



TRƯỜNG ĐẠI HỌC NGOẠI NGỮ - TIN HỌC THÀNH PHỐ HỒ CHÍ MINH
KHOA CÔNG NGHỆ THÔNG TIN
○📖○

BÁO CÁO KẾT THÚC HỌC PHẦN
THỊ GIÁC MÁY TÍNH

XÂY DỰNG THUẬT TOÁN TRỪ NỀN **(BACKGROUND SUBTRACTION)**

Giảng viên hướng dẫn: **TS. Nguyễn Thanh Bình**

Sinh viên thực hiện:

- | | |
|---------------------|------------|
| 1. Nguyễn Gia Huy | 22DH111255 |
| 2. Bùi Tuấn Kiệt | 22DH111797 |
| 3. Nguyễn Quốc Minh | 22DH112165 |
| 4. Lê Phạm Hoàng Vũ | 22DH114826 |

Thành phố Hồ Chí Minh, tháng 11 năm 2024



TRƯỜNG ĐẠI HỌC NGOẠI NGỮ - TIN HỌC THÀNH PHỐ HỒ CHÍ MINH
KHOA CÔNG NGHỆ THÔNG TIN
○ ○

BÁO CÁO KẾT THÚC HỌC PHẦN
THỊ GIÁC MÁY TÍNH

XÂY DỰNG THUẬT TOÁN TRỪ NỀN **(BACKGROUND SUBTRACTION)**

Mã lớp học phần: **241123056402**

Năm học: **2024 – 2025**

Học kỳ: **1**

Sinh viên thực hiện:

- | | |
|---------------------|------------|
| 1. Nguyễn Gia Huy | 22DH111255 |
| 2. Bùi Tuấn Kiệt | 22DH111797 |
| 3. Nguyễn Quốc Minh | 22DH112165 |
| 4. Lê Phạm Hoàng Vũ | 22DH114826 |

CONTENTS

CHƯƠNG 1. GIỚI THIỆU	1
1.1. Giới thiệu đề tài	1
1.2. Mục tiêu và nội dung đề tài	1
1.3. Giới hạn đề tài.....	1
1.4. Cấu trúc báo cáo	2
CHƯƠNG 2. Cơ sở lý thuyết.....	3
2.1. Giới thiệu tập dữ liệu	Error! Bookmark not defined.
2.2. Giới thiệu lý thuyết về đề tài	3
2.2.1. <i>Khái niệm</i>	3
2.2.2. <i>Nguyên lý hoạt động</i>	3
CHƯƠNG 3. Phương pháp tRỪ NỀN (backGROUND SUBTRACTION)	4
3.1. Yêu cầu bài toán	4
3.1.1. <i>Input</i>	4
3.1.2. <i>Output</i>	4
3.2. Giải thuật thực hiện(MOG2)	4
3.3. Phương pháp đánh giá.....	6
CHƯƠNG 4. HIỆN THỰC KẾT QUẢ.....	8
4.1. Yêu cầu hệ thống và tập dữ liệu thực nghiệm	8
4.2. Kết quả thực nghiệm	8
CHƯƠNG 5. KẾT LUẬN.....	9
5.1. Kết quả đạt được	9
5.2. Ưu và nhược điểm.....	9
5.2.1. <i>Ưu điểm của phương pháp MOG2:</i>	9
5.2.2. <i>Nhược điểm của phương pháp này:</i>	9
5.3. Hướng mở rộng tương lai.....	10
Phụ LỤC cODE	12
TÀI LIỆU THAM KHẢO	18

CHƯƠNG 1. GIỚI THIỆU

1.1. Giới thiệu đề tài

Thuật toán Trừ Nền (Background Subtraction) là một kỹ thuật quan trọng trong xử lý ảnh và thị giác máy tính, giúp tách biệt các đối tượng chuyển động ra khỏi nền trong video hoặc chuỗi hình ảnh. Mục tiêu chính của phương pháp này là nhận diện các phần của khung hình không thay đổi theo thời gian và loại bỏ chúng, chỉ giữ lại những phần chuyển động để phân tích. Điều này rất hữu ích trong các ứng dụng như giám sát an ninh, theo dõi giao thông và phân tích hành vi. Có nhiều phương pháp để thực hiện trừ nền, từ các thuật toán đơn giản như trừ trung bình và chạy trung vị, đến các phương pháp phức tạp hơn như mô hình nền Gaussian hỗn hợp (GMM) hay các kỹ thuật tiên tiến được tích hợp trong các thư viện như OpenCV, bao gồm MOG2 và KNN. Việc lựa chọn thuật toán phù hợp phụ thuộc vào đặc điểm cụ thể của ứng dụng, mức độ phức tạp của nền và yêu cầu về thời gian xử lý.

