Báo cáo cuối kì

Nhận diện bi-a và tìm đường đi trong game 8 Ball Pool

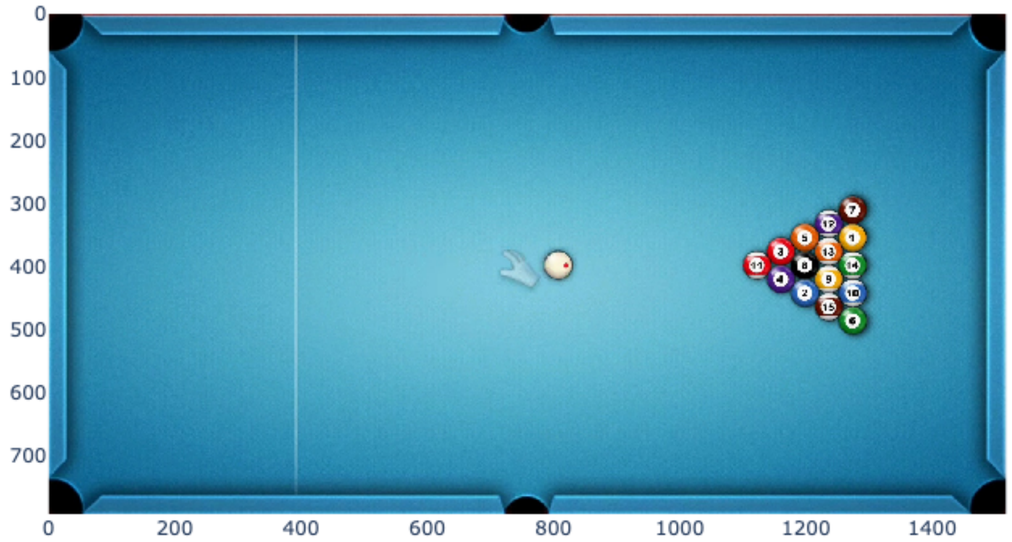
Bùi Khánh Duy - 20001898 - K65A3 - Nhóm 2
Giảng viên: Th.S Hà Mạnh Toàn

Mục lục

Phát biểu bài toán

- **Input:**
  - Video quay lại quá trình chơi game
- **Output:**
  - Video sau khi thêm kết quả
- **Quá trình thực hiện:**
  - Xây dựng chương trình

