****

**Processamento de Imagem e Visão**

**Tracking de objetos através de imagens RGB-D**

Grupo X

Francisco Azevedo, nº 80966

Francisco Pereira, nº 81381

Luís Almeida, nº 81232

1. **Introdução**

Um dos problemas mais relevantes em processamento de imagem é a deteção de objetos. A deteção de objetos consiste em reconhecer e segmentar regiões de uma imagem que fornecem informação relevante, quer esta seja um objeto que se move, um carro parado no estacionamento ou outro objetivo prático.

O projeto de Processamento de Imagem e Visão proposto tem como objetivo identificar objetos e o seu movimento ao longo de uma sequência de imagens captadas por duas câmaras estáticas em localizações distintas capazes de obter informação sobre cor e profundidade.

Uma sequência de imagens é um conjunto de *frames*. Por cada *frame* existem 4 imagens: duas por câmara. A informação é adquirida por uma câmara *Kinect* que adquire e armazena duas imagens por *frame*: uma representa cor a outra profundidade.



Figure a - a) imagem rgb e b) imagem depth

Na Figura 1 mostra-se um exemplo da imagem de cor (*rgb*) e de profundidade (*depth*) obtidas com o dispositivo. A imagem de profundidade indica, para cada pixel, qual a distância entre o plano da imagem da câmara – *image plane* - e o objeto correspondente na imagem *rgb*.

A formulação do problema é: o primeiro objetivo é identificar os objetos em cada câmara. De seguida é necessário representar estes objetos em três dimensões com a informação fornecida. Estas representações são no sistema de coordenadas de cada câmara logo é necessário transformar ambas as representações no mesmo sistema de coordenadas. Sabendo que as representações estão no mesmo sistema de coordenadas é preciso obter um consenso sobre que objetos foram detetados neste sistema. Por último é necessário fazer a correspondência de objetos entre frames (*tracking*).

1. **Modelo da câmara**

O modelo de câmara permite representar pontos em três dimensões num plano em duas dimensões sendo que o modelo matemático que expressa esta relação baseia-se no modelo *pin-hole.*

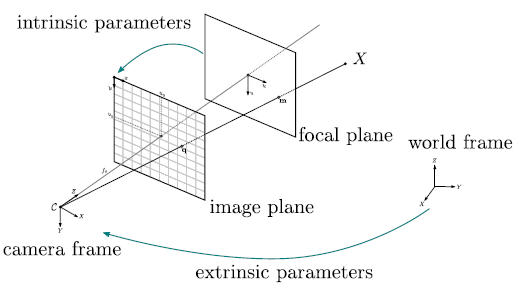


Figura 2 - Modelo da câmara pin-hole

É importante definir como representamos os pontos 3D em 2D e como se transforma pontos 3D em diferentes sistemas coordenadas pois o objetivo é identificar objetos em 3D tendo como ponto de partida imagens de cor e profundidade.

Este modelo utiliza parâmetros intrínsecos que permitem a conversão de coordenadas métricas em pixéis e parâmetros extrínsecos que definem a localização e orientação da câmara em relação a um sistema de coordenadas externo – referencial do mundo. A Figura 2 ilustra o modelo anterior.

Um ponto pertencentea projetado no plano de imagem terá coordenadas dadas pelas expressões da projeção em perspetiva

ou, em coordenadas homogéneas

A matriz intrínseca, que inclui os parâmetros intrínsecos converte pontos 3D (em coordenadas da câmara) em pontos 2D (em coordenadas homogéneas). Os parâmetros intrínsecos realizam a conversão de coordenadas métricas para pixéis que é obtida através das equações

onde:

* representam um ponto, em metros
* representam a localização do ponto no plano da imagem, em pixel
* é a distância entre o centro ótico e o ponto de interseção do eixo ótico com o plano de imagem – *principal point –* em metros.
* e são fatores de escala nas direções em pixel/m
* e são as coordenadas do *principal point*, em pixel

A transformação em coordenadas homogéneas mostra-se a seguir, em que a matriz inclui os parâmetros intrínsecos.

