

ĐẠI HỌC QUỐC GIA TP HCM
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA CÔNG NGHỆ THÔNG TIN
KỸ THUẬT PHẦN MỀM

Thuật toán phát hiện góc Harris

Đề tài: Báo cáo bài tập Lab 02

Môn học: Thị giác máy tính

Sinh viên thực hiện:

Lê Hoàng Sơn (21120127)

21120127@student.hcmus.edu.vn

Giáo viên hướng dẫn:

Phạm Minh Hoàng

Ngày 4 tháng 5 năm 2025



1 Hướng dẫn sử dụng

1.1 Cấu trúc thư mục

Chương trình phát hiện góc Harris được tổ chức trong thư mục **/Executable** theo cấu trúc thư mục sau:

- **Data/input**: Chứa các ảnh đầu vào
- **Data/output**: Chứa các ảnh đầu ra
- **21120127.exe**: File thực thi chương trình

1.2 Cách sử dụng

Để sử dụng chương trình, người dùng cần làm theo các bước sau:

- Để ảnh đầu vào vào thư mục **Data/input**
- Chạy file thực thi **21120127.exe** bằng tham số dòng lệnh theo cú pháp:

```
21120127.exe -harris <input_image_name> <output_image_name>
```

- Chương trình sẽ tự động đọc ảnh từ thư mục đầu vào, thực hiện phát hiện góc Harris và lưu ảnh kết quả vào thư mục đầu ra
- Kết quả sẽ được lưu dưới dạng ảnh với tên file được nhập từ dòng lệnh và lưu trong thư mục đầu ra **Data/output**

2 Phương pháp

2.1 Giới thiệu

Phát hiện góc Harris là một phương pháp phát hiện góc trong ảnh, được sử dụng phổ biến trong thị giác máy tính [1]. Nó được phát triển bởi Chris Harris và Mike Stephens vào năm 1988. Phương pháp này dựa trên việc phân tích biến thiên cục bộ của độ sáng trong một vùng nhỏ xung quanh

mỗi điểm ảnh để xác định các điểm ảnh có độ biến thiên lớn, tức là các góc trong ảnh.

Góc là những điểm ảnh có độ biến thiên lớn trong cả hai hướng x và y, tức là các điểm ảnh mà khi dịch chuyển một chút theo cả hai hướng này, độ sáng của chúng sẽ thay đổi mạnh mẽ. Điều này có nghĩa là các góc thường nằm ở các vị trí mà có sự thay đổi lớn về độ sáng, chẳng hạn như giao điểm của các đường thẳng hoặc các cạnh trong ảnh.

Phát hiện góc Harris có thể được sử dụng trong nhiều ứng dụng khác nhau trong thị giác máy tính, chẳng hạn như nhận diện đối tượng, theo dõi chuyển động và phân tích hình ảnh. Nó cũng là một phần quan trọng trong nhiều thuật toán xử lý ảnh khác, chẳng hạn như ghép ảnh (image stitching) và tái tạo 3D.

2.2 Thuật toán

Thuật toán phát hiện góc Harris [2] bao gồm các bước chính sau:

1. Chuyển ảnh về không gian xám (nếu ảnh đầu vào là ảnh màu) theo công thức:

$$I(x, y) = 0.299 \cdot R + 0.587 \cdot G + 0.114 \cdot B \quad (1)$$

Trong đó:

- $I(x, y)$ là giá trị độ sáng của pixel tại tọa độ (x, y)
- R, G, B là các kênh màu đỏ, xanh lá cây và xanh dương của ảnh màu

