Universidade de Lisboa

Instituto Superior Técnico

Processamento de Imagem e Visão

Opção de Projecto 2

Identificação de Pessoas em Movimento com uma Câmara Kinect em Posição Fixa

Francisco Oliveira	MEEC	75167
Inês Lourenço	MEEC	75637
Nuno Lages	MEIC	82162

30/12/2015

Introdução e Descrição do Problema

A identificação de objectos e, particularmente, de pessoas em movimento em imagens, é um problema cuja solução tem múltiplas aplicações. Ao pretender seguir-se a trajetória da pessoa, as maiores dificuldades relacionam-se com a variação da dimensão das pessoas nas imagens com a profundidade (distância da câmara) a que estas se encontrem, sendo particularmente problemático reconhecê-las a distâncias superiores a cinco metros. Outra dificuldades passam pela distinção de duas pessoas que estejam muito próximas entre si.

No presente documento apresentamos um estudo que aborda este problema e as referidas dificuldades. Aplicamos a nossa solução a quatro *datasets* disponibilizados no âmbito da cadeira de Processamento de Imagem e Visão, do Instituto Superior Técnico. Mais especificamente, o caso abordado é aquele em que se tem uma câmara *kinect* e uma câmara RGB fixas numa sala onde existe apenas uma entrada/saída e há pessoas a entrar, sair e movimentar-se livremente na sala.

Solução

Neste capítulo apresentamos a solução para o problema de seguir objectos com *kinect* descrito no capítulo introdutório. Cada *dataset* é uma sequência de matrizes com informação de profundidade e RGB (não utilizadas) de uma cena por onde circulam pessoas. A solução foi implementada num programa para o ambiente Matlab. Cada subsecção trata de um dos problemas com que se lidou; são descritas as fases de resolução do problema bem como as alternativas preteridas.

Subtração de Background

De modo a identificar as pessoas em movimento (*foreground*), é conveniente, em primeiro lugar, identificar todas as áreas das imagens que são o ambiente, estático, da cena, ou seja, o *background*, de modo a descartá-las e facilitar a identificação e manipulação dos pixels correspondentes a objetos de interesse.

Um dos métodos mais utilizados [1] para encontrar o *background* numa sequência de imagens consiste em calcular a mediana de cada pixel, ou seja, recolher os valores para cada pixel em todos os instantes e calcular a sua

mediana. A utilização da mediana tem a vantagem, relativamente a outras medidas de tendência central, nomeadamente a média, de ser robusta a variações significativas nos dados. No caso de imagens, esta propriedade da mediana impede que os valores identificados como *background* sejam afetados pelo movimento dos objetos no *foreground*. Mais concretamente, se tivermos uma pessoa a deslocar-se na cena, ao calcular a média de todos os pixels em todas as imagens, estes virão, provavelmente, significativamente afetados pelos valores correspondentes à pessoa (*foreground*). Por outro lado, utilizando a mediana em detrimento da média, o resultado será afectado pela pessoa apenas se esta permanecer aproximadamente na mesma posição mais de metade do tempo considerado.

O cálculo da mediana implica o carregamento, idealmente, de todas as imagens da sequência para a memória do computador. Isto pode ser um problema em computadores pessoais com limitações de memória. Na nossa implementação, resolvemos este problema carregando um subconjunto de imagens igualmente espaçadas no tempo representativas da sequência do princípio ao fim. Deste modo, impede-se que o processo exceda a memória que lhe é permitida.

Em alguns dos *datasets* testados muitas das imagens tem muitas pessoas (*foreground*), e a utilização da mediana simples como descrita acima não é suficiente, obtendo-se um *background* influenciado pela presença das pessoas. De modo a corrigir este aspecto, calculou-se a mediana do *background* calculado como descrito anteriormente com a última imagem da sequênca. Este método permitiu obter um *background* mais fiável uma vez que em todos os *datasets* testados a última imagem corresponde apenas ao *background* real, não havendo nenhum objeto que se mova noutras imagens, e evita problemas de falta de memória que poderiam decorrer da utilização de mais imagens no cálculo da mediana. No entanto, caso os recursos computacionais o permitam, é desejável usar todas (e apenas) as imagens da sequência na determinação da mediana.

Reorientação do Referencial da Cena

A reorientação do referencial de modo a fazer coincidir o chão da sala com o plano *xy* é de grande utilidade para os processamentos posteriores. O chão é uma superfície plana, pelo que pode ser modelado como um plano na sequência de imagens de interesse e usado como referência para uma transformação que o sobrepõe ao plano *xy*. É, aliás, a maior superfície plana nas imagens, pelo que é identificável por aplicação directa do algoritmo Random Sampling Consensus (RANSAC) [1].