Os parâmetros extrínsecos representam a transformação entre os referenciais da câmara e do mundo e são definidos pela seguinte expressão através das matrizes *R* e *T*:

onde:

* é um ponto expresso no referencial da câmara.
* é um ponto expresso no referencial externo.
* é uma matriz de rotação que expressa a rotação entre o referencial externo e o referencial da câmara. é uma matriz ortogonal que satisfaz

* é um vetor de translação que expressa a origem do referencial do mundo no referencial da câmara.

Caso o referencial do mundo não coincida com o referencial da câmara, é necessário primeiro projetar-se as coordenadas 3D do ponto no referencial da câmara para depois se poder obter a representação do ponto no plano de imagem, em pixel. Assim, ao generalizar-se o modelo define-se uma matriz – matriz da câmara que inclui os parâmetros intrínsecos e extrínsecos da seguinte forma:

sendo um ponto representado no plano da imagem através da expressão

1. **Introdução teórica dos Métodos e Algoritmos utilizados**

Nesta secção introduzem-se os algoritmos utilizados juntamente com análise da teoria subjacente. Antes de mais, embora não seja um algoritmo, é necessário explicar como representar um objeto em 3D através de imagens de profundidade. As imagens de profundidade são indexadas pelas colunas e linhas (*x* e *y*) e a profundidade *f(x,y) = z* é o valor da função respetiva a esse indíce. Logo basta utilizar esse valor como mais uma coordenada e projetá-lo de perpendicular face ao plano *xy* obtendo assim o objeto em 3D.

Na parte 2 deste projeto não são fornecidas as matrizes de rotação e de translação entre os referenciais das duas câmaras pelo que é essencial a sua determinação de modo a ter-se informação dos objetos num referencial externo. Este referencial externo coincide com o referencial de uma das câmaras – câmara 1 - sendo que a matriz de rotação do referencial do mundo para o referencial da câmara referencial é a matriz identidade de dimensão 3x3 e o vetor de translação que representa a origem do referencial do mundo no referencial da câmara é composto por zeros. No cálculo da matriz extrínseca entre a câmara 2 e o referencial do mundo necessário obter-se pelo menos 4 pares de pontos coincidentes entre as duas imagens. Uma vez que existem 12 incógnitas e a cada par de pontos correspondem 3 equações, 4 pares de pontos são suficientes para resolver o sistema de equações.

A escolha dos pontos coincidentes é feita com base num algoritmo de deteção e correspondência de *keypoints* denominado *SIFT* – *Scale Invariant Feature Transform* – cujos diferentes passos são explicados em baixo.

**Scale Invariant Feature Transform (SIFT)**

A escolha de pontos de interesse – *keypoints* - numa imagem permite recolher informação relevante que pode ser utilizada para a identificação da mesma região noutra imagem. O algoritmo *SIFT* é uma ferramenta mais robusta em relação ao algoritmo *Harris Corner detection* uma vez que este último apenas deteta *keypoints* correspondentes entre duas imagens se existir uma rotação entre os pontos, não sendo útil quando existe mudança de escala entre as imagens.

Uma das características diferenciadoras de um *keypoint* é o facto de corresponder a regiões com textura, onde existem variações significativas de intensidade em ambas as direções na sua vizinhança. Um canto é um bom exemplo de um *keypoint*, porém ao aumentar-se a resolução da imagem a que pertence o canto, se se escolher a mesma vizinhança usada para identificar o canto do primeiro caso, apenas se identifica um contorno que não corresponde a uma região diferenciadora sendo que não existe *match* entre as duas regiões de cada imagem. A Figura 3 ilustra o problema descrito.