2. Tính toán ma trận gradient của ảnh:

Đối với $\text{gradientSize} = 3$:

$$\text{Sobel}_x = \frac{1}{4} \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \cdot \begin{bmatrix} -1 & 0 & 1 \end{bmatrix} \quad (2)$$

$$\text{Sobel}_y = (\text{Sobel}_x)^T \quad (3)$$

Đối với gradientSize = 5:

$$\text{Sobel}_x = \frac{1}{16} \begin{bmatrix} 1 \\ 4 \\ 6 \\ 4 \\ 1 \end{bmatrix} \cdot \begin{bmatrix} -1 & -2 & 0 & 2 & 1 \end{bmatrix} \quad (4)$$

$$\text{Sobel}_y = (\text{Sobel}_x)^T \quad (5)$$

Đối với gradientSize = 7:

$$\text{Sobel}_x = \frac{1}{64} \begin{bmatrix} 1 \\ 6 \\ 15 \\ 20 \\ 15 \\ 6 \\ 1 \end{bmatrix} \cdot \begin{bmatrix} -1 & -4 & -5 & 0 & 5 & 4 & 1 \end{bmatrix} \quad (6)$$

$$\text{Sobel}_y = (\text{Sobel}_x)^T \quad (7)$$

3. Tính toán ma trận covariance M cho mỗi pixel trong ảnh:

$$M = \begin{bmatrix} I(p)_x^2 & I(p)_x I(p)_y \\ I(p)_x I(p)_y & I(p)_y^2 \end{bmatrix} \quad (8)$$

Trong đó:

- p là tọa độ ảnh trong B , một cửa sổ kích thước $k \times k$ xung quanh pixel p
- $I(p)_x$ và $I(p)_y$ là gradient của ảnh tại pixel p theo hướng x và y
- $I(p)_x = I(p)_x * \text{Sobel}_x$ và $I(p)_y = I(p)_y * \text{Sobel}_y$

4. Tính toán giá trị Harris cho mỗi pixel trong ảnh bằng công thức:

$$R = \det(M) - k \cdot \text{trace}(M)^2 \quad (9)$$

Trong đó:

- k là một hằng số (thường được chọn là 0.04)
- $\det(M)$ là định thức của ma trận $M = I(p)_x^2 I(p)_y^2 - (I(p)_x I(p)_y)^2$
- $\text{trace}(M)$ là tổng các phần tử trên đường chéo chính của ma trận $M = I(p)_x^2 + I(p)_y^2$

Giá trị R sẽ cho biết mức độ góc tại pixel p . Nếu R lớn hơn một ngưỡng nhất định, pixel p sẽ được coi là một góc.

5. Áp dụng ngưỡng để lọc các điểm ảnh có giá trị Harris lớn hơn một ngưỡng nhất định. Các điểm này sẽ được coi là các góc trong ảnh.

Ở đây, chúng tôi sử dụng ngưỡng $\text{thresholdStrength} = 1e6$, tức là chỉ giữ lại các điểm ảnh có giá trị Harris lớn hơn 1 triệu.

6. Áp dụng thuật toán không cực trị (non-maximum suppression) để loại bỏ các điểm ảnh không phải là cực trị trong vùng lân cận của chúng. Điều này giúp giữ lại chỉ các điểm ảnh có giá trị Harris lớn nhất trong vùng lân cận.

- Tạo một bản sao của ảnh kết quả để lưu trữ kết quả xử lý.
- Với mỗi pixel tại tọa độ (x, y) trong ảnh (trừ các pixel biên):
 - So sánh giá trị của pixel hiện tại với tất cả các pixel lân cận trong cửa sổ có kích thước $\text{windowSize} \times \text{windowSize}$ xung quanh nó.
 - Nếu có bất kỳ pixel lân cận nào có giá trị lớn hơn pixel hiện tại, đặt giá trị pixel hiện tại trong ảnh kết quả thành 0.
 - Nếu pixel hiện tại có giá trị lớn nhất trong vùng lân cận, giữ nguyên giá trị của nó.
- Thay thế ảnh ban đầu bằng ảnh kết quả đã xử lý.

