THI GIAC MAY TINH 20/09/2022

Đặc trưng của biến cục bộ(Local Invariant Features)

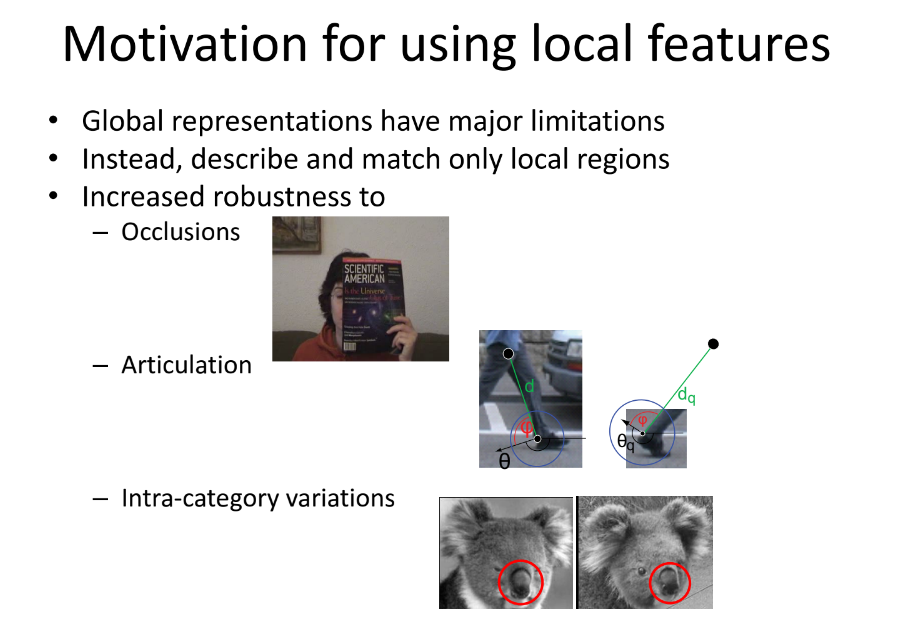
* Movation
* Requirements, invariances

Bài toán Image Matching:



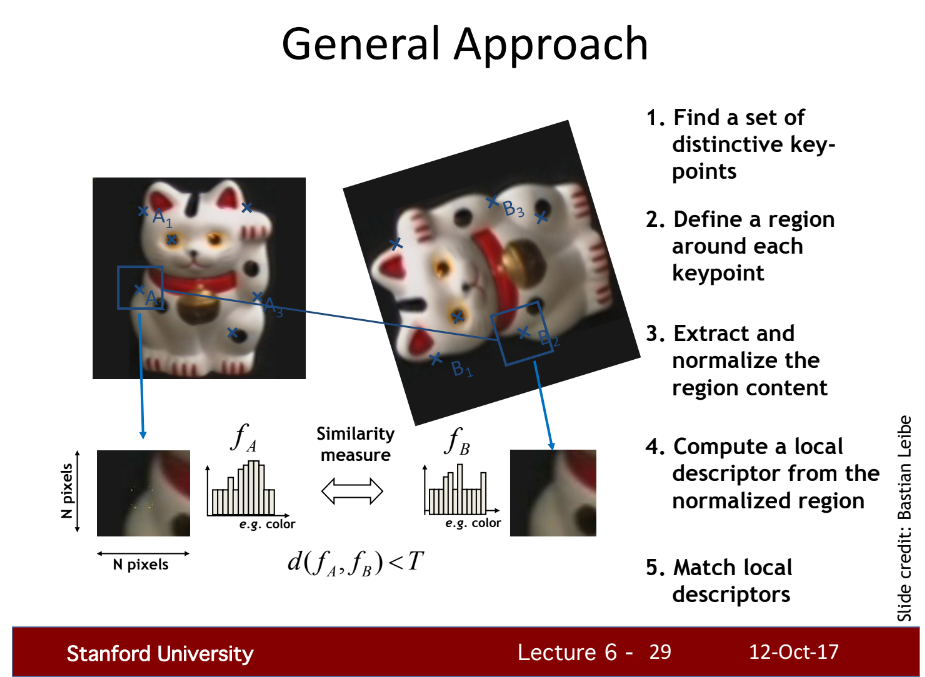
* So sánh ảnh khớp nhau
* VD: Bức tượng, mái vòm,... => đặc trưng nhỏ,(đặc trưng cục bộ bất biến, không thay đổi so với màu sắc, góc độ chụp)