1.2. Mục tiêu và nội dung đề tài

Với đề tài về giải thuật MOG2 để phân tích chuyển động trong video, bạn sẽ tập trung vào một phương pháp phổ biến trong xử lý ảnh và trừ nền. MOG2 (Mixture of Gaussians Version 2) là một thuật toán mạnh mẽ được sử dụng rộng rãi để xây dựng mô hình nền với khả năng thích nghi với những thay đổi chậm trong cảnh quay, như sự di chuyển của bóng, thời tiết, hoặc thay đổi ánh sáng.

Giải thuật này hoạt động bằng cách mô hình hóa mỗi pixel trong nền dưới dạng một hỗn hợp của các phân phối Gaussian. Khi một điểm ảnh không phù hợp với mô hình nền, nó được đánh dấu là một phần của đối tượng chuyển động. MOG2 cũng có khả năng phát hiện và cập nhật nền hiệu quả, giúp phân biệt rõ ràng giữa đối tượng chuyển động và nền phức tạp. Đây là lựa chọn lý tưởng cho các ứng dụng như giám sát giao thông, phân tích hành vi và theo dõi đối tượng trong video.

1.3. Giới hạn đề tài

Đề tài này sẽ tập trung vào việc ứng dụng và phân tích thuật toán MOG2 trên các video có độ phân giải vừa phải và nền không quá phức tạp. Các tình huống như ánh sáng

CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI

thay đổi đột ngột, bóng đổ, và các yếu tố khác có thể ảnh hưởng đến độ chính xác sẽ được nghiên cứu sơ bộ nhưng không phải là mục tiêu tối ưu hóa chính của đề tài.

1.4. Cấu trúc báo cáo

Phần 1: Giới thiệu – Cung cấp tổng quan về đề tài, nêu mục tiêu, nội dung và giới hạn của nghiên cứu.

Phần 2: Cơ sở lý thuyết – Trình bày các khái niệm cơ bản về kỹ thuật xử lý ảnh và giải thuật Trừ Nền, đặc biệt là mô hình nền Gaussian hỗn hợp (MOG2) và các giải thuật liên quan.

Phần 3: Phương pháp – Mô tả chi tiết yêu cầu của bài toán, giải thuật MOG2 và cách đánh giá hiệu quả thực hiện.

Phần 4: Hiện thực kết quả – Cung cấp thông tin về hệ thống thực nghiệm, dữ liệu sử dụng, kết quả và phân tích thực nghiệm.

Phần 5: Kết luận – Tổng kết những kết quả đạt được, chỉ ra ưu và nhược điểm của phương pháp, đồng thời đề xuất hướng phát triển trong tương lai.

Tài liệu tham khảo – Liệt kê các tài liệu và nguồn tham khảo.

Phụ lục code demo – Đính kèm mã nguồn minh họa cho giải thuật và các bước xử lý.

CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

2.1. Giới thiệu lý thuyết về đề tài

2.1.1. Khái niệm

Giải thuật trừ nền là một kỹ thuật trong thị giác máy tính dùng để tách vật chuyển động ra khỏi hình nền trong video hoặc hình ảnh. Dùng để phát hiện vật trong hình ảnh hoặc trong video

2.1.2. Nguyên lý hoạt động

Trừ nền hoạt động dựa trên nguyên lý rằng nền của một khung hình (frame) trong video thường không thay đổi theo thời gian, trong khi các đối tượng chuyển động sẽ thay đổi. Quá trình trừ nền bao gồm các bước chính:

1. **Khởi tạo nền:** Xây dựng hoặc ước lượng một mô hình của nền từ một chuỗi khung hình.
2. **Phân biệt nền và đối tượng:** So sánh từng khung hình hiện tại với mô hình nền để phát hiện các điểm ảnh có sự thay đổi, đại diện cho các đối tượng chuyển động.
3. **Cập nhật nền:** Điều chỉnh mô hình nền theo thời gian để thích ứng với những thay đổi nhỏ trong cảnh nền, như ánh sáng hoặc các vật thể nhỏ di chuyển.