Kết quả cuối cùng là một tập các điểm có giá trị khác 0, đại diện cho các góc trong ảnh. Các điểm này là các cực đại cục bộ trong không gian đáp ứng Harris, đảm bảo rằng chỉ có các góc

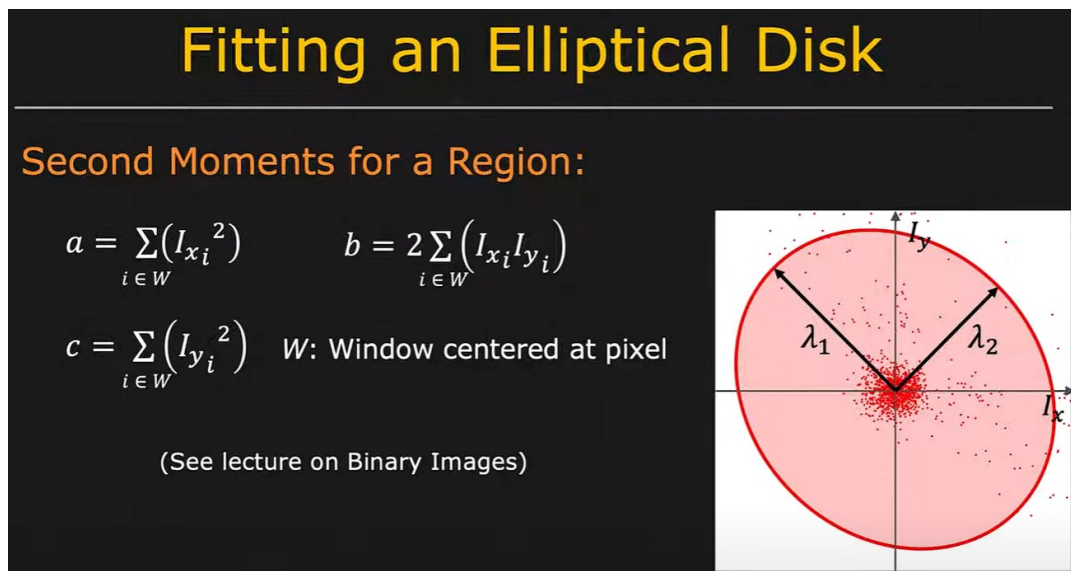
"mạnh" nhất được phát hiện và không có nhiều điểm góc được phát hiện gần nhau.

Ở đây chúng tôi sử dụng kích thước cửa sổ là 5×5 , tức là $k = 5$. Kích thước này được chọn để đảm bảo rằng các điểm ảnh trong vùng lân cận đủ lớn để phát hiện các góc mà không bị nhiễu bởi các chi tiết nhỏ trong ảnh.

- Cuối cùng, vẽ các điểm góc đã phát hiện lên ảnh gốc. Các điểm này sẽ được đánh dấu bằng một màu sắc khác biệt (thường là màu xanh) để dễ dàng nhận diện.

2.3 Minh họa thuật toán

2.3.1 Fitting an Elliptical Disk



Hình 1: Minh họa việc fitting một đĩa elliptical trong thuật toán phát hiện góc Harris [3]

Việc fitting một đĩa elliptical (elliptical disk) là một cách trực quan để hiểu về cách thuật toán Harris phát hiện góc. Ma trận M đại diện cho biến thiên cường độ ảnh trong cửa sổ lân cận của một điểm. Các giá trị riêng λ_1 và λ_2 của ma trận này xác định hình dạng của một ellipse.