Một số vấn đề:



1. Bị che khuất
2. Quay, biến dạng
3. Có nhiều điểm tương đồng

Thuật toán Image Matching:

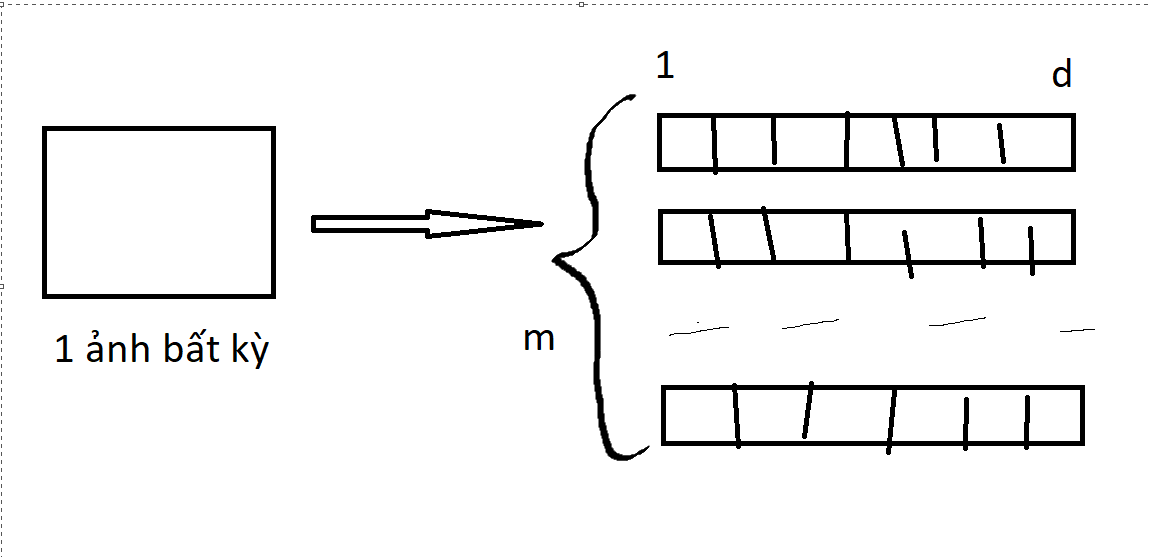


Bước 1: Tìm 1 tập điểm đặc trưng, đặc biệt trên ảnh(tập tọa độ điểm đặc trưng của ảnh) – local descriptor

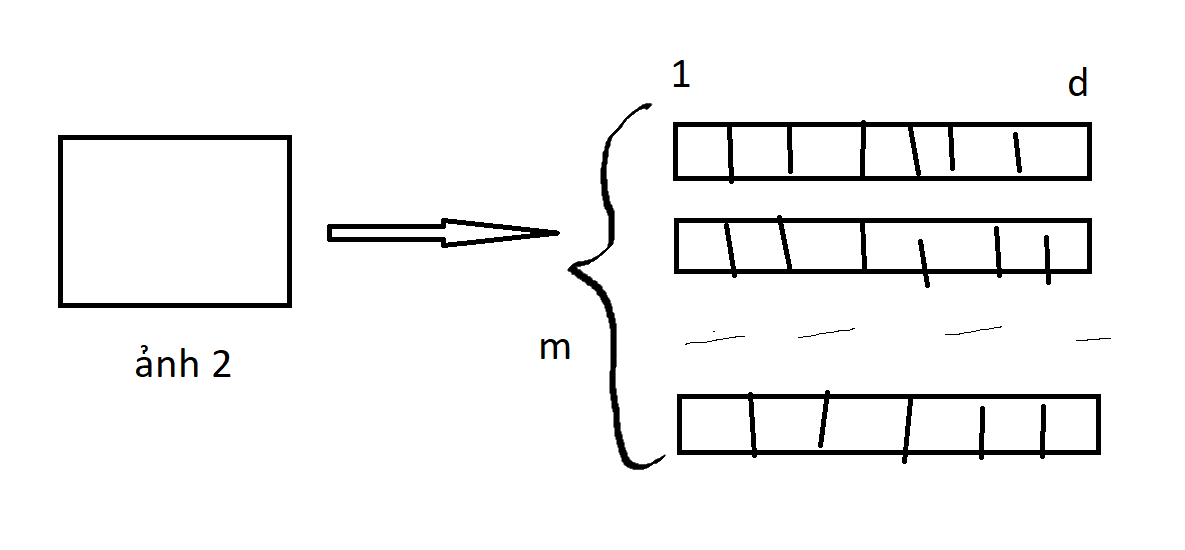
Bước 2: Định nghĩa(khoanh vùng tọa độ chứa các điểm đặc trưng)

