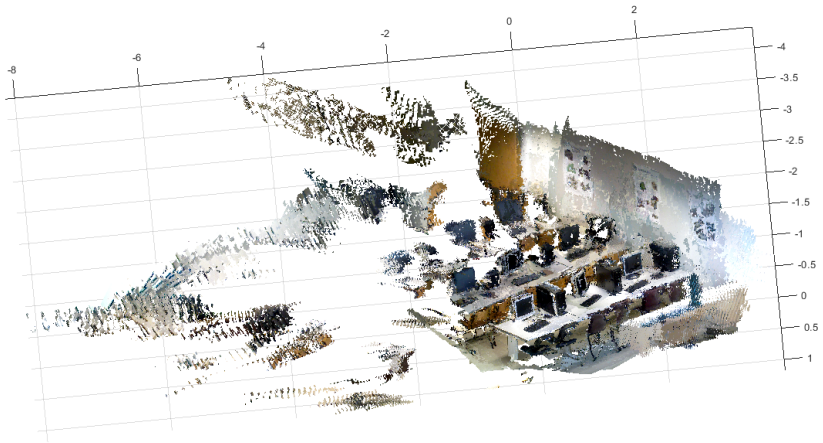
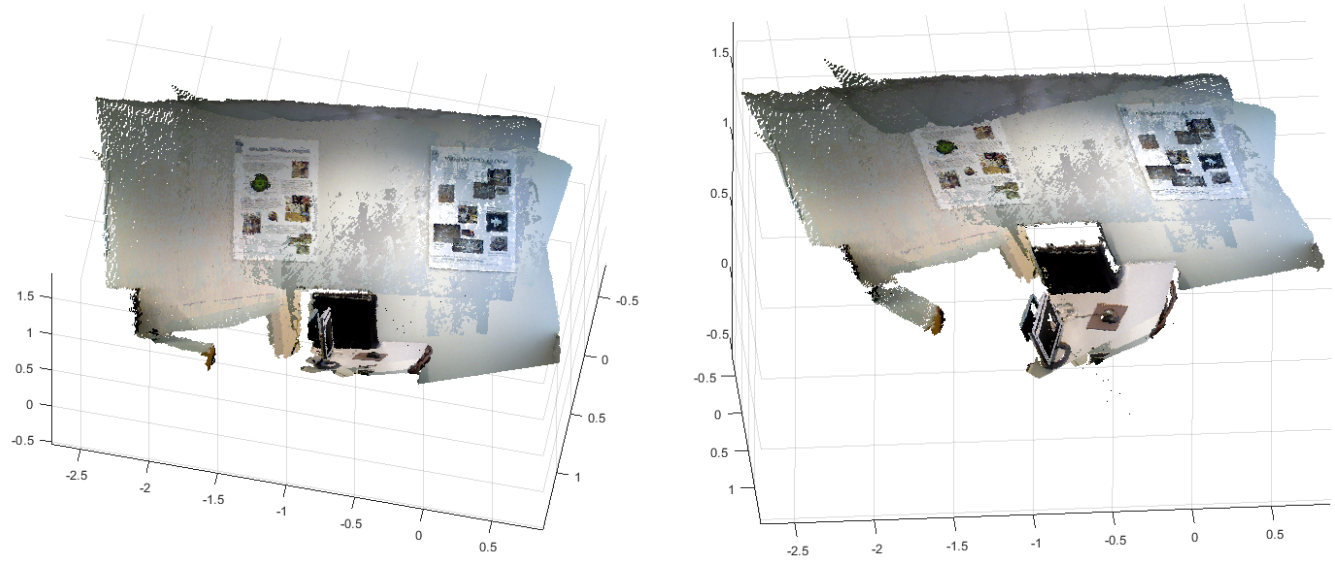


Figura 10: Reconstrução 3D do laboratório de PIV recorrendo a imagens do conjunto `labpiv`

3.4 Dataset `malandreco`

Este dataset foi um dos mais simples, tendo funcionado sem problemas, sendo poss+ivel notar uma boa reconstrução e estando todas as imagens correctamente representadas. Os resultados podem ser observados de duas perspectivas diferentes na Figura 11.