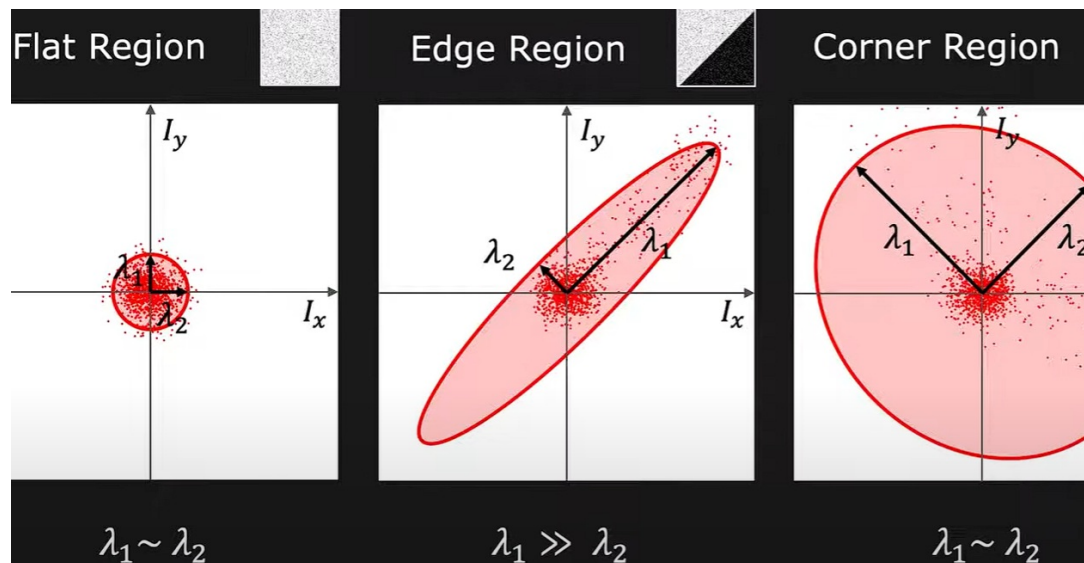
- Khi $\lambda_1 \approx 0$ và $\lambda_2 \approx 0$: Vùng ảnh phẳng, không có đặc điểm nổi bật
- Khi $\lambda_1 \approx 0$ và λ_2 lớn: Vùng ảnh là một cạnh
- Khi cả λ_1 và λ_2 đều lớn: Vùng ảnh chứa một góc

Ellipse Axes Lengths:

$$\lambda_1 = E_{max} = \frac{1}{2} \left[a + c + \sqrt{b^2 + (a - c)^2} \right]$$

$$\lambda_2 = E_{min} = \frac{1}{2} \left[a + c - \sqrt{b^2 + (a - c)^2} \right]$$

Hình 2: Minh họa tính toán phát hiện góc Harris bằng elliptical disk



Hình 3: Phân loại các vùng trong ảnh: vùng phẳng (flat region), cạnh (edge), và góc (corner)

Hình 3 minh họa cách phân loại các vùng trong ảnh dựa trên giá trị riêng λ_1 và λ_2 của ma trận M . Đây là cơ sở lý thuyết quan trọng giúp thuật toán Harris xác định được chính xác các góc trong ảnh.

Tương quan giữa Harris response score và so sánh λ_1 và λ_2 có thể được hiểu như sau: $R = \det(M) - k \cdot \text{trace}(M)^2$

R lớn khi λ_1 và λ_2 đều lớn, tức là khi cả hai giá trị riêng đều lớn hơn một ngưỡng nhất định. Điều này có nghĩa là vùng ảnh đó có độ biến thiên lớn trong cả hai hướng x và y, tức là có một góc.

R nhỏ khi λ_1 và λ_2 đều nhỏ, tức là khi cả hai giá trị riêng đều nhỏ hơn một ngưỡng nhất định. Điều này có nghĩa là vùng ảnh đó có độ biến thiên nhỏ trong cả hai hướng x và y, tức là không có đặc điểm nổi bật nào trong vùng đó.

R lớn khi λ_1 lớn và λ_2 nhỏ (hoặc ngược lại), tức là khi một trong hai giá trị riêng lớn hơn một ngưỡng nhất định trong khi giá trị còn lại nhỏ hơn. Điều này có nghĩa là vùng ảnh đó có độ biến thiên lớn trong một hướng (hướng x hoặc y) nhưng nhỏ trong hướng còn lại, tức là có một cạnh.

