

ĐẠI HỌC QUỐC GIA HÀ NỘI
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ



BÁO CÁO BÀI TẬP LỚN
MÔN HỌC: XỬ LÝ ẢNH VÀ THỊ GIÁC ROBOT
ĐỀ TÀI: XÁC ĐỊNH VỊ TRÍ TÂM VẬT TRÊN MẶT PHẪNG

GVHD : PGS. TS. Hoàng Văn Xiêm

KS. Đỗ Ngọc Minh

Lớp : RBE3015 1

Họ tên : Bùi Đình Đăng - 21020899

HÀ NỘI - 2023

MỤC LỤC

1. Giới thiệu tổng quan đề tài.....	2
2. Một số yêu cầu về bài toán.....	3
2.1. Thiết lập mô hình hệ thống	3
2.1 Thiết lập các điều kiện môi trường của bài toán.....	3
3. Mô phỏng hệ thống thuật toán	4
3.1 Sơ đồ thuật toán của hệ thống	4
3.2 Các bước thực hiện	4
3.2.1 Tách hệ quy chiếu	4
3.2.2 Xác định vị trí tâm vật.....	5
4. Kết quả và đánh giá.....	6
5. Kết luận và định hướng phát triển.....	7
5.1 Kết luận	7
5.2 Định hướng phát triển	7

1. Giới thiệu tổng quan đề tài

Bài toán xác định tâm vật thể 2D trên mặt phẳng đóng vai trò quan trọng trong công nghiệp, đặc biệt trong lĩnh vực tự động hóa và kiểm soát chất lượng sản phẩm. Việc xác định tâm vật thể 2D giúp trong việc định vị và theo dõi vật thể trong thời gian thực, từ đó tạo ra các ứng dụng quan trọng như hệ thống định vị hàng tự động trong dây chuyền sản xuất, kiểm tra và phân loại sản phẩm, hoặc thậm chí trong ứng dụng quan trọng như hệ thống an ninh và giám sát.

Trong sản xuất, việc xác định tâm vật thể 2D có thể được áp dụng trong việc kiểm tra vị trí của các linh kiện đang trong quá trình sản xuất hoặc kiểm tra độ chính xác của sản phẩm cuối cùng trước khi đóng gói và xuất xưởng. Tính tự động hóa càng cao và khả năng xác định nhanh chóng tâm vật thể 2D sẽ giúp tối ưu hóa hiệu suất sản xuất và giảm thực trong quá trình sản xuất hàng loạt.

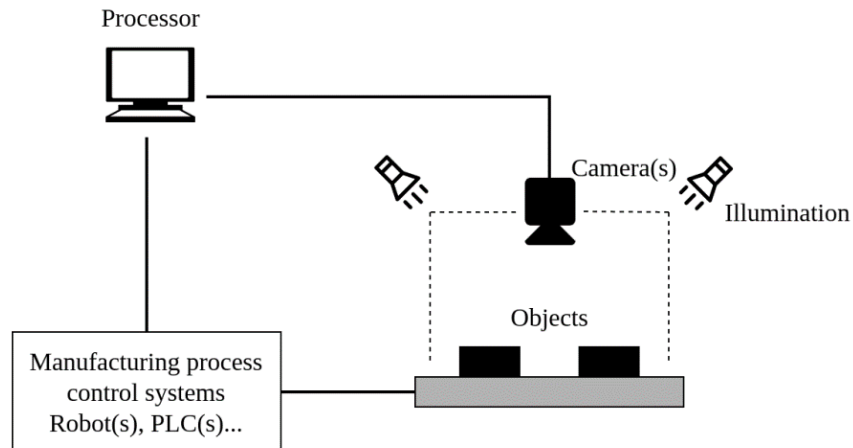
Ngoài ra, việc xác định tâm vật thể 2D cũng có thể được áp dụng trong lập trình robot công nghiệp để thực hiện các nhiệm vụ như lắp ráp tự động, vận chuyển và sắp xếp hàng hóa hoặc kiểm tra chất lượng sản phẩm. Điều này nhằm mục tiêu tăng cường hiệu quả và đảm bảo tính chính xác trong các ứng dụng công nghiệp lớn.

Khi thực hiện bài toán xác định tâm vật thể 2D trên mặt phẳng, mục tiêu của em là phát triển kỹ năng phân tích và xử lý hình ảnh, nắm vững các thuật toán xử lý ảnh và thực hành kỹ năng lập trình trong môi trường thị giác máy tính. Bài toán này cũng giúp em rèn luyện khả năng tư duy logic và giải quyết vấn đề trong lĩnh vực khoa học máy tính. Ngoài ra, từ bài tập này, em có thể hiểu rõ hơn về ứng dụng của xử lý ảnh trong thực tế, từ sản xuất đến an ninh và giám sát.