CHƯƠNG 3. PHƯƠNG PHÁP TRỪ NỀN (BACKGROUND SUBTRACTION)

3.1. Yêu cầu bài toán

trong đề tài này liên quan đến việc phân tích chuyển động trong video bằng thuật toán MOG2 để trừ nền. Cụ thể:

3.1.1. Input

Một video hoặc chuỗi hình ảnh chứa cảnh quay với các đối tượng chuyển động và nền tĩnh.

Các video có độ phân giải vừa phải và nền không quá phức tạp.

Có thể có những yếu tố gây nhiễu như bóng đổ, thay đổi ánh sáng nhẹ.

3.1.2. Output

Video đã qua xử lý, trong đó các đối tượng chuyển động được tách biệt khỏi nền và được đánh dấu rõ ràng.

Hình ảnh nhị phân hoặc các khung hình trong đó các đối tượng chuyển động được thể hiện bằng các vùng màu trắng trên nền đen (foreground mask).

Thông kê về số lượng đối tượng được phát hiện trong mỗi khung hình.

Kết quả phân tích về độ chính xác trong việc phát hiện đối tượng chuyển động và xử lý các trường hợp nhiễu.

3.2. Giải thuật thực hiện(MOG2)

Bước 1: Khởi tạo Mô hình Gaussian cho từng Điểm ảnh:

- Với mỗi điểm ảnh, MOG2 khởi tạo một hỗn hợp gồm K Gaussian. Mỗi Gaussian đại diện cho một trạng thái nền tiềm năng của điểm ảnh đó, các thông số của mỗi Gaussian bao gồm(Mean, Variance, Weight).

Bước 2 : Phân loại pixel và cập nhật Mô hình Gaussian cho Điểm ảnh Mới

Khi một khung hình mới được xử lý:

a. Phân Loại Pixel :

-Giá trị pixel được kiểm tra xem nó có khớp với bất kỳ Gaussian nào trong mô hình nền hay không

Một pixel được coi là khớp với một Gaussian nếu:

$$|x - \mu| < k \cdot \sigma$$

b. Cập nhật mô hình Gaussian:

-Nếu pixel khớp với một Gaussian:

1. **Cập nhật trọng số (w):**

Trọng số w của Gaussian khớp sẽ tăng lên, phản ánh rằng Gaussian này hiện tại đại diện tốt hơn cho pixel

2. **Cập nhật giá trị trung bình**

Giá trị trung bình của Gaussian được cập nhật bằng cách sử dụng tốc độ học:

$$\mu^{(t)} = (1 - \rho) \cdot \mu^{(t-1)} + \rho \cdot x$$

3. **Cập nhật phương sai**

Phương sai của Gaussian được điều chỉnh để phản ánh sự thay đổi trong giá trị pixel

$$\sigma^{2(t)} = (1 - \rho) \cdot \sigma^{2(t-1)} + \rho \cdot (x - \mu^{(t)})^2$$

Nếu pixel không khớp với bất kỳ Gaussian nào:

1. Khởi tạo một Gaussian mới với giá trị trung bình bằng giá trị pixel hiện tại

2. Trọng số ban đầu của Gaussian này nhỏ và phương sai lớn

Bước 3: Xác định Nền và Foreground :

- Sau khi đã cập nhật các Gaussian, thuật toán sắp xếp các Gaussian của điểm ảnh đó theo thứ tự trọng số từ cao xuống thấp.