A nossa implementação do algoritmo RANSAC aplicado à identificação dos pontos no chão funciona da seguinte maneira:

1. Seleccionamos aleatoriamente 3 pontos do *background* estimado através da mediana.
2. Calculamos os parâmetros a, b, c e d da equação do plano ax + by + cz = d recorrendo a decomposição em valores singulares (SVD).
3. Determina quantos pontos estão a menos de uma certa distância ("erro", utilizámos 5 cm) do plano calculado.
4. Repete 499 vezes e vai guardando os parâmetros do plano para o qual mais pontos estão dentro da margem de erro. Os índices dos pontos dentro da margem de erro para o melhor plano são também guardados para serem utilizados na transformação (translação e rotação) da cena.

Por uma questão de eficiência, utilizamos apenas metade dos pontos disponíveis na imagem de *background*. Esta opção não implicou prejuízos nos resultados a jusante.

Para a determinação dos parâmetros de cada plano testado pelo RANSAC, recorreu-se a SVD, uma técnica aplicável a matrizes reais ou complexas. No nosso caso, representamos as coordenadas dos 3 pontos escolhidos aleatoriamente para definir um plano numa matriz tridimensional real. Denominemos esta matriz M. A SVD decompõe a matriz M no produto de três matrizes:

$$M = U \Sigma V^*$$

Nesta equação, as matrizes U e V^* são matrizes de rotação e Σ é uma matriz de escala. A matriz V^* contém, na quarta coluna, os parâmetros a, b, c e d que definem o plano de equação ax + by + cz = d.

Deteção do Espaço da Cena e Voxelização

A detecção dos limites da cena foi também um passo de pré-processamento necessário para que se pudesse utilizar toda a informação sobre a cena nas imagens. De modo a identificar os seus limites, no mesmo ciclo onde são carregadas as imagens utilizadas para identificar o *background* através da mediana, são identificados os maiores e os menores valores registados ao longo dos eixos x e y para esse conjunto de imagens. Estes valores são então utilizados como referência para construir uma matriz 800x800 cujas células são *bins*, voxels

contendo o número de pontos que lhes correspondem verticalmente. É possível que existam alguns pontos fora dos limites determinados no conjunto global de imagens. Se existirem, estes são descartados; isto não tem influência na detecção das pessoas, uma vez que a informação perdida é muito pequena.

Rotação da Cena

Nesta fase pretende-se reorientar a cena de modo a que fique no primeiro octante do espaço e que o chão coincida com o plano xy.

Para este fim, começa-se por se executar uma translação que centra o referencial no centro chão - este ponto é determinado como sendo o ponto médio dos pontos encontrados aquando da determinação do chão (pontos a menos de 5 cm do plano do chão). Uma vez centrado o referencial no ponto médio do chão, procede-se à rotação de modo que o plano do chão fique sobreposto ao plano xy. Por fim, procede-se a uma translação que move todos os pontos da cena para o primeiro octante do referencial.

Detecção de Objectos em Movimento

Após o reposicionamento da cena conforme descrito nas secções anteriores, o programa percorre todas as imagens da sequência e procura identificar e seguir as pessoas em movimento. (Quando há pessoas que não se movimentam, estas não são identificadas, são indistintas do *background*.)

Detecção de Objectos do *Foreground*

Nesta subsecção descreve-se o processo de identificação de pessoas no *foregound* para cada núvem de pontos.

O primeiro passo do processo consiste em aplicar à núvem de pontos a transformação descrita na secção Rotação da Cena. Após esta transformação, observa-se que, em geral, há muitos pontos que não correspondem aos objectos de interesse relativamente próximos do chão. Por esta razão, fazemos uma selecção dos pontos entre 1 e 2.1 metros de altura, onde a maioria dos pontos correspondem a pessoas. São estes pontos que são voxelizados, novamente, numa matriz 800x800.

Após a voxelização, por uma questão de eficiência, testa-se se a área total na matriz (soma de elementos diferentes de 0) é inferior a um certo número. Isto

significa que na imagem em causa não há pessoas e a análise pode passar imediatamente à imagem seguinte.

Se a imagem contiver uma área total superior ao *threshold* definido, são identificados todos os componentes ligados na mesma. Para cada componente é contada a soma dos pontos nos respectivos voxels. Aqueles objectos que tenham um número total de pontos superior a um determinado *threshold* é considerado uma pessoa. A posição da pessoa é considerada como sendo o centróide do objecto, calculado com a função Matlab *regionprops*.

Considerou-se utilizar a área das projecções das pessoas no plano do chão como critério de identificação das mesmas, em vez das respectivas somas de pontos. Ou seja, objectos com áreas superiores a certo *threshold* seriam classificados como pessoas. No entanto, quando comparados os resultados entre as duas abordagens, concluiu-se que a soma de pontos é um critério que identifica correctamente, e apenas, as pessoas em mais imagens.