Bước 3: Trích xuất vùng và chuẩn hóa nội dung trong vùng đó.

Bước 4: Tính đặc trưng trên vùng đặc biệt đã được chuẩn hóa đó.



1 ảnh bất kỳ -> trích xuất các đặc trưng được m thanh mô tả cục bộ - local descriptor => thanh mô tả được lưu trữ dưới dạng vector



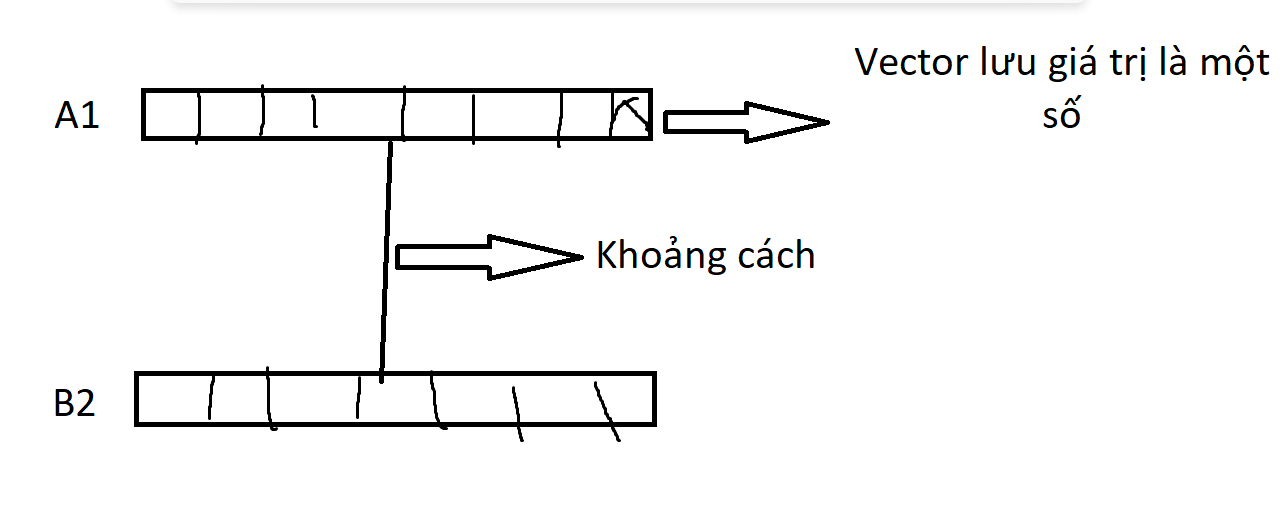
ảnh 2 cũng như vậy?

Ví dụ: ảnh 1 có thanh mô tả A1, ảnh 2 có thanh mô tả B2 làm sao để biết 2 thanh có giống nhau hay không?

* Câu trả lời là “Không” vì cho dù chụp cùng một vật thể mặc dù không thay đổi góc chụp hay độ sáng,... thì cơ bản vẫn là 2 tấm ảnh hoàn toàn khác nhau.

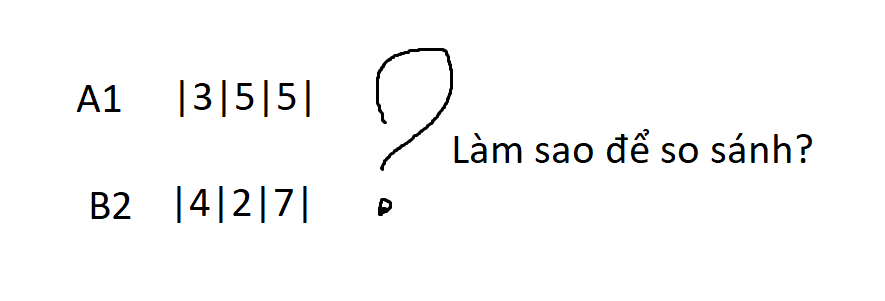
Bước 5: So sánh local descriptor của 2 ảnh

VD: Làm sao để biết thanh A1 của ảnh 1 có giống thanh B2 của ảnh 2?