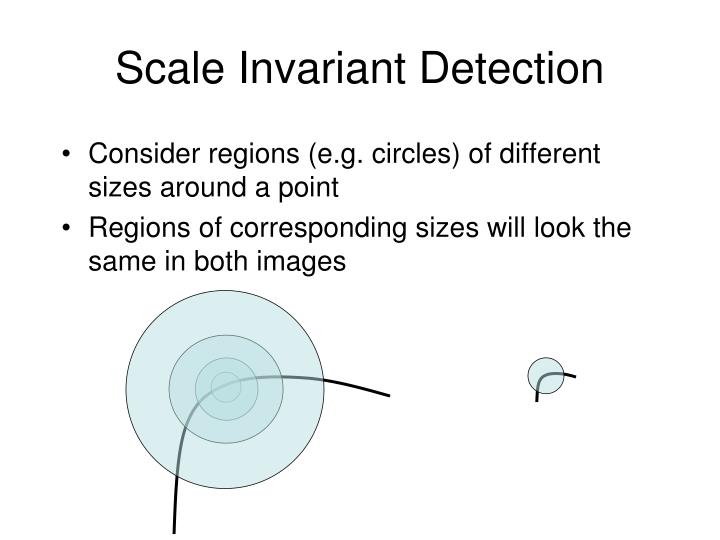


Figura 3 - Variação de escala na deteção de keypoints

A deteção de *keypoints* é feita aplicando-se para diferentes tamanhos da imagem e diferentes tamanhos da vizinhança – janela de procura - uma sequência de filtros gaussianos que corresponde à Diferença de Gaussianas ) sendo que o filtro gaussiano é dado pela convolução de com sendo dado pela expressão

onde são as coordenadas de um pixel da imagem . Ao encontrar-se um extremo local para um conjunto de valores obtém-se uma lista de potenciais *keypoints* identificados pela vizinhança de para a escala

Uma vez identificados os candidatos a *keypoints* aplicam-se alguns métodos que tornam a escolha mais válida. O primeiro passo baseia-se na eliminação de contornos uma vez que a função apresenta valores elevados para regiões de variações elevadas apenas numa direção. Usando um método semelhante ao efetuado no algoritmo *Harris corner detection,* determina-se se se está na presença de uma região onde existe variação apenas numa direção. Este facto é indicativo de uma zona correspondente a um contorno, sendo este uma escolha de *keypoint* inválida*.*

A invariância à orientação tem de ser também verificada pelo que é o algoritmo SIFT inclui um método de análise do gradiente da imagem na vizinhança dos candidatos a *keypoint* que permite a sua identificação quando existe uma rotação na imagem. Na vizinhança do *keypoint* é calculado um histograma que determina o número de pixéis cuja orientação corresponde a um dado *bin* do eixo horizontal. Escolhendo-se 4 blocos de 4x4 pixéis nessa vizinhança obtém-se o histograma de orientação com apenas 8 *bins.* Através da informação do histograma cria-se um vetor denominado descritor do *keypoint* sendo composto por 128 valores. A Figur4 é uma representação gráfica deste método, onde é possível verificar-se os blocos de 4x4 pixéis e as 8 orientações do gradiente mais frequentes.

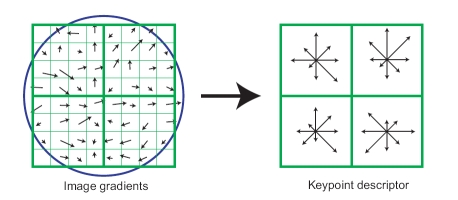


Figure 4 - Cálculo dos descritores de um keypoint através do histograma da orientação

Uma vez determinados os vários *keypoints* de uma imagem, é necessário encontrar-se correspondências entre *keypoints* entre imagens para o cálculo das matrizes e . Ao comparar-se a diferença(erro) entre os descritores de um *keypoint* numa imagem e todos os descritores de *keypoints* da nova imagem encontra-se os dois descritores com menor erro. Caso as duas distâncias sejam semelhantes, não existe correspondência evidente entre os *keypoints.* Caso contrário, se as distâncias forem muito diferentes considera-se a existência de um *match* entre *keypoints.* Estes matches são utilizados posteriormente nos algoritmos *RANSAC* e *procrustes*.