Phát hiện các góc

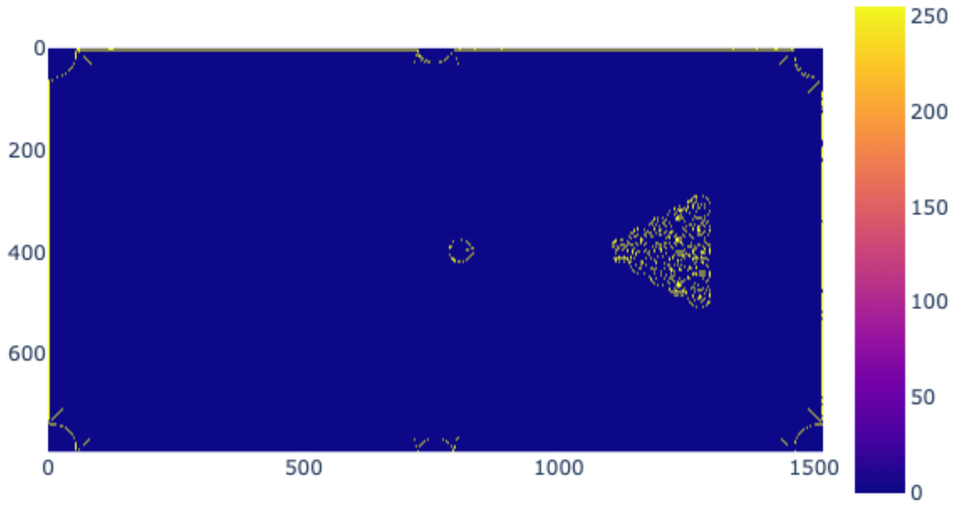


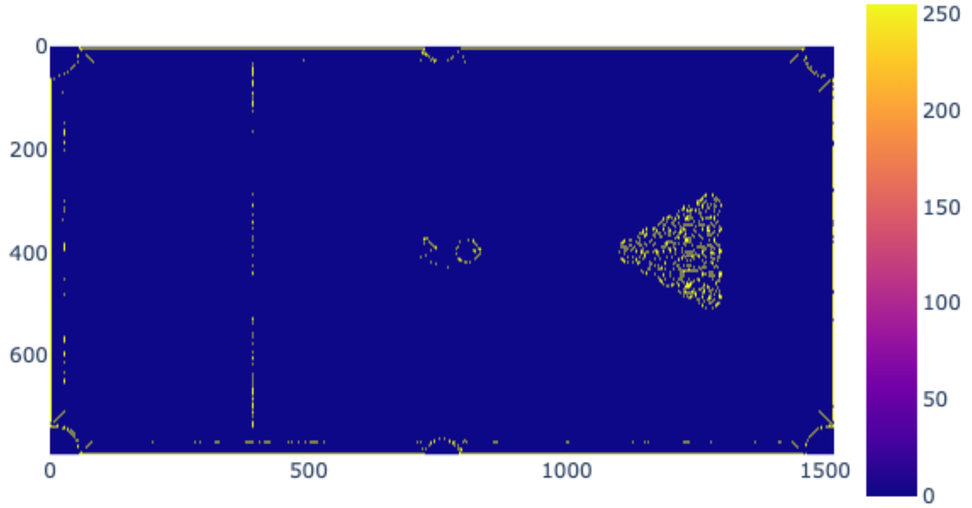
Phát hiện các viên bi

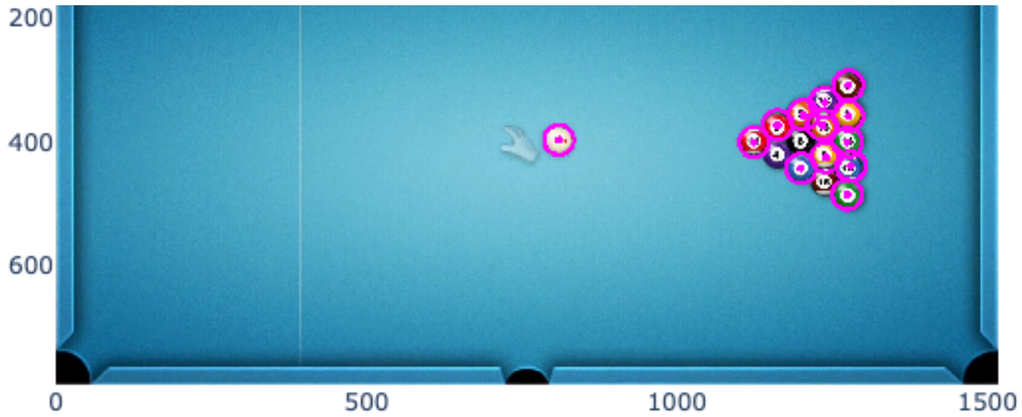
Áp dụng làm sắc

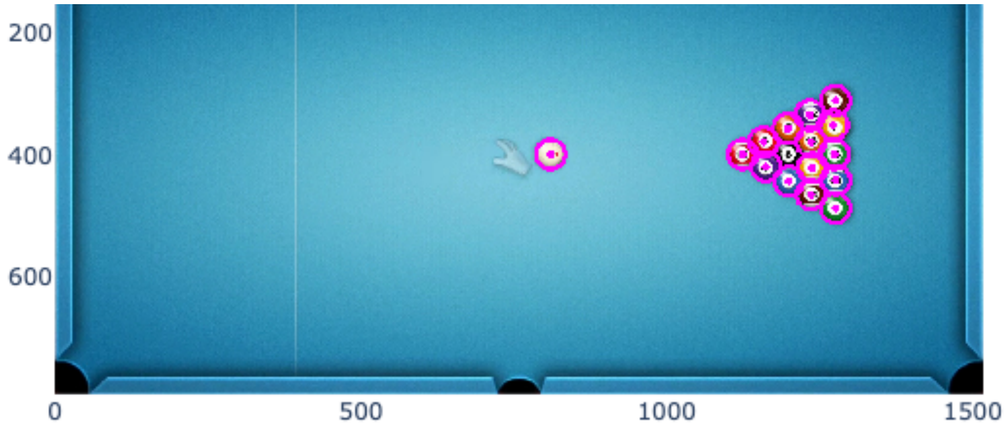
- Nguyên lý:
  - Trừ ảnh gốc với ảnh được làm mờ sử dụng Gaussian Blur, theo cách có trọng số để giữ giá trị của một vùng không đổi).
  - Kết quả: Giúp việc phát hiện viên bi trong ảnh tốt hơn