* Ta sẽ tính khoảng cách từ A1 đến B2

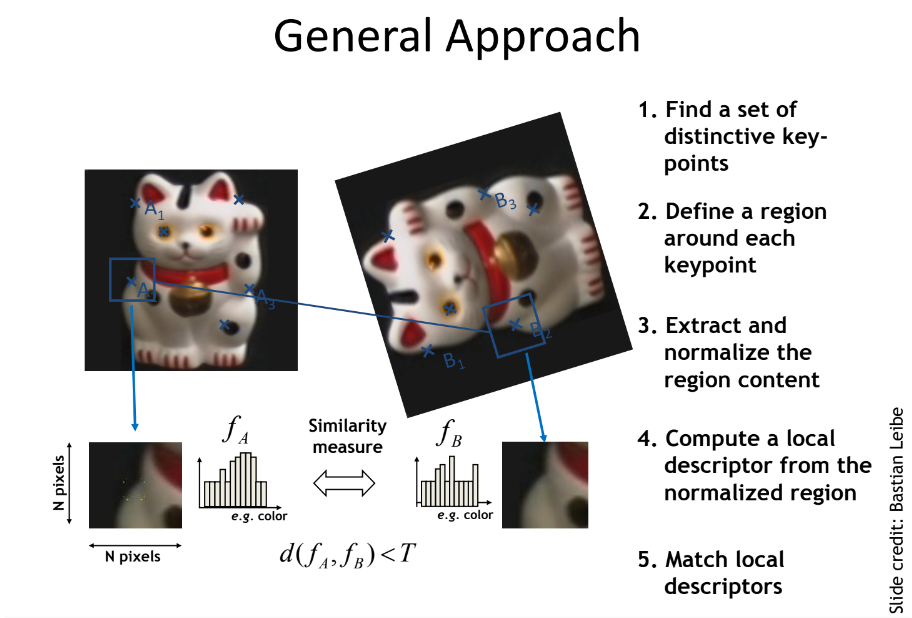
VD: Có 2 thanh A. B như hình, áp dụng công thức tính khoảng cách?



Áp dụng công thức khoảng cách Euclide

d = = so sánh với ngưỡng T

Xem slide thêm



Hoặc công thức L2:

=

Quay lại bước 1 để nói thêm:

Vấn đề 1: Tìm được các điểm giống nhau độc lập ở cả 2 hình ảnh

Vấn đề 2: Tìm được 1 điểm khớp với quá nhiều điểm của ảnh còn lại

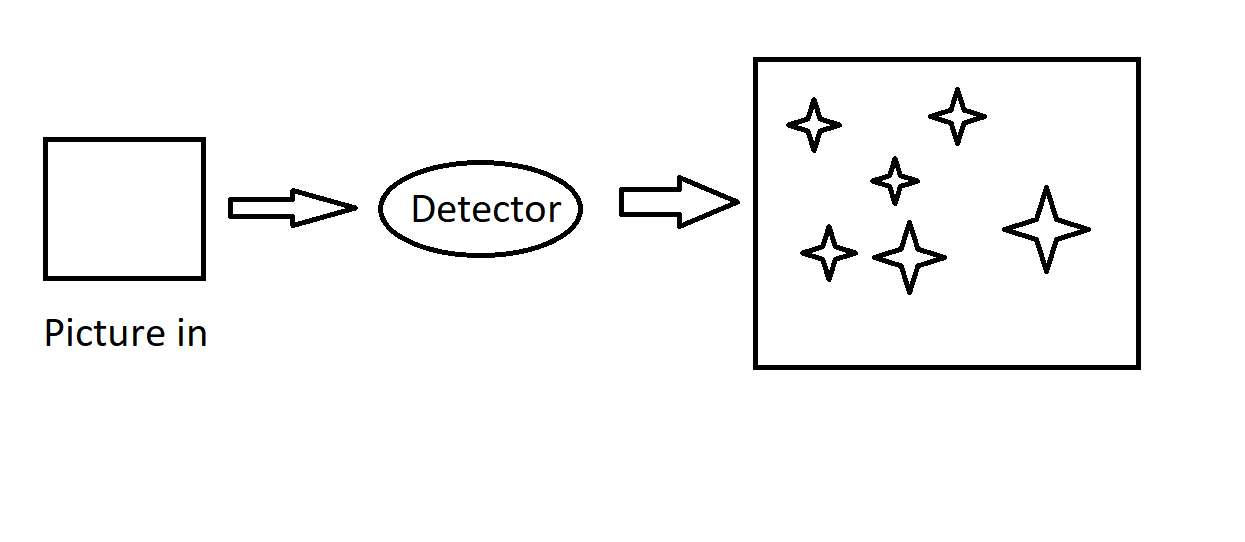
Solve:

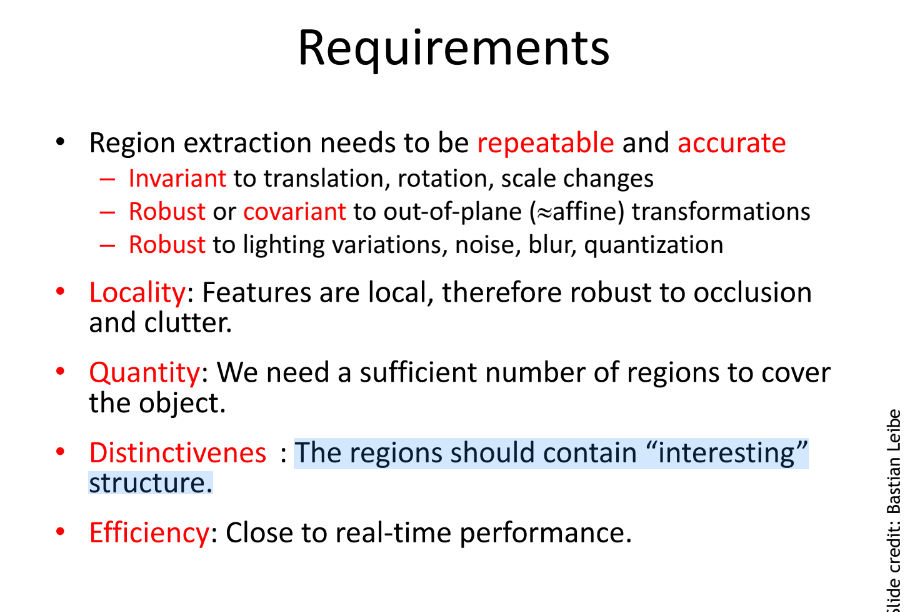
+ Một bộ phát hiện những điểm lặp lại.

+ Một bộ mô tả đặc trưng tốt nhất có thể.

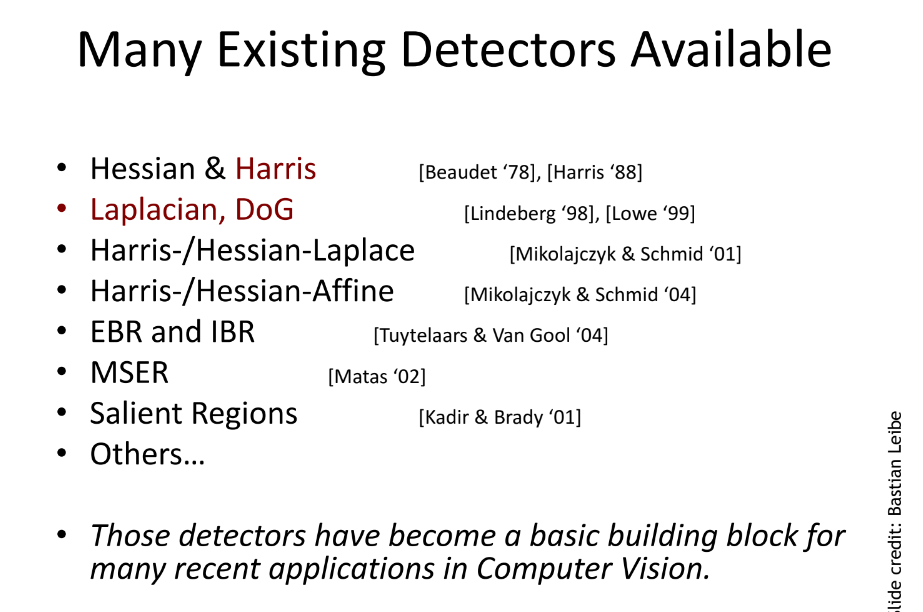
Vấn đề 3: Bất biến với các phép biến đổi hình học, biến đổi ánh sáng, màu sắc.