Correspondência entre Objectos em Imagens Sucessivas

A correspondência entre a representação de pessoas numa determinada imagem e na imagem seguinte faz-se com base na distância entre os respectivos centróides na primeira imagem e os centróides na segunda imagem: um centróide na primeira imagem é associado a um centróide na segunda imagem por minimização da soma das distâncias entre os centróides associados. Para este efeito, é utilizado o algoritmo Húngaro, o qual resolve este problema de correspondência em tempo polinomial $O(n^3)$.

Casos Especiais

Os principais desafios são as situações nas quais as pessoas entram ou saem da cena, se aproximam e as respectivas núvens de pontos ficam indistintas, ou quando uma pessoa fica oculta atrás de outra. Neste último caso, o número de pontos pode não ser suficiente para a identificar como pessoa ou os pontos correspondentes à pessoa podem ficar divididos em duas regiões distintas. Os desafios que o programa deve ser capaz de resolver, são, portanto:

1. Distinguir quando uma pessoa sai de cena e quando está de tal modo próxima de outra que as suas núvens de pontos se fundem.
2. Reconhecer quando a núvem de pontos de uma mesma pessoa se divide em duas.
3. Identificar correctamente cada pessoa quando estas se aproximam e depois se afastam.

1) Entrada e Saída de Pessoas da Cena

O programa assume que o aparecimento de mais uma região para a qual é calculado um centróide corresponde à entrada de mais uma pessoa na cena, com excepção dos casos previstos na subsecção Divisão de Pessoas em Duas Regiões. Por outro lado, assume-se que o desaparecimento de uma região e, consequentemente, de um centróide, corresponde à saída de uma pessoa de cena, com excepção dos casos previstos na subsecção Objectos Sobrepostos.

A solução encontrada para distinguir uma saída de uma aproximação entre duas pessoas (com fusão das núvens de pontos) foi, para as situações em que houve menos uma pessoa identificada que na imagem anterior, verificar se o número de pontos aumentou significativamente relativamente ao objeto que lhe corresponde na imagem anterior. Na prática, funciona bem testar se o objecto tem um número de pontos pelo menos 1.8x maior. Se for o caso, considera-se que a nova núvem de pontos corresponde às duas pessoas que lhe estão mais próximas na imagem anterior. Esta abordagem funciona para os *datasets* estudados uma vez que não temos situações em que, simultaneamente, duas pessoas estejam muito próximas e mais uma pessoa entre na cena.

2) Divisão de Pessoas em Duas Regiões

Por vezes acontece que para a núvem de pontos correspondente a uma só pessoa são produzidas duas regiões separadas na matriz de voxelização. Este problema (desafio 2) é resolvido por pós-processamento: depois de processadas todas as imagens da sequência, o *array* de regiões encontradas é percorrido e é analisado o deslocamento de cada objeto relativamente à imagem anterior. A eliminação de objetos com deslocamento significativamente inferior ao habitual permite resolver este problema. Por exemplo, no *dataset filinha*, as pessoas deslocam-se, em média, entre imagens, cerca de 30 cm, pelo que percorrem vários metros enquanto permanecem na cena. Descartar os objectos que, no total, não chegam a percorrer 50 cm é suficiente para resolver o desafio 2 neste *dataset*.

3) Objetos Sobrepostos

Há imagens onde duas pessoas distintas se encontram tão próximas que são identificadas como um só objeto pelo nosso programa. Este foi o problema mais difícil de resolver do projecto.

Quando este problema acontece, temos uma região pelo menos 1.8x maior que a que lhe é associada pelo algoritmo Húngaro. Temos também menos uma pessoa que as identificadas na imagem anterior. Além disso, na imagem seguinte, ou eventualmente algumas imagens à frente, voltaremos a ter as duas pessoas correctamente identificadas pelo programa. Quando as duas pessoas voltam a ser correctamente identificadas, de modo a representar a posição das pessoas nas imagens anteriores em que estavam juntas (com um só centróide), calculam-se os pontos médios entre os seus centróides na primeira imagem em que voltam a estar separadas e o seu centróide conjunto. Se houver várias imagens em que as pessoas estiveram juntas, os novos centróides são utilizados, sucessivamente, para calcular os centróides das imagens anteriores até que as pessoas tenham centróides individuais em todas as imagens. A correspondência entre centróides de imagem para imagem pode agora ser determinada como habitualmente.

Foram pensadas várias alternativas para resolver este problema, a seguir expostas, mas foram preteridas a favor da apresentada no parágrafo anterior.