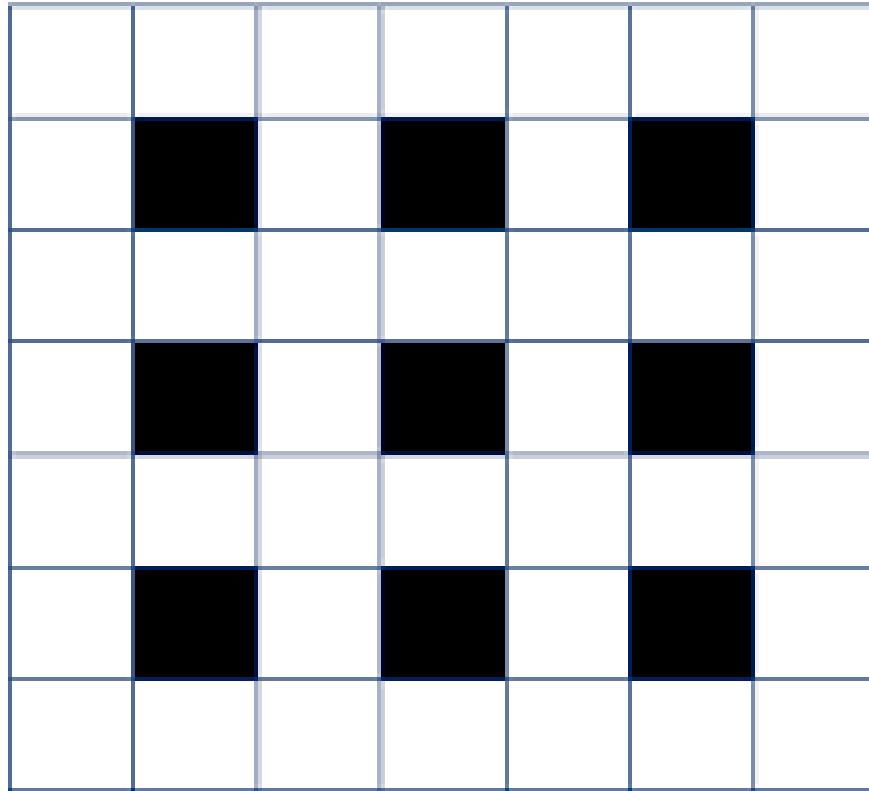
3 Thực nghiệm

3.1 Mô tả

Trong phần thực nghiệm, chúng tôi sử dụng hai ảnh thử nghiệm chính:



Hình 4: Ảnh ngôi nhà - chứa nhiều góc cạnh rõ ràng và cấu trúc hình học



Hình 5: Ảnh bàn cờ - mẫu hình học cơ bản với các góc rõ ràng

Hai ảnh thử nghiệm này được chọn vì đặc điểm khác nhau của chúng:

- **Ảnh ngôi nhà:** Chứa nhiều đặc điểm hình học phức tạp như góc cạnh của mái nhà, cửa sổ, và tường. Ảnh này giúp đánh giá khả năng phát hiện góc trong môi trường thực tế với nhiều chi tiết và nhiễu.
- **Ảnh bàn cờ đen trắng:** Đại diện cho mẫu hình học cơ bản với các góc rõ ràng, sắc nét và tương phản cao. Đây là trường hợp lý tưởng để kiểm tra độ chính xác của thuật toán Harris trong điều kiện tối ưu.