Yêu cầu của một bộ Detector:





* Lặp lại chính xác(ko lệch vị trí)
* Tính cục bộ(bộ mô tả) ko bị ảnh hưởng
* Số lượng: nhiều hay ít(phân lớp, gom cụm)
* Distinctiveness(phân biệt): khác với những điểm khác
* Hiệu quả(Effience): nhanh hay chậm



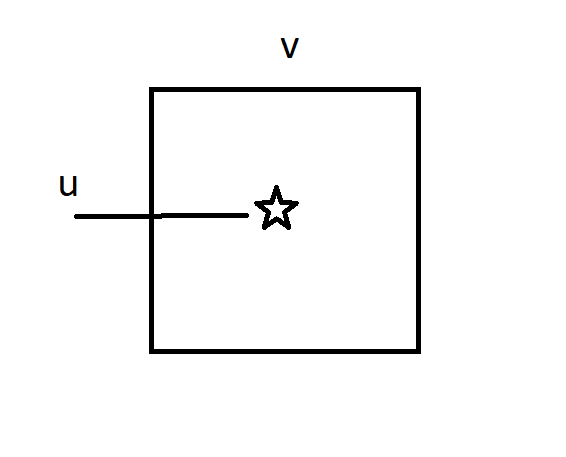
Tham khảo thêm

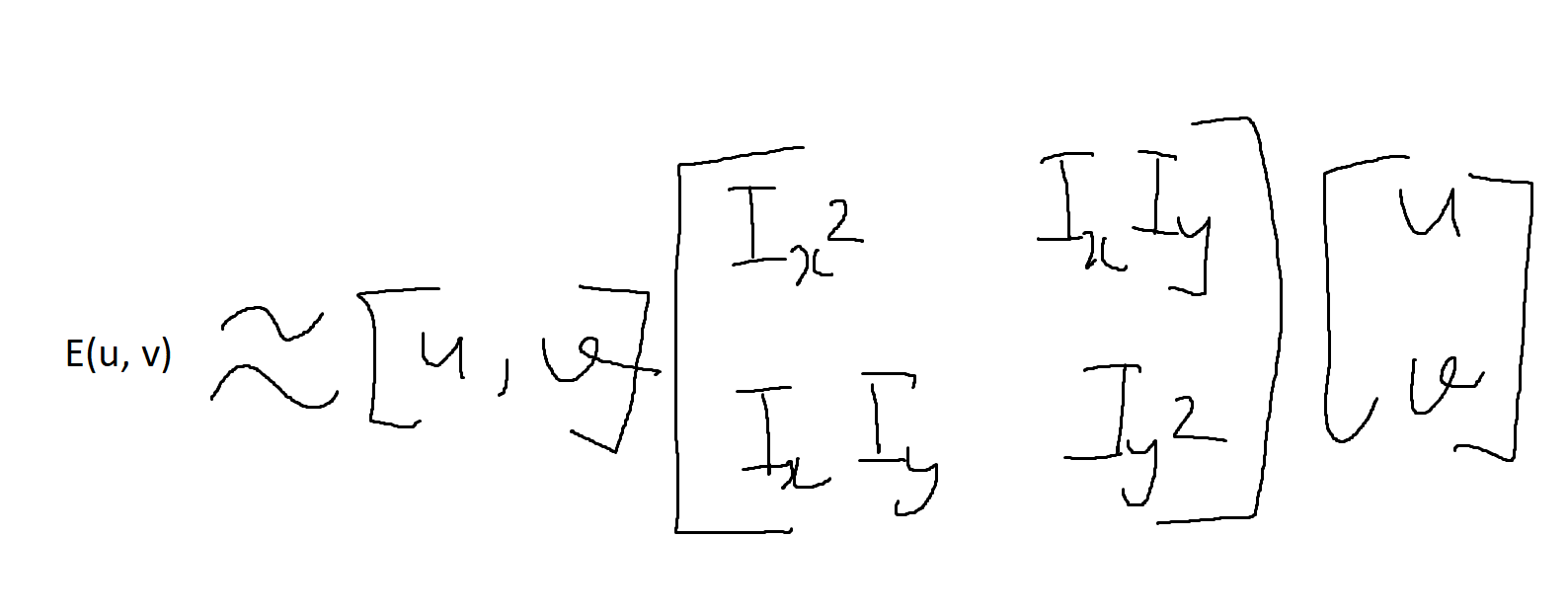
Thuật toán Harris Corner Dectector(Thuật toán phát hiện góc Harris)