- Các Gaussian có **trọng số cao nhất** và **phương sai nhỏ** thường được coi là nền,

vì

chúng mô tả những trạng thái ổn định và ít thay đổi của điểm ảnh

- Các Gaussian khác có thể được xem là **foreground** nếu điểm ảnh hiện tại không khớp

3.3. Phương pháp đánh giá

Để đánh giá hiệu suất của hệ thống phát hiện đối tượng trong video, chúng tôi đã sử dụng ba thang đo chính là **Precision**, **Recall**, và **F1 Score**. Các thang đo này cho phép đánh giá độ chính xác và mức độ bao phủ của mô hình trong việc phát hiện các đối tượng chuyển động trong khung hình.

3.3.1. Precision:

Precision đo lường mức độ chính xác của các đối tượng được phát hiện so với số lượng đối tượng thực tế. Nó được tính bằng tỷ lệ giữa số lượng đối tượng phát hiện đúng (True Positives) với tổng số đối tượng phát hiện được (bao gồm cả những phát hiện sai). Trong thử nghiệm của chúng tôi, hệ thống đạt được **Precision** là **73.00%**. Điều này cho thấy rằng, trong số các đối tượng mà hệ thống đã phát hiện, có khoảng 73% là chính xác.

3.3.2. Recall:

Recall là thước đo mức độ đầy đủ của các đối tượng đã được phát hiện trong vùng quan sát. Nó tính tỷ lệ giữa số đối tượng phát hiện đúng (True Positives) với tổng số đối tượng thực tế cần phát hiện (bao gồm cả các đối tượng bị bỏ sót). Trong kết quả này, **Recall** của hệ thống là **12.55%**, tức là hệ thống chỉ phát hiện được khoảng 12.55% các đối tượng thực tế có trong video. Đây là một kết quả tương đối thấp, cho thấy hệ thống bỏ sót nhiều đối tượng.

3.3.3. F1 Score:

F1 Score là trung bình điều hòa của Precision và Recall, giúp cân bằng giữa hai thước đo này. Với **F1 Score** đạt **21.41%**, hệ thống của chúng tôi hiện chưa đạt hiệu suất tối ưu do độ bao phủ còn thấp. F1 Score thấp chủ yếu do Recall thấp, cho thấy mô hình cần cải thiện khả năng phát hiện để tăng khả năng bao phủ đối tượng trong khung hình.

3.3.4. Kết quả

Kết quả này cho thấy rằng hệ thống hiện tại có khả năng xác định chính xác đối tượng đã phát hiện (Precision cao) nhưng vẫn còn bỏ sót nhiều đối tượng (Recall thấp). Trong các ứng dụng thực tế, để nâng cao hiệu quả của hệ thống, chúng tôi có thể điều chỉnh thuật toán, giảm bớt hiện tượng bỏ sót các đối tượng trong khung hình, và tối ưu hóa các tham số như kích thước kernel hoặc các ngưỡng lọc nhiễu.

CHƯƠNG 4. HIỆN THỰC KẾT QUẢ

4.1. Yêu cầu hệ thống và tập dữ liệu thực nghiệm

Yêu cầu Hệ thống:

CPU: Bộ xử lý đa nhân (multi-core) có xung nhịp cao để xử lý khung hình nhanh

GPU (tùy chọn): Đối với các hệ thống yêu cầu xử lý video thời gian thực hoặc số lượng video lớn, GPU như NVIDIA (CUDA hỗ trợ tốt cho OpenCV) sẽ tăng tốc độ