Trước / Sau


0
200
400
600
0
500
1000
1500
250
200
150
100
50
0

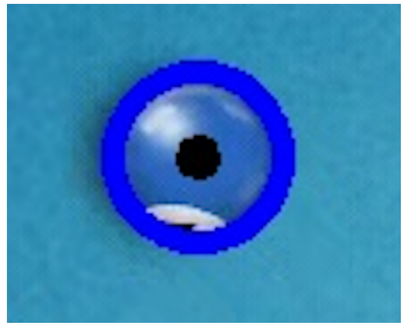
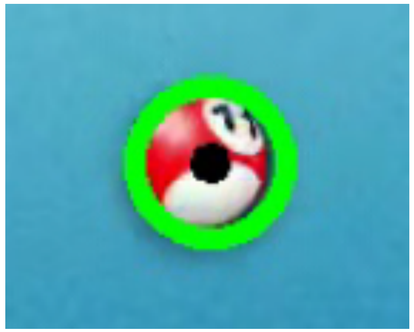
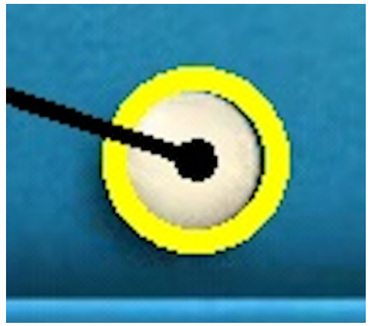

0
200
400
600
0
500
1000
1500
250
200
150
100
50
0


200
400
600
0
500
1000
1500


200
400
600
0
500
1000
1500

Phân loại bằng đếm số điểm ảnh

- Solid_ball: `color_count / total_count >= 0.8`
- Stripped_ball: `color_count / total_count >= 0.2`
- Black_ball: `black_count / total_count >= 0.5`
- White_ball: `white_count / total_count >= 0.89`


Tìm đường đi

Logic

- Tìm viên bi trắng
- Tìm các viên bi thuộc loại của người chơi, mặc định là bi trơn (Stripped)
- Bi đen sẽ được chọn cuối cùng
- Tìm lỗ gần nhất

Đảm bảo các điều kiện sau:

- Không có vật cản từ bi trắng đến bi mục tiêu
- Không có vật cản từ bi mục tiêu các lỗ.

Xây dựng đồ thị

- Coi tập các viên bi là tập các đỉnh (trừ viên bi trắng)
- Xây dựng tập các cạnh trong đồ thị:
  - Với mỗi cặp 2 đỉnh:
    - Nếu tạo ra cạnh hợp lệ: thêm vào tập cạnh với trọng số = khoảng cách của hai đỉnh,
    - Ngược lại, thêm vào tập cạnh trọng số = khoảng cách của hai đỉnh + 10000
      - Cạnh hợp lệ là cạnh không giao với bi thứ 3 hoặc viền của bàn bi-a

Dijkstra

Để tìm đường ngắn nhất

- Thực hiện tìm đường đi lần lượt từ bi cái đến 6 lỗ
- Đường đi được chọn sẽ có độ dài ngắn nhất
- Input: Đỉnh xuất phát và đỉnh kết thúc
- Output: Đường đi ngắn nhất, dưới dạng danh sách

Dijkstra
Thực hiện

1. Khởi tạo từ điển shortest_paths với khoá là đỉnh hiện tại và giá trị là (None, 0) tương ứng cho (đỉnh đã đi qua, tổng trọng số của đường đi)
2. Tạo set visited để đánh dấu các đỉnh đã đi qua
3. Thực hiện vòng lặp:
    - Thêm đỉnh hiện tại vào visited
    - Duyệt tất cả các đỉnh lân cận (có cạnh với đỉnh hiện tại):
        - Lấy ra tổng trọng số của đỉnh hiện tại từ tập shortest_paths
        - Cập nhật lại tổng trọng số của đỉnh lân cận từ vị trí hiện tại, nếu tổng trọng số mới bé hơn giá trị hiện tại
4. Sau khi kết thúc vòng lặp, xây dựng đường đi ngắn nhất bằng cách duyệt ngược từ đỉnh cuối đến đỉnh bắt đầu. Kết quả là danh sách đường đi sau khi đảo ngược lại.

Các vấn đề

- Tốc độ xử lý chưa đủ nhanh, dù một số phần tính toán với các vector được thêm thư viện numba, thể hiện ở video kết quả
- Độ chính xác vẫn chưa được như kì vọng, hiện tượng nhận diện nhầm xảy ra khá nhiều.
- Nếu các viên bi nằm ở trước cửa lỗ thì không nhận diện được.