Tìm góc(Corner):

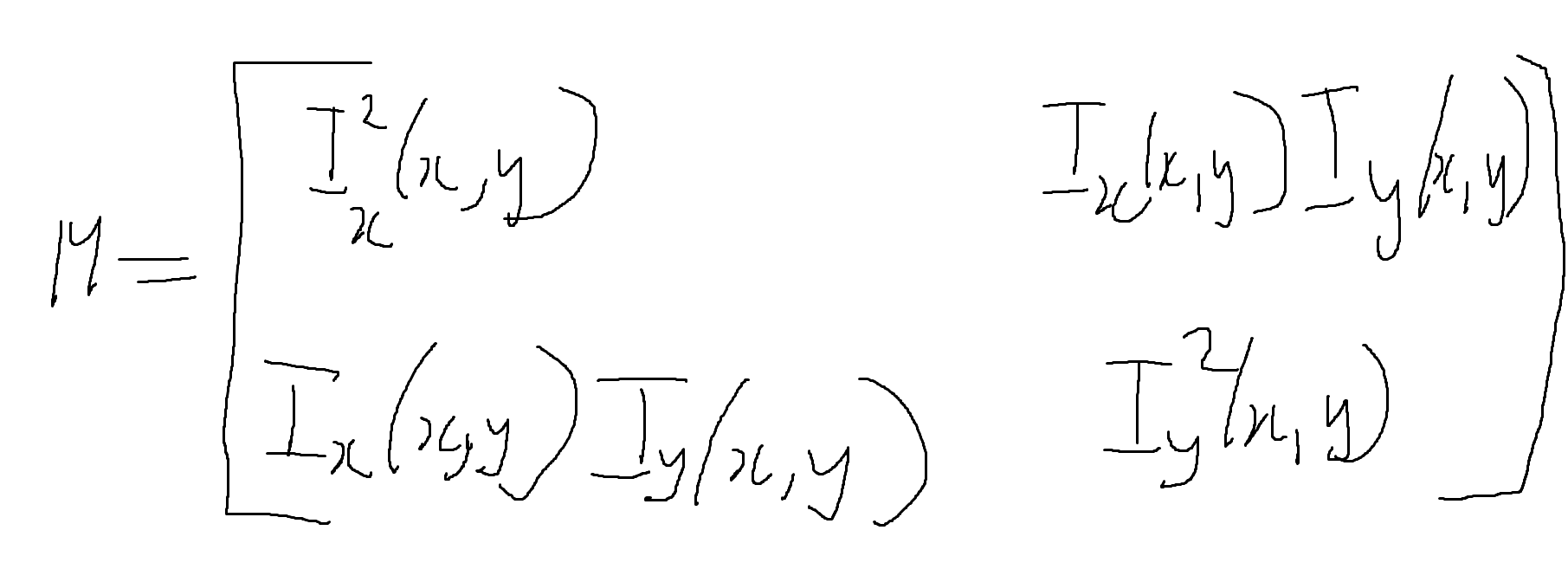
Đạo hàm cả x, y đều lớn thì => vùng chứa điểm đặc trưng đó là góc(corner)