RAM: Để lưu trữ các mô hình Gaussian của mỗi điểm ảnh cho từng khung hình,

tối

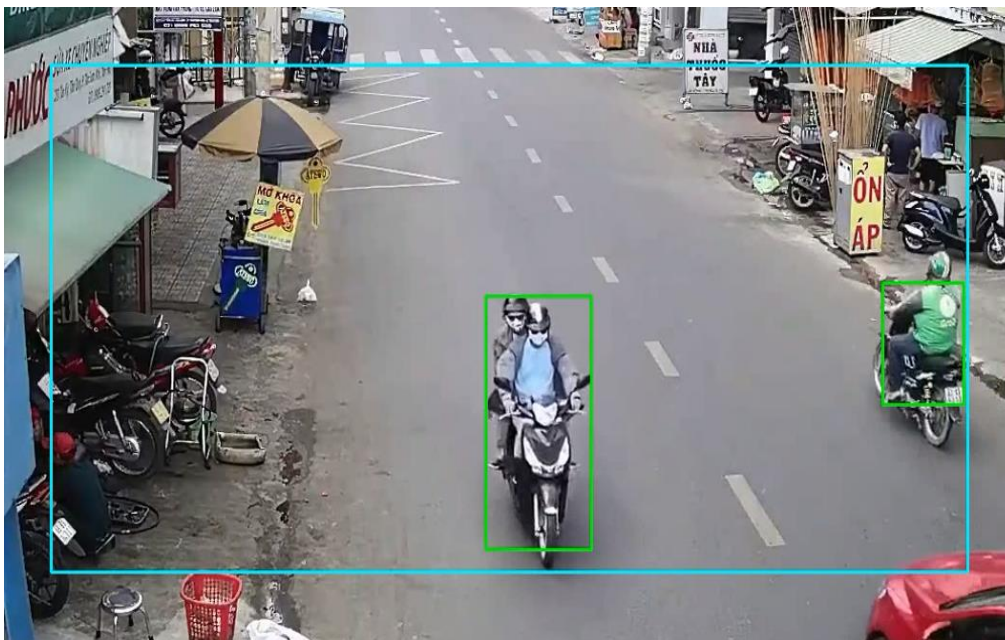
tối thiểu 8GB RAM là cần thiết

Thư viện phần mềm: Opencv, Numpy

Tập Dữ Liệu Thực Nghiệm:

Video ngắn: Dùng để thử nghiệm và kiểm tra nhanh độ chính xác của thuật toán

4.2. Kết quả thực nghiệm



CHƯƠNG 5. KẾT LUẬN

5.1. Kết quả đạt được

Qua đề án trên cũng như trải nghiệm với giải thuật Trừ Nền (Background Subtraction) thì chúng em rút ra được nhiều bài học về việc ứng dụng các thuật toán vào trong đời sống hơn và các ứng dụng quan sát cũng như thiết bị giám sát thì đằng sau đó là cả những giải thuật phức tạp và khó nhằn. Qua bài đề án này em thấy chúng em còn thiếu kinh nghiệm và cũng như kiến thức cần bổ sung nhiều hơn để có thể tối ưu hóa hết khả năng của giải thuật Trừ Nền và phát triển nó hơn vào ứng dụng đời sống

5.2. Ưu và nhược điểm

5.2.1. Ưu điểm của phương pháp MOG2:

- **Hiệu quả trong việc phát hiện đối tượng di chuyển:**

Sử dụng kỹ thuật trừ nền (background subtraction) giúp nhanh chóng xác định các đối tượng di chuyển, phù hợp cho các ứng dụng giám sát hay theo dõi.

- **Xử lý nhiễu tốt:**

Các bước xử lý nhiễu bao gồm co ảnh (erosion), giãn ảnh (dilation), Gaussian blur, và phép đóng (morphology closing) giúp giảm nhiễu, giữ lại các đối tượng có kích thước lớn và loại bỏ các đốm nhỏ hoặc nhiễu không cần thiết.

- **Nhận diện và lọc các đối tượng theo vùng quan tâm:**

Kiểm tra xem đối tượng có nằm trong vùng quan tâm được xác định bởi hình chữ nhật, phù hợp cho các tình huống cần giám sát một khu vực cụ thể (như cửa ra vào).

- **Dễ điều chỉnh ngưỡng kích thước:**

Loại bỏ các đối tượng có diện tích nhỏ dưới ngưỡng nhất định, giúp giảm thiểu các trường hợp phát hiện sai.

5.2.2. Nhược điểm của phương pháp này:

- **Không linh hoạt với môi trường thay đổi:**

Phương pháp trừ nền nhạy cảm với sự thay đổi của nền như ánh sáng, thời tiết, hoặc các yếu tố môi trường thay đổi, dẫn đến việc phát hiện sai.

- **Hiệu quả kém khi có nhiều đối tượng và nhiễu phức tạp:**

Khi có nhiều đối tượng hoặc khi nền phức tạp, các bước khử nhiễu và trích xuất đối tượng có thể gây mất chi tiết hoặc bỏ sót một số đối tượng.

