

Project 3 Questions

Questions

Alexandre Yuji Kajihara
Noemi Scherer

Q1: Imagine we were tasked with designing a feature point which could match all of the following three pairs of images. Which real world phenomena and camera effects might cause us problems? Use the MATLAB function *corner* to investigate. *corner(I, 1000)*.

RISHLibrary: [One Two](#) — *Chase*: [One Two](#) — *LaddObservatory*: [One Two](#)

A1: Os fenômenos do mundo real e efeitos de câmera que podem nos causar problemas na hora de projetar um ponto de *feature* seriam: a iluminação, já que dependendo de como a luz está refletindo na foto pode fazer com que um pixel se torne mais escuro, do que ele é originalmente. Além disso, se estiver muito escuro a imagem pode apresentar alguns ruídos, que podem comprometer a correspondências entre as *features* das imagens. Um outro fator que pode afetar é a lente da câmera que foi tirada a foto da imagem, já que dependendo da lente a mesma pode provocar distorções, um exemplo de lente são as *fish eye*, que distorce a imagem nas extremidades. Um outro fator é o ângulo, que pode provocar com que um ponto apareça em um imagem, mas não apareça em outra.

Q2: In designing our feature point, what characteristics might we wish it to have? How should we design for the fundamental tradeoff between feature point invariance and discriminative power?

A2: Existem dois tipos de características que desejamos obter. A primeira é a *feature*, rastreada com precisão utilizando uma técnica de busca local. Essa característica é mais adequada quando as imagens são tiradas de pontos de vista próximo ou em rápida sucessão. A segunda é detectar de maneira independente os recursos em todas as imagens em consideração, em seguida, combinar os recursos com base em sua aparência local. A segunda característica é mais adequada quando é esperada uma grande quantidade de movimento ou mudança de aparência, por exemplo, unir panoramas e reconhecimento de objetos. Para projetar o compromisso fundamental, entre a invariância de ponto de recurso e o poder discriminativo de uma *feature* só pode ser revelado quando é acompanhado de algumas por outros atributos. Além disso, existem algumas maneiras de avaliar o poder discriminativo de cada *feature*, por exemplo a razão de variância aumentada (AVR) em que é a razão entre a variação entre as classes de *feature* e variação dentro da classe da *feature*. Além disso, outras medidas para o poder discriminativo de uma *feature* seria o ganho de informação e informação mútua.

Q3: In the Harris corner detector, what do the eigenvalues of the 'M' second moment matrix represent?

A3: Os autovalores da matriz de segundo momento 'M' é conhecida como tensor de estrutura de um pixel. Esse arranjo bidimensional representa a classificação dos pontos da imagem. Então é calculado autovalores (λ_1 e λ_2) e determina se a janela corresponde a um canto.

- Se $\lambda_1 \approx 0$ e $\lambda_2 \approx 0$, então este pixel (x, y) não possui *features* interessante;
- Se $\lambda_1 \approx 0$ e λ_2 for algum valor positivo grande, uma bora será encontrada no canto;
- Se λ_1 e λ_2 tiveram alguma valor positivo grande, é encontrado um canto.

Q4: Explain the difference between the Euclidean distance and the cosine similarity metrics between descriptors. What might their geometric interpretations reveal about when each should be used? Given a distance metric, what is a good method for feature descriptor matching and why?

A4: A diferença entre a distância Euclidiana e métrica de similaridade de cosseno, é que a distância Euclidiana (ou métrica Euclidiana), em que quando é utilizado ele revela a distância entre dois pontos medida com uma régua, que pode ser comprovado pela aplicação repetida do teorema de Pitágoras. A distância Euclidiana para dois pontos $0 = (x_1, \dots, x_n)$ e $y = (y_1, \dots, y_n)$ no espaço n é definido como, pode ser na equação 1.

$$d(x, y) := \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \dots + (x_n - y_n)^2} = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \quad (1)$$

Já a métrica de similaridade de cosseno, é utilizada para calcular o cosseno do ângulo entre eles, uma medida de orientação e não magnitude. Para construir a equação de similaridade de cosseno é resolver a equação do produto para o $\cos\theta$, que pode ser visualizado na equação 2 e 3.

$$\vec{a} \cdot \vec{b} = \|\vec{a}\| \|\vec{b}\| \cos\theta \quad (2)$$

$$\cos\theta = \frac{\vec{a} \cdot \vec{b}}{\|\vec{a}\| \|\vec{b}\|} \quad (3)$$

A correspondência de descritor de *feature* é um dos estágios críticos em vários aplicativos de visão computacional, especialmente na recuperação de imagens, dimensionamento, etc. Desenvolver algoritmos eficientes e preciso para a correspondência de descritor de *feature*, afeta muito a eficiência e a precisão dos algoritmos de visão, então um bom método seria a qual alcança velocidade na busca rápida com uma alta precisão de correspondência.