Visão Estéreo

Bruno Fernandes Carvalho - 15/0007159
Dep. Ciência da Computação - Universidade de Brasília (UnB)
Princípios de Visão Computacional - Turma A
Data de realização: 06/05/2017
brunofcarvalho1996@gmail.com

Abstract

Knowing objects distances from the camera is essential in many applications, as for example a robot's navigation system. One of many ways to calculate a scene's depth map is through stereo vision, which relates images taken in the same time by different cameras to obtain the scene distance. This project shows how it can be calculated, as well as the factors that influences and causes error in the depth map formation.

Abstract

Obter a distância de objetos à câmera é essencial para diversas aplicações, como por exemplo um sistema de navegação de um robô. Um dos meios de calcular o mapa de profundidade da cena é por meio da visão estéreo, que consiste em relacionar imagens tiradas ao mesmo tempo por câmeras diferentes e obter as distâncias desejadas. Nesse projeto é visto como isso pode ser calculado, assim como os fatores que influenciam e os que causam erros na formação do mapa de profundidade.

1. Objetivos

Essa atividade tem como base entender e explorar uma técnica de modelagem 3D (visão estéreo), assim como desenvolver um algoritmo que extraia o mapa de profundidade a partir de pares de imagens estéreas.

2. Introdução

Umas das técnicas utilizadas para determinar a distância de objetos numa cena e, consequentemente, criar um mapa de profundidade da mesma é chamada de visão estéreo ou múltiplas vistas. Esta é baseada na seguinte ideia: duas ou mais fotos são tiradas ao mesmo tempo por câmeras diferentes (uma foto por câmera) que normalmente estão alinhadas. Dessa forma, várias fotos de pontos de vista diferentes retratam a mesma cena, permitindo que seja obtida relações entre as fotos tiradas para calcular a profundidade. O esquemático a seguir exemplifica como isso é feito para duas câmeras.

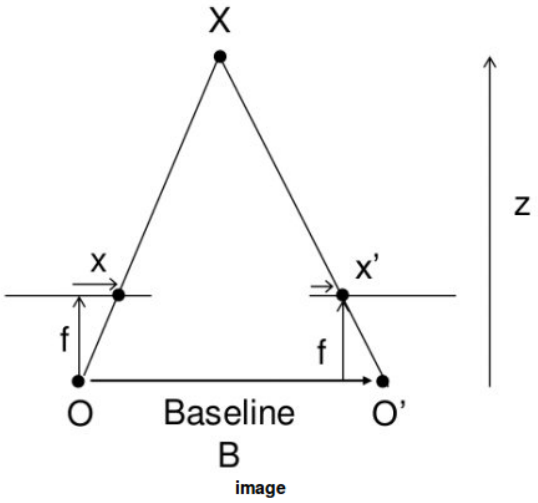

Figura 1. Esquemático da técnica de múltiplas vistas

A partir de semelhanças de triângulos e relações geométricas, acha-se facilmente a distância Z apresentada na figura 1 se for conhecido o mesmo ponto da cena nas duas imagens. Assim, o principal desafio dessa técnica é achar a localização de um ponto em uma imagem na outra. Isso é difícil, pois os mesmos pontos da cena estão deslocados na outra imagem, ou seja, não têm a mesma coordenada (x,y). Além disso, isso permite que alguns pontos sejam omitidos na outra imagem, já que ambas não cobrem igualmente a cena.

Para minimizar os problemas citados acima, são usadas câmeras perfeitamente alinhadas e com uma distância de baseline não muito grande. Essas características, junto com um processo de retificação das imagens, garantem que o ponto de uma imagem estará na mesma linha da outra,

porém em uma coluna diferente. Isso simplifica muito o problema, já que não é necessário procurar um pixel na outra imagem toda. Além disso, para garantir que será achado o mesmo ponto nas duas imagens, é criado uma máscara de tamanho variável que conterá os vizinhos do ponto de interesse. Percorrendo com essa máscara a linha toda, é possível encontrar a posição do pixel que mais se aproxima do ponto procurado.

Tendo a posição do mesmo ponto nas duas imagens usando a técnica descrita anteriormente e usando relações matemáticas, conclui-se que a distância de um ponto às câmeras é:

$$Z = \frac{Baseline * f}{x_{im} - x'_{im}} \tag{1}$$

3. Materiais e Metodologia

A partir de pares de imagens capturadas por uma câmera estéreo calibrada e retificada disponíveis no Moodle, foi criado um mapa de profundidade a partir da técnica de múltiplas vistas já mencionada na seção anterior. Para isso, foram utilizadas medidas hipotéticas de baseline e foco da câmera (B = 120mm e f = 25mm).

Antes de realizar o processamento da profundidade, pergunta-se ao usuário o tamanho da máscara W, que influencia diretamente nos resultados encontrados. Assim, durante a mesma execução do programa, é possível processar duas vezes com tamanhos da janela diferentes e comparar os resultados, já que o mapa de profundidade é salvo no mesmo diretório do executável.