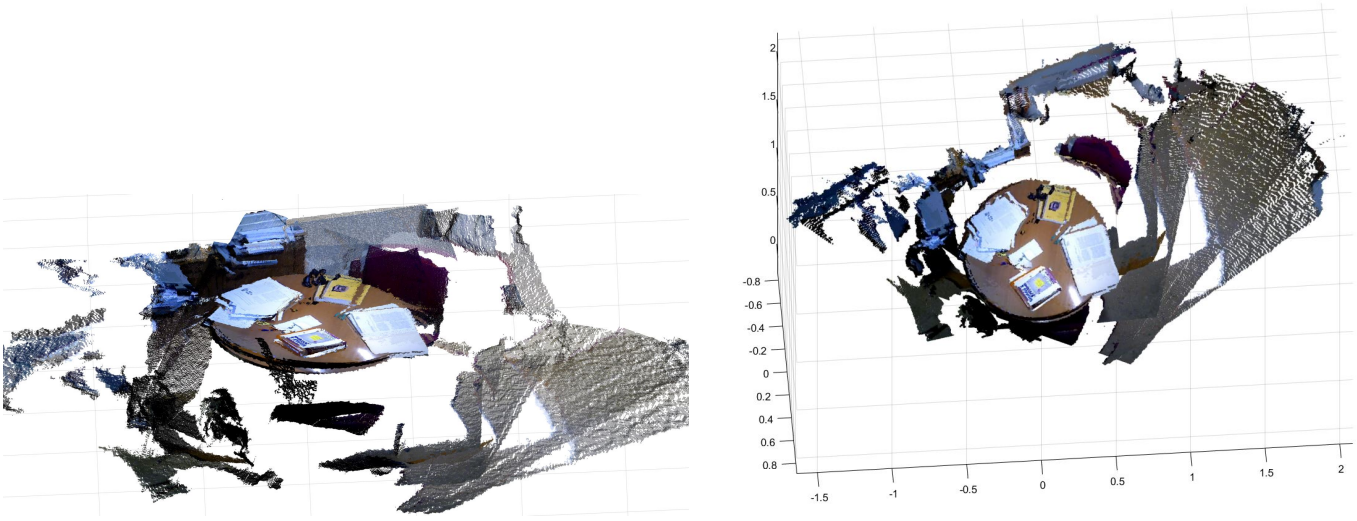
(a) (b)

Figura 11: Reconstrução 3D de computador e uma parede com recurso a imagens de `malandreco`

3.5 Dataset `table`

O *dataset* `table` contém imagens que contêm muitos objetos à volta do principal que é a mesa. Pelo facto de as imagens terem sido tiradas à volta da mesma parte da sala, a mesa. Isto cria uma dificuldade acrescida na reconstrução do background, pois as características identificadas vão estar maioritariamente em cima da mesa, tornando por exemplo o chão bastante mais complicado de descrever. É possível de afirmar que as transformações se tornam "overfitted"aos pontos em cima da mesa.

Na Figura 12 é possível observar duas perspectivas desta reconstrução.



(a) Perspectiva de frente (b) Perspectiva de cima

Figura 12: Reconstrução 3D de uma mesa com recurso a imagens de `table`

3.6 Dataset `newpiv2`

O dataset `newpiv2` contem imagens de uma sala, em que há um objecto bastante grande, sendo o mesmo mudado de sítio em diferentes imagens. Numa fase inicial do problema, antes de ser implementado o descrito nas secções 2.2 e 2.6, a reconstrução obtida era a representada na Figura 13.