**RANdom SAmple Consensus (RANSAC)**

O algoritmo *RANSAC* consiste num método iterativo que estima os parâmetros de um modelo matemático através de um conjunto de dados que inclui pontos válidos e pontos que não satisfazem o modelo - *outliers.* A sequência de passos deste método mostra-se de seguida:

1. Definir o modelo que melhor classifica a distribuição dos dados.
2. Escolher um conjunto de dados cujo número é suficiente para se determinar os parâmetros do modelo. Por exemplo, caso o modelo seja uma reta são necessários apenas dois pontos para se calcular e .
3. Determinar os parâmetros através do conjunto de dados selecionado.
4. Verificar o número de dados que satisfazem o modelo com uma certa tolerância, isto é, cujo erro (diferença entre o valor de e o valor real) é inferior a um *treshold* arbitrário. Isto equivale a definir-se uma função de custo.

Estes dados são considerados válidos, isto é, *inliers.* Os dados que se afastam significativamente do modelo consideram-se *outliers.* Finalmente, o modelo define-se como um modelo válido caso o número de *inliers* seja muito superior ao número de *outliers*.

1. Repetir os passos 2 a 4 com outro conjunto de dados, pertencentes ao *dataset* inicial e escolher os dados que levam a um maior número de *inliers.* Estima-se novamente os parâmetros apenas com os *inliers* selecionados.

No caso particular desta aplicação do algoritmo *RANSAC* o modelo é definido pelos parâmetros que são os elementos de e e é da forma

sendo e *keypoints* pertencentes à imagem da câmara 1 e da câmara 2, respetivamente. São necessários pelo menos 4 pares para determinar os parâmetros.

A função de custo é

onde corresponde ao índice do par de pontos identificados entre câmaras. A solução para a matriz de rotação e para o vetor de translação é encontrada através da resolução de um problema de *Procrustes*. Este método determina a matriz ortogonal que minimiza , dadas as matrizes e .

1. **Implementação**
2. Obtenção de *background* e *foreground*

Ao longo de uma sequência de imagens existem regiões cujos pixéis têm uma profundidade (dada pela imagem de profundidade) constante o que implica que correspondem ao *background*. Todas as regiões que não pertençam ao *background* são consideradas objetos. Para se identificar os pixéis pertencentes ao *background* determina-se a mediana de todas as imagens de profundidade pertencentes à sequência uma vez que os pixéis que pertencem ao *background* são os mais frequentes. Este processamento é executado para a sequência de imagens de cada câmara pelo que se obtém duas matrizes de *background*.

O *foreground* são as regiões que delimitam objetos relevantes em cada imagem desta sequência. Para cada imagem da sequência de uma câmara calcula-se a diferença absoluta entre a imagem de profundidade e a matriz *background* anteriormente calculada. De seguida seleciona-se apenas os pixéis superiores à média desta diferença absoluta (não se conta pontos mortos da *Kinect*) para ter um *foreground* mais exigente.

Para distinguir objetos parcialmente oclusos (i.e. uma pessoa atrás de outra) calcula-se a magnitude do gradiente da imagem de profundidade o que deteta os contornos dos objetos. Usa-se estes contornos para separar objetos nas condições referidas.

1. Obtenção dos objetos através das *labels*
2. Consenso entre objetos identificados no sistema de coordenadas externo
3. Identificação de movimento de objetos – *tracking*
4. **Discussão de Resultados**

Dataset lab1: não detetamos os alunos porque 1) a variação de volume é reduzida sendo que pode ser interpretada como ruído; 2) são considerados background; obtêm-se dois objetos diferentes porque o 1o é identificado como a cabeça do professor e o 2o objeto é já o corpo completo do professor; corre em média em 4.6 segundos.