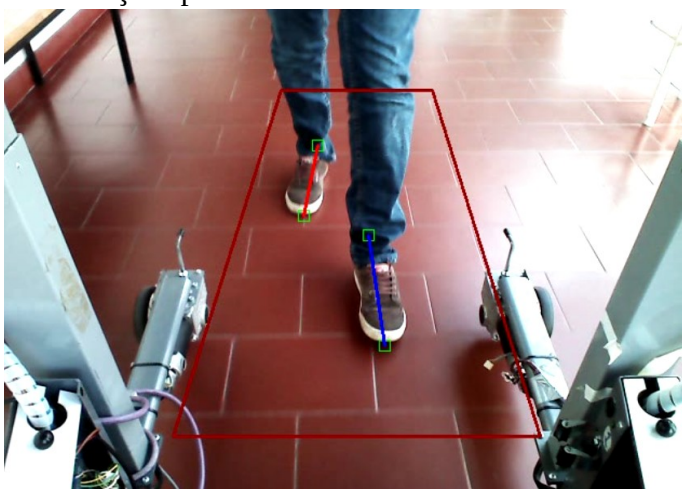
Aplicação de métodos clássicos de processamento de imagem na detecção e localização dos pés

A partir de uma sequência de imagens/video capturadas com uma câmera RGB-D (cor + profundidade - vídeo e imagens anexados), é pretendido o desenvolvimento de um algoritmo para a detecção dos pés, particularmente as articulações do tornozelo e ponta do pé, utilizando para isso métodos e ferramentas de processamento de imagem clássico. A deteção em tempo real destes pontos de interesse é de particular relevância na avaliação da marcha de um determinado sujeito, visto permitir efetuar uma análise de marcha detalhada através de uma multitude de métricas espacio-temporais, tendo especial valor no seguimento da reabilitação de pacientes com limitações ao nível da marcha (p.e. casos clínicos de ataxia pós-enfarte).



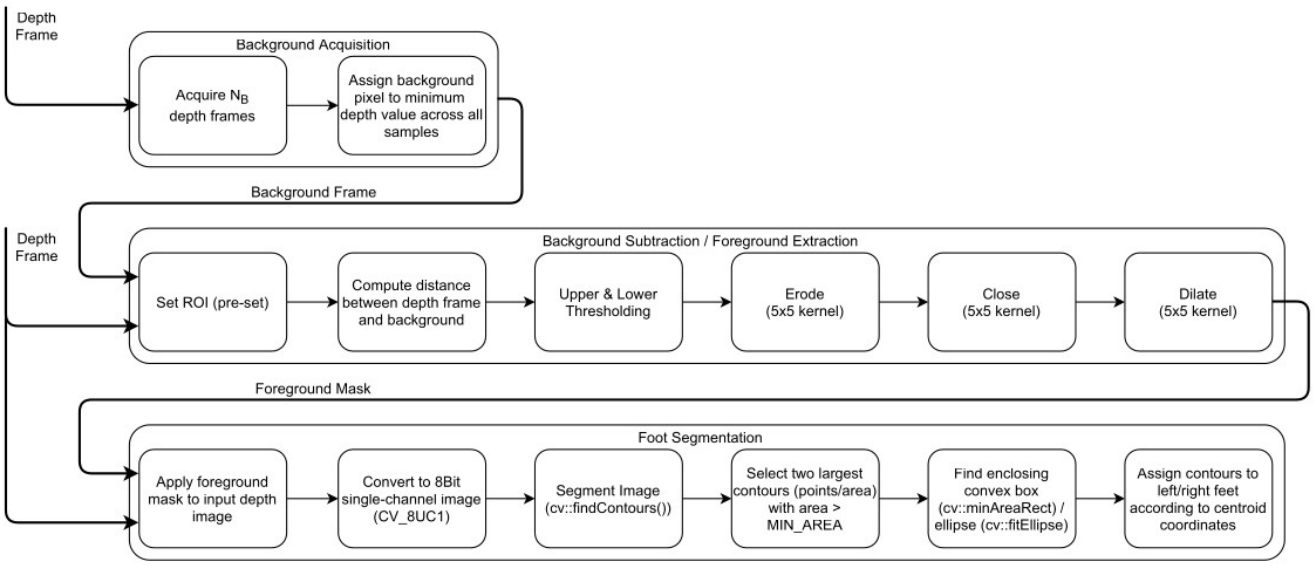
O trabalho incidirá sobre dados adquiridos com sujeitos saudáveis na utilização de um andarilho inteligente equipado com duas câmaras RGBD convencionais (tipo Kinect), que incluem **imagens de cor** (vídeo RGB) e de **profundidade**, pelo que várias abordagens podem ser adotadas de modo a obter os resultados pretendidos.

O resultado pretendido são as posições 2D (linha, coluna) dos píxeis associados com o **tornozelo** e **ponta do pé** esquerdo e direito (Figura 2). Serão fornecidos valores de referência “ground truth” para que os alunos possam validar/quantificar a distância entre as posições encontradas e o valor real. Opcionalmente, a avaliação pode também ser feita em 3D através da reconstrução das



coordenadas mundo com os parâmetros intrínsecos das câmeras, de modo a inferir se a soluçáo se aproxima do valor referência (<= 3cm "state of the art").

Exemplo de solução:



É também fornecida uma solução funcional que se espera servir como exemplo/modelo a seguir no delineamento/desenvolvimento de uma nova solução que endereçe as limitações atuais. O código foi desenvolvido em C++ para sistemas Unix com recurso às bilbiotecas OpenCV que fornecem ferramentas de processamento de imagem.

Todo o processamento de imagem se encontra assente na classe *FeetDetector*, que implementa através de funções estáticas os vários blocos apresentados na Figura 3. De modo a correr a deteção dos pés deve ser utilizado o método *FeetDetector::detect()* que retorna um objecto do tipo *std::vector< cv::Point2i >* com os pontos associados aos pontos de interese que se pretende detetar.

A parametrização do algoritmo é feita através de uma estrutura do tipo *cv::FeetDetectorParameters*, que permite alterar os vários parâmetros que influenciam o funcionamento/processo de deteção (e.g. dimensão de kerneis morfológicos, dimensões referência do sujeito, thresholds para profundidade e área de contorno, etc.).

Conceptualmente, o algoritmo utilizado encontra-se ilustrado na Figura 3; Inicialmente é feita a remoção do chão através da comparação com uma referência pré-adquirida (*floor_reference.png*), o que permite isolar os pixeis relativos aos pés do sujeito. Seguidamente são aplicadas várias operações morfológicas (*cv::morphologyEx*) que suavizam as formas dos pés, e são detectados os contornos através da função cv::findContours() providenciada pelo OpenCV. Cada um dos contornos obtidos é aproximado a um rectangulo rodado de acordo com a orientação do pé (*cv::minAreaRect*), o que facilita a obtenção dos extremos superiores e inferiores para cada pé.

Paralelamente, é também fornecido código que carrega e sincroniza as imagens de cor e profundidade, e que se espera que seja utilizado caso os alunos decidam desenvolver a solução em C++, ficando a cargo dos mesmos a implementação de mecanismos de sincronização (caso sejam necessários) noutras linguagens de programação. De modo a obter imagens sincronizadas, basta instanciar um objecto da classe *TrialData* fornecendo o caminho para onde os dados se encontram guardados e chamar o método *TrialData::next()* que retorna um *std::vector< cv::Mat >* com as imagens cada sensor.