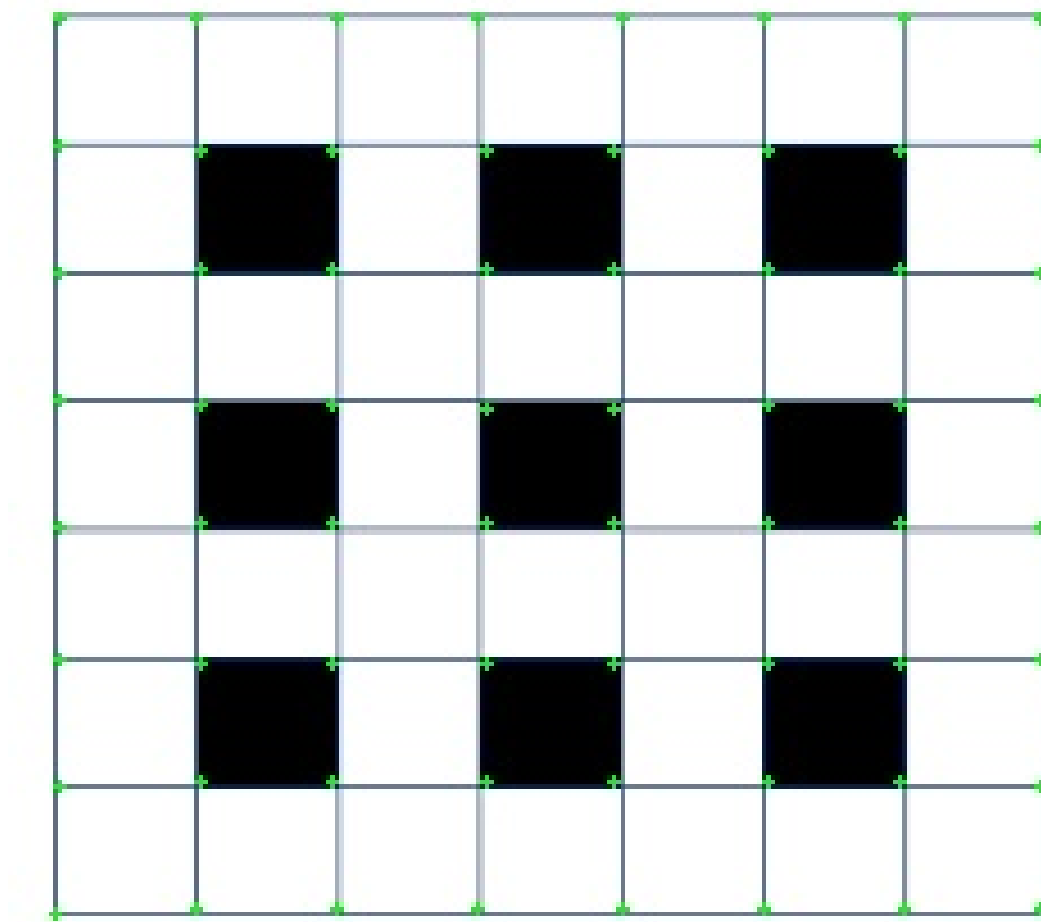
3.2 Kết quả

3.2.1 Kết quả thử nghiệm với ảnh ngôi nhà



Hình 6: Kết quả phát hiện góc Harris trên ảnh ngôi nhà

3.2.2 Kết quả thử nghiệm với ảnh bàn cờ



Hình 7: Kết quả phát hiện góc Harris trên ảnh bàn cờ

3.3 Phân tích

Hình 6 hiển thị kết quả của thuật toán phát hiện góc Harris trên ảnh ngôi nhà. Các điểm màu xanh đánh dấu các vị trí được xác định là góc. Có thể thấy thuật toán đã phát hiện thành công các góc tại các vị trí như góc của mái nhà, cửa sổ và các cạnh nơi có sự thay đổi đột ngột về cường độ sáng. Do đây là hình phức tạp, nên thuật toán phát hiện được nhiều góc hơn so với ảnh bàn cờ. Tuy nhiên, một số góc có thể không được phát hiện hoặc bị nhiễu do các yếu tố như ánh sáng không đồng đều, độ tương phản thấp hoặc do tham số ngưỡng không được điều chỉnh phù hợp.

Hình 7 thể hiện kết quả của thuật toán phát hiện góc Harris trên ảnh bàn cờ. Với mẫu hình học đơn giản và rõ ràng này, thuật toán đã phát hiện chính xác các góc tại giao điểm của các ô cờ đen và trắng. Đây là minh chứng cho độ chính xác cao của thuật toán trong trường hợp lý tưởng với

các góc rõ ràng và tương phản cao.

3.4 Kết luận

Trong báo cáo này, chúng tôi đã trình bày chi tiết về thuật toán phát hiện góc Harris, từ lý thuyết đến thực nghiệm. Chúng tôi đã sử dụng hai ảnh thử nghiệm khác nhau để đánh giá hiệu suất của thuật toán. Kết quả cho thấy thuật toán hoạt động tốt trong việc phát hiện các góc trong ảnh, đặc biệt là trong các trường hợp có độ tương phản cao và hình học rõ ràng.

Tài liệu

- [1] “Harris Corner Detector,” Wikipedia, https://en.wikipedia.org/wiki/Harris_corner_detector
- [2] “Harris Corner Detection,” NVIDIA VPI Documentation, https://docs.nvidia.com/vpi/algo_harris_corners.html
- [3] “Harris Corner Detector Explained,” YouTube, https://www.youtube.com/watch?v=Z_HwkG90Yvw