- **Khả năng nhận diện hạn chế:**

Phương pháp này chỉ phát hiện các đối tượng chuyển động và không phân biệt được từng loại đối tượng cụ thể. Nếu cần phân loại (như phân biệt người và vật thể khác), cần thêm mô hình nhận diện phức tạp hơn.

- **Yêu cầu điều chỉnh tham số cho từng môi trường:**

Các tham số như ngưỡng nhị phân, kích thước kernel cần được điều chỉnh lại để phù hợp với từng bối cảnh khác nhau, dẫn đến khó khăn khi áp dụng trong các môi trường đa dạng.

- **Tốc độ xử lý và hiệu năng:**

Với số lượng lớn khung hình, các bước khử nhiễu và trích xuất đối tượng có thể tốn nhiều tài nguyên xử lý và làm giảm hiệu năng hệ thống nếu không được tối ưu tốt.

5.3. Hướng mở rộng tương lai

Nói về các ưu điểm của MOG2 thì chúng ta có thể thấy phương pháp này có thể xử lý nhanh các khung hình chuyển động không quá phức tạp nhưng việc xử lý khung hình có tốc độ chuyển động cao cũng như muốn xác định các vật thể đó là gì một cách chính xác thì nhóm em đang phát triển giải thuật Trừ Nền nhưng với phương pháp khác có thể kể đến như :

- Kết hợp trừ nền với mạng Neural nhân tạo như YOLO (You Only Look Once), SSD (Single Shot MultiBox Detector) hoặc Faster R-CNN.
- Kết hợp trừ nền với Phân khúc ảnh (Semantic Segmentation) như Mask R-CNN hoặc DeepLab
- ...

CHƯƠNG 5. KẾT LUẬN

Chúng em đang dự định trong tương lai sẽ sử dụng một trong những phương pháp trên để ứng dụng vào mô hình Camera giám sát để có thể phát hiện chính xác đối tượng di chuyển trong khung hình để người dùng có thể dễ dàng giám sát những người di chuyển xung quanh khu vực cần giám sát.

PHỤ LỤC CODE

Hình 1: Tải thư viện 1.....	12
Hình 2: Khởi tạo đọc video và gọi biến bằng MOG2 1	13
Hình 3: Thiết lập thông số cho việc ghi video 1	13
Hình 4: Khởi tạo đường dẫn OutPut 1.....	13
Hình 5: Đặt tọa độ khung hình 1	14
Hình 6: Sử dụng phương pháp trừ nền 1	15
Hình 7: Tìm, xác định và vẽ đối tượng 1.....	16
Hình 8: In ra kết quả và giải phòng TN 1	17
Hình 9 : Khai báo các biến đánh giá 1	17

```
import cv2
import numpy as np
```

Hình 1: Tải thư viện 1

```
# Tạo đối tượng trừ nền (sử dụng MOG2) để tách nền khỏi đối tượng chuyển động
backSub = cv2.createBackgroundSubtractorMOG2()

# Đọc video đầu vào từ đường dẫn
cap = cv2.VideoCapture('/content/drive/MyDrive/Colab
Notebooks/video/video_mog2.mp4')
```

Hình 2: Khởi tạo đọc video và gọi biến bằng MOG2 1

```
fourcc = cv2.VideoWriter_fourcc(*'mp4v') # Định dạng mã video codec cho file mp4
frame_width = int(cap.get(3)) # Độ rộng của khung hình video
frame_height = int(cap.get(4)) # Chiều cao của khung hình video
```

Hình 3: Thiết lập thông số cho việc ghi video 1

```
# Khởi tạo đối tượng ghi video chính thức sau khi xử lý
out = cv2.VideoWriter('output2.mp4', fourcc, 20.0, (frame_width,
frame_height))

# Khởi tạo đối tượng ghi video lưu ảnh sau khi trừ nền nhưng chưa khử nhiễu
out_no_denoise = cv2.VideoWriter('output_no_denoise.mp4', fourcc, 20.0,
(frame_width, frame_height))

# Khởi tạo đối tượng ghi video lưu ảnh sau khi trừ nền và đã khử nhiễu
out_noise = cv2.VideoWriter('output_noise.mp4', fourcc, 20.0,
(frame_width, frame_height))
```