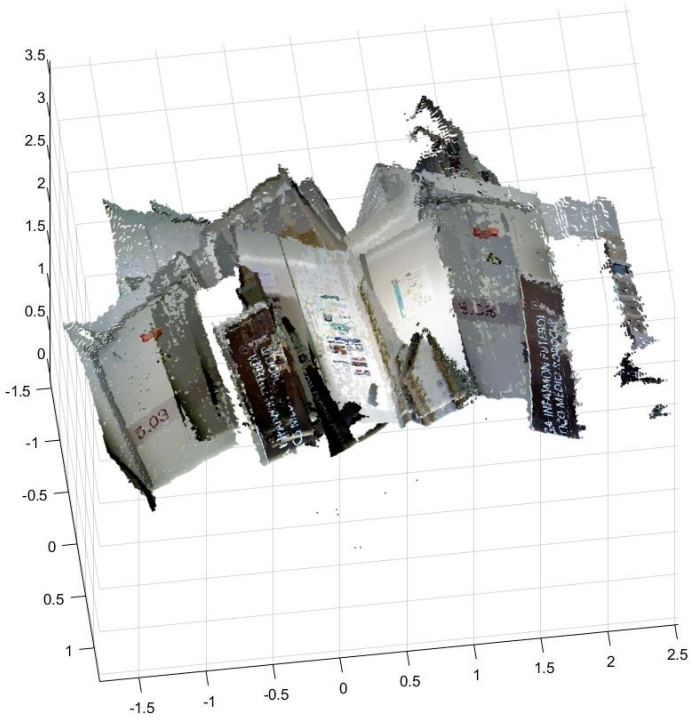


Figura 13: Primeira reconstrução 3D obtida utilizando as imagens de `newpiv2`

O que acontece neste caso é que o algoritmo está a detetar, nalgumas correspondências, pontos somente pertencentes ao objeto referido e como o mesmo muda de sítio a reconstrução do background fica completamente destruida. Uma imagem com as correspondências que geravam problemas pode ser observada na Figura 14.

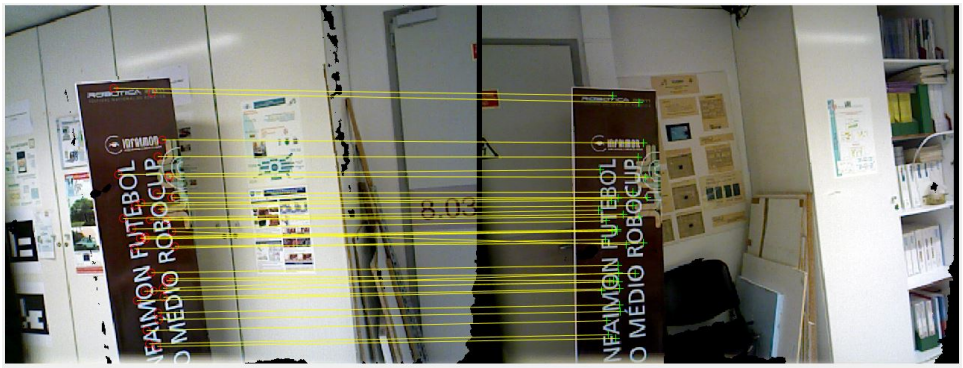
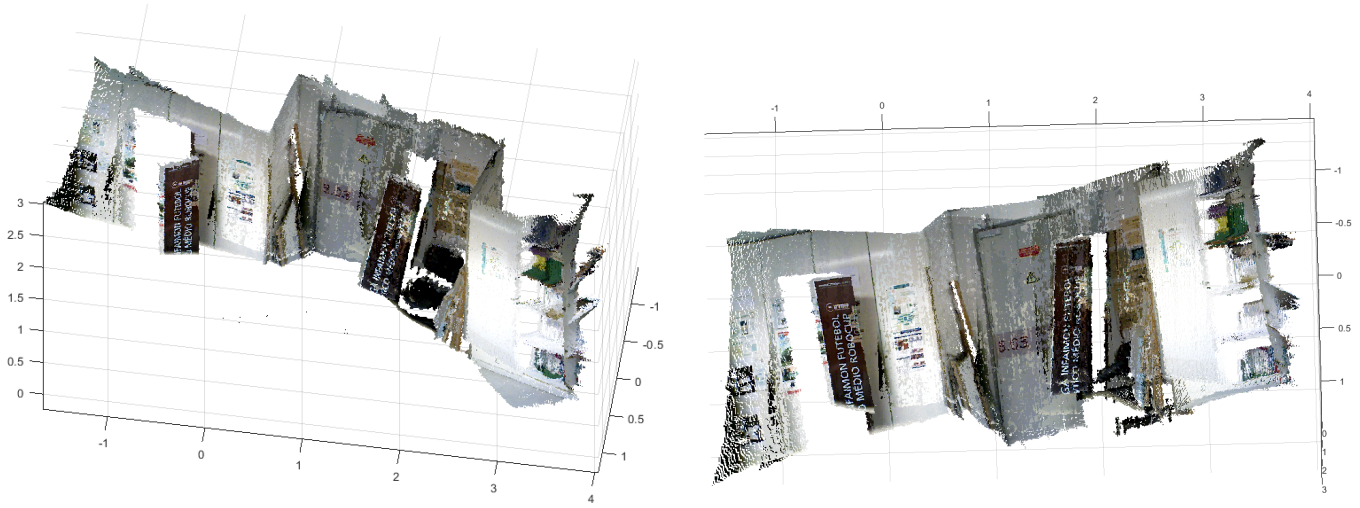