1. Estimar dois centróides com base na área correspondente às duas pessoas juntas
    a. Ideia neste caso seria encontrar os dois pontos com maior distância entre si na região correspondente às pessoas juntas, calcular a recta que passa por eles, dividir a região em duas pela recta perpendicular a esta e calcular centróides para as duas regiões obtidas. A cada pessoa seria atribuído o centróide mais próximo do seu centróide anterior. Este método funcionaria particularmente mal quando uma pessoa ficasse oculta atrás de outra, uma vez que neste caso a divisão da região ao longo do eixo de maior comprimento não seria adequada. Por outro lado, as trajectórias das pessoas podem cruzar-se e neste caso a atribuição dos centródes seguintes serão erradas.
2. Estimar as alturas das pessoas com base na informação tridimensional e utilizar as alturas para distinguir as pessoas
    a. Esta solução, além de mais difícil de implementar, pode falhar devido aos movimentos das pessoas se estas tiverem alturas parecidas.

Resultados Experimentais

Neste capítulo são escritos os resultados obtidos com cada *dataset*, sendo apontadas as falhas da abordagem escolhida e discutido como estas podem ser resolvidas no futuro.

O *output* do programa consiste num cell array onde cada célula contém um array que contém os centróides de um objeto (pessoa) para as diferentes imagens onde foi identificado. As coordenadas do centróide apresentadas estão convertidas para o sistema de eixos original (pré-voxelização).

Para todos os datasets, o programa desenvolvido é capaz de seguir pessoas sem erros quando estas estão isoladas, com excepção da situação no *dataset confusão* na qual uma das pessoas se afasta muito do *kinect*. Nesta situação, o número de pontos que representa a pessoa torna-se muito reduzido e não chega ao *threshold* definido para que um objecto seja classificado como pessoa.

Para o *dataset um*, nas imagens 11 e 24 há um objeto identificado além da pessoa em movimento, no entanto este é eliminado no pós-processamento, uma vez que não se move como se espera de uma pessoa. Portanto o resultado final do programa para este dataset é correcto.

Com o *dataset filinha*, todas as pessoas são correctamente identificadas. Em dois casos começam a ser seguidas apenas na segunda imagem em que aparecem, o que é ainda assim muito positivo, considerando que é apenas uma imagem de diferença e que se tem um total de 6 pessoas. Há mais objetos (que não pessoas) que são identificados: uma cabeça nas imagens 5 e 6 e alguns artefactos noutras imagens mas estes são eliminados no pós-processamento, obtendo-se o resultado final desejado. Há que ter em conta que o método referido na subsecção Objetos Sobrepostos apenas foi implementado para o agrupamento de dois objetos de uma vez, pois era o necessário para o sucesso do algoritmo neste projeto, notando-se neste *dataset* o seu correto funcionamento.

Relativamente ao *dataset confusão*, nos nossos testes ocorreu o desaparecimento de uma pessoa ao fundo da cena simultaneamente com a entrada de outra. Isto conduziu à não identificação destas situação e à associação errada de centróides entre essa imagem e a anterior. Isto torna difícil uma análise mais detalhada do resultado. O facto de o programa perder de vista pessoas quando estas se afastam na sala é um aspecto que deveria ser melhorado. Quando a pessoa volta a aproximar-se, é identificada como uma pessoa nova na cena. Isto deveria ser corrigido utilizando um critério como a proximidade entre os lugares de aparecimento e desaparecimento.

Conclusão

O estudo e os testes desenvolvidos durante a elaboração deste programa mostram que o *tracking* de pessoas com um *kinect* fixo é um problema relativamente simples quando se trata de pessoas isoladas, mas pode ser bastante complexo quando as pessoas se aproximam. Neste caso, têm de ser

assumidas algumas condições para que o programa seja capaz de reconhecer as pessoas correctamente quando estas se afastam, e estas condições nem sempre se aplicam.

O programa desenvolvido funciona muito bem para os *datasets um* e *filinha*. Em particular, o pós-processamento com base no movimento é bastante poderoso a distinguir pessoas de objectos detectados extemporaneamente mas que não se movem. Por outro lado, o programa tem falhas para os *datasets 2* e *confusão*. Nomeadamente, não é capaz de reconhecer que se trata da mesma pessoa quando esta se afasta muito da câmara e depois regressa; este problema pode ainda estar na base de associações inválidas entre pessoas na imagem actual e na imagem anterior. Existem técnicas poderosas baseadas na análise da cor (nomeadamente do tom) das pessoas e das respectivas roupas que poderiam ter sido usadas para melhorar a correspondência entre as pessoas em imagens consecutivas. No entanto estas não foram implementadas no presente projeto. Tais técnicas poderiam, por exemplo, comparar o tom médio de objectos em imagens sucessivas e aplicar o algoritmo Húngaro para fazer a associação das pessoas. Possivelmente, a comparação dos próprios histogramas de cores em vez da média talvez tivesse resultados melhores.

References

[1] Richard Szeliski (2010) *Computer Vision: Algorithms and Applications*, Springer