Hình 4: Khởi tạo đường dẫn OutPut 1

```
# Đặt tọa độ cho hình chữ nhật ở giữa khung hình
```

CHƯƠNG 5. KẾT LUẬN

```
center_x = frame_width // 2 # Tọa độ x của trung tâm khung hình
center_y = frame_height // 2 # Tọa độ y của trung tâm khung hình
top_left = (center_x - frame_width // 4, center_y - frame_height // 4) #trừ đi một phần tư độ rộng
(frame_width // 4) để dịch chuyển sang bên trái.
, center_y - frame_height // 4) #trừ đi một phần tư chiều cao
(frame_height // 4) để dịch chuyển lên trên.
bottom_right = (center_x + frame_width // 4, center_y + frame_height //
4) # Góc dưới bên phải của hình chữ nhật
```

Hình 5: Đặt tọa độ khung hình 1

```
total_detected_objects = 0 # Khởi tạo biến đếm số lượng đối tượng phát
hiện trong khung hình
while True:
    ret, frame = cap.read() # Đọc từng khung hình của video, trả về 'ret'
    (True nếu đọc được, False nếu không) và 'frame' là khung hình
    if not ret:
        break # Nếu không còn khung hình nào (ret=False), dừng vòng lặp

    # Áp dụng bộ trừ nền để tạo mặt nạ nền cho khung hình hiện tại
    fgMask = backSub.apply(frame)
    fgMask_color = cv2.cvtColor(fgMask, cv2.COLOR_GRAY2BGR) # Chuyển ảnh
    mặt nạ sang ảnh màu (định dạng BGR) để có thể ghi vào video

    # Ghi khung hình sau khi trừ nền nhưng chưa khử nhiễu
    out_no_denoise.write(fgMask_color)

    # Khử nhiễu trên mặt nạ bằng các phép toán xử lý ảnh
    kernel = np.ones((5, 5), np.uint8) # Tạo kernel 5x5 với tất cả phần
    tử là 1, loại uint8 (để thực hiện các phép toán giãn và co)
    fgMask = cv2.erode(fgMask, kernel, iterations=1) # Phép co ảnh để
    loại bỏ các đốm nhiễu nhỏ; 1 lần lặp lại (iterations=1)
    fgMask = cv2.dilate(fgMask, kernel, iterations=1) # Phép giãn ảnh để
    phục hồi vùng bị mất do phép co; 1 lần lặp lại (iterations=1)
    fgMask = cv2.GaussianBlur(fgMask, (3, 3), 0) # Làm mượt bằng bộ lọc
    Gaussian với kích thước 3x3 và độ lệch chuẩn 0
    fgMask = cv2.morphologyEx(fgMask, cv2.MORPH_CLOSE, kernel) # Đóng
    (close) các lỗ nhỏ trong mặt nạ, sử dụng kernel 5x5
    _, fgMask = cv2.threshold(fgMask, 130, 255, cv2.THRESH_BINARY) #
    Ngưỡng hóa để chuyển thành ảnh nhị phân: giá trị > 130 thành 255 (trắng),
    còn lại thành 0 (đen)
```

CHƯƠNG 5. KẾT LUẬN

```
# Chuyển đổi mặt nạ đã khử nhiễu thành ảnh màu để ghi vào video
fgMask_color_denoised = cv2.cvtColor(fgMask, cv2.COLOR_GRAY2BGR)

# Ghi lại khung hình sau khi trừ nền và khử nhiễu
out_noise.write(fgMask_color_denoised)
```