Định nghĩa hàm mô tả sự thay đổi độ sáng xung quanh điểm của các điểm ảnh





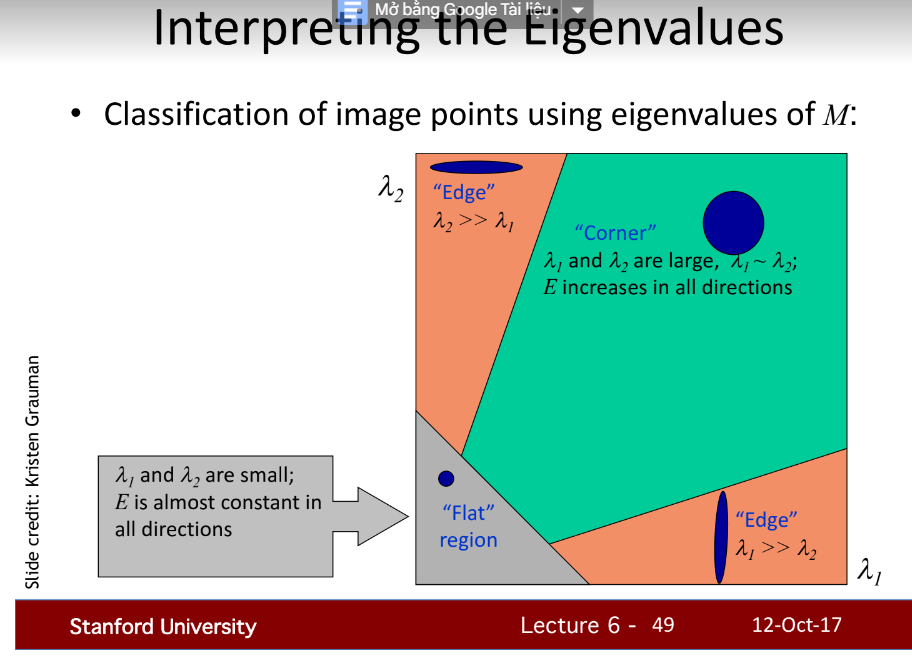
Với mỗi điểm ảnh xung quanh ta xây dựng M bằng



Nếu => khả năng cao là góc : Trường hợp này M là lý tưởng nhất  
  
Ngược lại: chéo hóa:

là các giá trị riêng của m và R là vector riêng của M(kiến thức này trong môn DSTT và HH)

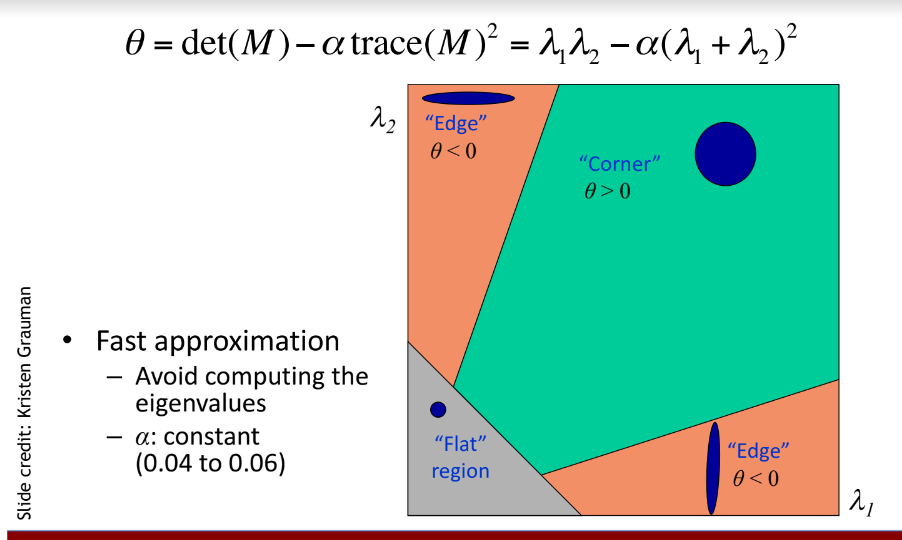
Để xác định điểm x, y có phải góc hay không ta sẽ xét các giá trị của



Định nghĩa khả năng là góc của điểm ảnh như sau:

= ngưỡng để so sánh

là giá trị đã được tính sẵn nằm trong khoảng 0,04~0,06



Cài đặt thuật toán:

1. Tính đạo hàm theo ,
2. Tính: , ,
3. Tính Gausse lên : , , (GaussianBlur)
4. Tính M sau khi dùng Gausse

=

1. Loại bỏ các điểm không phải maximum(như bài Canny)
2. Phân ngưỡng cuối:

Giữ lại các giá trị có ngưỡng (như hình trên)

BÀI TẬP VỀ NHÀ LÀM NỘP TỚI 23:59 THỨ 2 26/09

+ Sáng có điểm danh

+ Rảnh thì coi trước slide buổi sau.