A metodologia para encontrar os pixels de uma imagem na outra já foram descritos na Introdução, isto é, para cada pixel na imagem esquerda, a máscara deve percorrer na mesma linha do pixel a imagem direita procurando pela posição que mais se pareça ela. Para saber qual é essa posição, é usada a soma das diferenças absolutas entre os valores dos pixels nas máscaras das duas imagens, e a que tiver menos custo é a posição procurada. Nos pixels das bordas, algumas partes da máscara saem da imagem, sendo necessário atribuir o valor do pixel central às posições da janela que estão fora. Também pode ser atribuído o valor zero em vez do valor central, porém a opção usada representa melhor a correspondência entre as imagens.

Além disso, para realizar o procedimento mencionado para as imagens de entrada do programa, é feito um redimensionamento destas para o tamanho 320x320, tendo como objetivo reduzir o tempo necessário para realizar os cálculos. Ainda assim, demora-se um tempo razoável para realizar a correspondência de todos os pontos, variando com o tamanho da janela W.

Vale deixar claro que, para o desenvolvimento dos algoritmos, foi utilizado a linguagem de programação C++, assim como técnicas de orientação a objetos. Na hora de execução do código no terminal, é essencial que seja lido o arquivo README.txt para que os argumentos de entrada sejam certos, garantindo que o algoritmo seja executado corretamente.

4. Resultados

Usando os pares de imagens disponibilizados no Moodle, realizou-se os procedimentos citados para encontrar os pontos correspondentes nas duas imagens e o respectivo mapa de profundidade. Assim, foi possível comparar esse mapa para valores de W diferentes (tamanho da janela), conforme visto nas figuras 2 e 3.

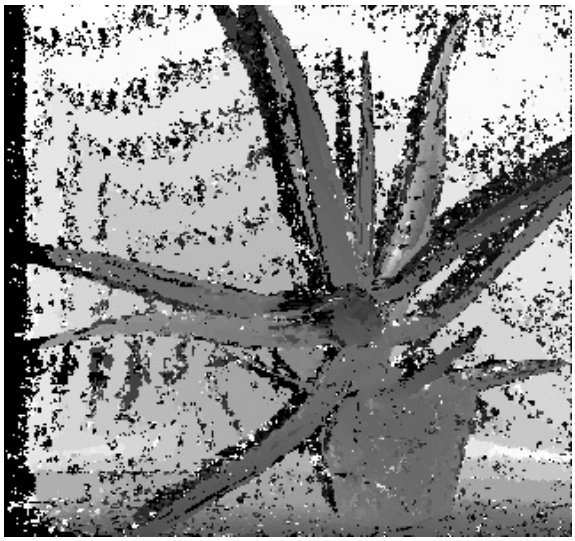

Figura 2. Mapa de profundidade para W igual a 3

Figura 3. Mapa de profundidade para W igual a 15

Como pode ser observado, quando a janela de modelo é pequena (W = 3), mais detalhes da cena são representados no mapa, porém aparece mais ruídos devido à variações aleatórias de luminosidade e erros na captação das imagens. O ruído aumenta pois menos vizinhos do pixel central são comparados nas duas imagens, significando que qualquer alteração mínima pode causar erro na correspondência dos pontos. Já na janela grande (W = 15) isso é atenuado, pois mais pixels definem um ponto na imagem, visto que a janela é maior. Em compensação a esse benefício, menos detalhes da cena são capturados e apenas regiões mais borradas são mapeadas.

Outra observação que pode ser feita é a quantidade de pixels pretos nas primeiras colunas das imagens acima. Isso ocorre de forma proposital, já que estes representam os pontos que não tiveram correspondência nas duas imagens, ou seja, os pontos da imagem esquerda que não estão contidas na da direita.

Apesar das diferenças, ambos mapas de profundidade representam de forma razoável a cena, atribuindo de forma correta distâncias parecidas ao mesmo objeto. Isso pode ser concluído a partir da forma dos objetos no mapa feito. Apesar dos bons resultados, ainda é visto bastante ruído, que também pode estar relacionado à variação de luminosidade devido à posições diferentes das câmeras, gerando, as vezes, variações significativas nos valores dos pixels.

5. Conclusão

Ficou claro como funciona a técnica de modelagem 3D visão estéreo, que utiliza múltiplas vistas para obter relações de profundidade entre objetos da cena. Além disso, foi implementado um algoritmo que reproduz um mapa com essa relações, o qual teve resultados satisfatórios. Ainda, entendeu-se como o valor da janela W altera os resultados do mapa e quais são os possíveis causadores de ruídos entre as duas imagens capturas, gerando erros no mapa.

Referências

[1] G. Bradski and A. Kaehler, *Learning OpenCV*, 1st ed. O'Reilly.

[2] Forsyth and Ponce, *Computer Vision - A Modern Approach*, 1st ed. Pearson.

[3] OpenCV API Reference. (2017, 31 Março). [Online]. Available: http://docs.opencv.org/2.4/modules/refman.html

[1] [2] [3]