Hình 6: Sử dụng phương pháp trừ nền 1

```
# Tìm đường viền của các đối tượng trong mặt nạ
fsMask = cv2.Canny(fgMask, 20, 200) # Sử dụng bộ lọc Canny để phát
hiện cạnh với ngưỡng dưới 20 và ngưỡng trên 200
contours, _ = cv2.findContours(fsMask, cv2.RETR_TREE,
cv2.CHAIN_APPROX_SIMPLE) # Tìm đường viền, sử dụng phương pháp RETR_TREE
để lấy tất cả các đường viền và CHAIN_APPROX_SIMPLE để giảm số điểm

# Vẽ hình chữ nhật phát hiện đối tượng trên khung hình gốc
cv2.rectangle(frame, top_left, bottom_right, (255, 255, 0), 2) # Vẽ
hình chữ nhật màu xanh dương nhạt (BGR: 255, 255, 0) với độ dày viền 2
pixel

# Đếm số lượng đối tượng trong khung hình hiện tại
detected_objects = 0
true_positives = 0
false_positives = 0
false_negatives = 0

for i in range(len(contours)): # Duyệt qua từng đường viền được phát
hiện
    (x, y, w, h) = cv2.boundingRect(contours[i]) # Tính tọa độ góc
trên trái (x, y) và kích thước (w: width, h: height) của hình chữ nhật bao
quanh đối tượng
    cx = x + w // 2 # Tọa độ x của tâm hình chữ nhật (vị trí x cộng
nửa chiều rộng w)
    cy = y + h // 2 # Tọa độ y của tâm hình chữ nhật (vị trí y cộng
nửa chiều cao h)

    # Kiểm tra xem đối tượng có nằm trong vùng hình chữ nhật phát hiện
không
    logic = top_left[0] < cx < bottom_right[0] and top_left[1] < cy <
bottom_right[1] # Điều kiện: tâm của đối tượng phải nằm trong tọa độ vùng
phát hiện
```

CHƯƠNG 5. KẾT LUẬN

```
area = cv2.contourArea(contours[i]) # Diện tích của đường viền

if area < 500: # Bỏ qua các đối tượng có diện tích nhỏ hơn 500
pixel (được coi là nhiễu hoặc không đủ lớn để tính là một đối tượng)
    continue
if logic:
    true_positives += 1
    total_true_positives += 1
    cv2.rectangle(frame, (x, y), (x + w, y + h), (0, 255, 0), 2) #
Vẽ hình chữ nhật màu xanh lá (BGR: 0, 255, 0) với độ dày viền 2 pixel
quanh đối tượng phát hiện
    cv2.putText(frame, 'People', (500, 100),
cv2.FONT_HERSHEY_SIMPLEX, 2, (0, 0, 255), 2) # Thêm nhãn 'People' tại vị
trí (500, 100) trên khung hình, màu đỏ, cỡ chữ 2

    detected_objects += 1 # Tăng biến đếm đối tượng phát hiện
trong khung hình

# Cộng số lượng đối tượng phát hiện trong khung hình vào tổng số đối
tượng ,
    total_true_positives += true_positives
    total_false_positives += false_positives
    total_false_negatives += len(contours) - true_positives -
false_positives

    total_detected_objects += detected_objects

# Ghi video chính thức sau khi xử nhiễu và phát hiện đối tượng
out.write(frame) # Ghi khung hình đã xử lý (có hình chữ nhật và nhãn)
vào video đầu ra
```

Hình 7: Tìm, xác định và vẽ đối tượng 1

```
# Sau khi xử lý video xong, tính các thang đo đánh giá
precision = total_true_positives / (total_true_positives +
total_false_positives) if (total_true_positives + total_false_positives) >
0 else 0
recall = total_true_positives / (total_true_positives +
total_false_negatives) if (total_true_positives + total_false_negatives) >
0 else 0
```

CHƯƠNG 5. KẾT LUẬN

```
f1_score = 2 * precision * recall / (precision + recall) if (precision +
recall) > 0 else 0

# In ra các thang đo đánh giá
print("Tổng số đối tượng được phát hiện:", total_detected_objects)
print("Precision:", precision)
print("Recall:", recall)
print("F1 Score:", f1_score)

# Giải phóng tài nguyên
cap.release()
out.release()
out_no_denoise.release()
out_noise.release()
```

Hình 8: In ra kết quả và giải phòng TN 1

```
total_detected_objects = 0
total_true_positives = 0
total_false_positives = 0
total_false_negatives = 0
```

Hình 9 : Khai báo các biến đánh giá 1

TÀI LIỆU THAM KHẢO

Tham khảo các trang YouTube về cách áp dụng giải thuật trừ nền và từ các buổi thực hành về phương pháp trừ nền.