Figura 14: Correspondências entre duas imagens do dataset `newpiv2` que geram problemas na reconstrulção 3D

Após ser alterado o algoritmo para fazer o indicado nas secções 2.2 e 2.6 tem-se a reconstrução 3D present na Figura 15.



(a) (b)

Figura 15: Reconstrução 3D de paredes com recurso a imagens de `newpiv2`

3.7 Dataset `office`

Para este dataset não é possível obter uma reconstrução correcta, tendo sido obtida a reconstrução presente na Figura 16, isto acontece pois o mesmo é bastante mau.

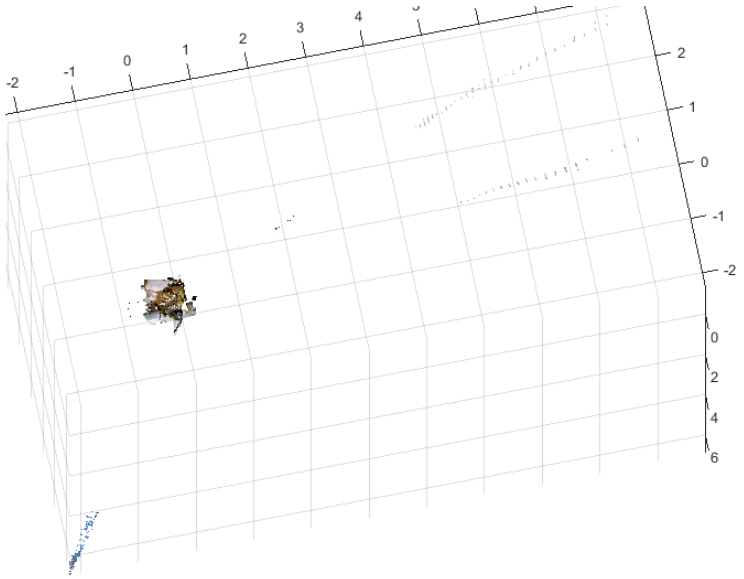


Figura 16: Reconstrução 3D do conjunto `office`

Este dataset apresenta uma quantidade de zeros bastante grande nas matrizes de profundidade o que faz com que as imagens rgbd fiquem completamente inutilizáveis e inviabiliza a obtenção de uma reconstrução

3D razoável.

Isto deve-se ao facto de existirem maioritariamente imagens com o candeeiro, que como já foi explicado anteriormente, reflete indesejavelmente os raios infravermelhos do sensor de profundidade o que tem impacto nas matrizes de profundidade das imagens. Para verificar isto é apresentada na Figura 17 uma imagem rgbd dentro das que se obtém com este conjunto.

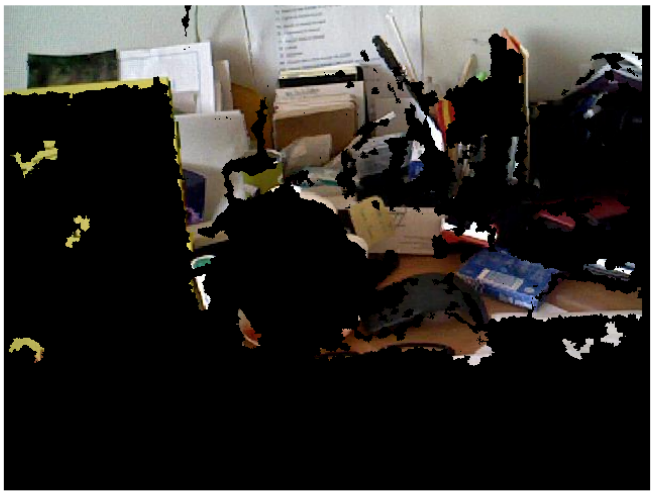


Figura 17: Uma das imagens RGBD obtida para o conjunto `office`