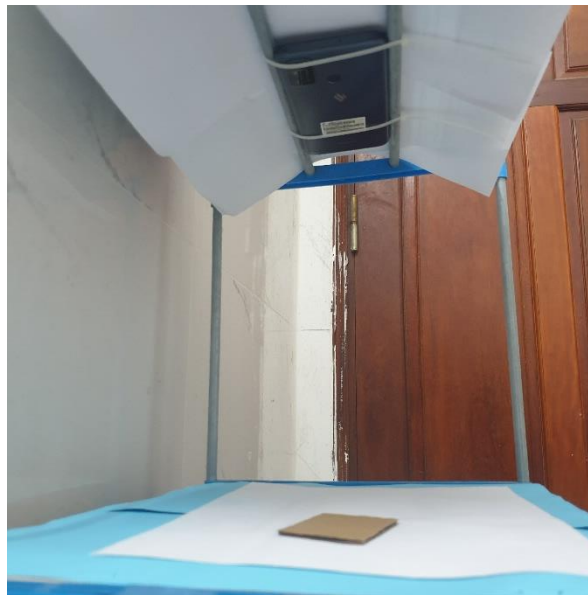
2. Một số yêu cầu về bài toán

2.1. Thiết lập mô hình hệ thống

Mô hình thực nghiệm sẽ được thiết kế với một Camera cố định được đặt phía trên và bên dưới đặt một mặt phẳng để làm hệ quy chiếu (chẳng hạn là đặt tờ giấy A4 làm hệ quy chiếu), sau đó đặt vật mình cần xác định tâm như hình minh họa sau.



Hình 1. Mô hình tổng quát hệ thống.



Hình 2. Mô hình thực tế của hệ thống.

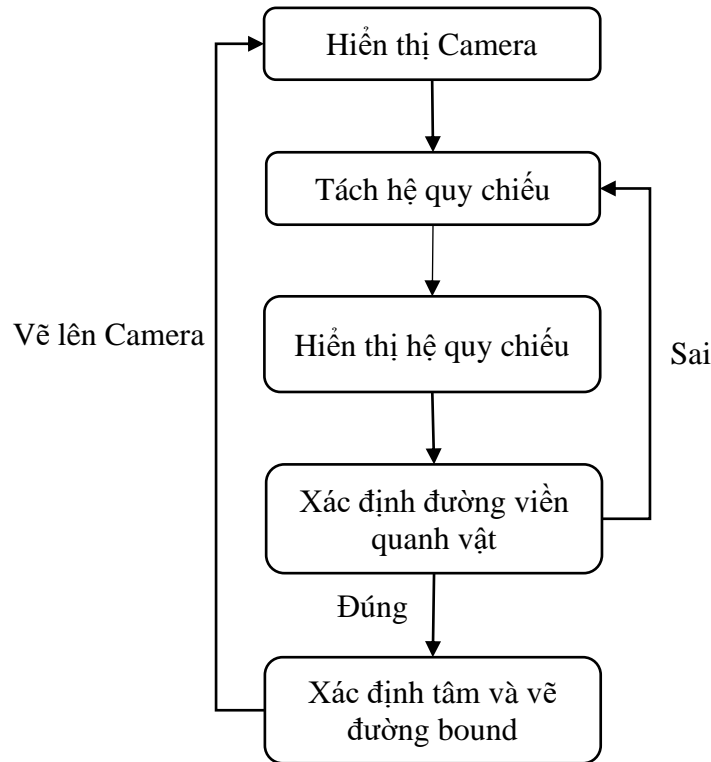
2.1 Thiết lập các điều kiện môi trường của bài toán

Hệ thống được lắp đặt đã thỏa mãn các điều kiện sau:

- + Vị trí đặt Camera song song với mặt phẳng đặt vật.
- + Độ sáng môi trường hợp lý để tránh hiện tượng ánh sáng quá yếu (không xác định được vật) hoặc ánh sáng quá mạnh (tạo ra vệt bóng dài khiến khó xác định vật).

3. Mô phỏng hệ thống thuật toán

3.1 Sơ đồ thuật toán của hệ thống



Hình 3. Sơ đồ thuật toán

3.2 Các bước thực hiện

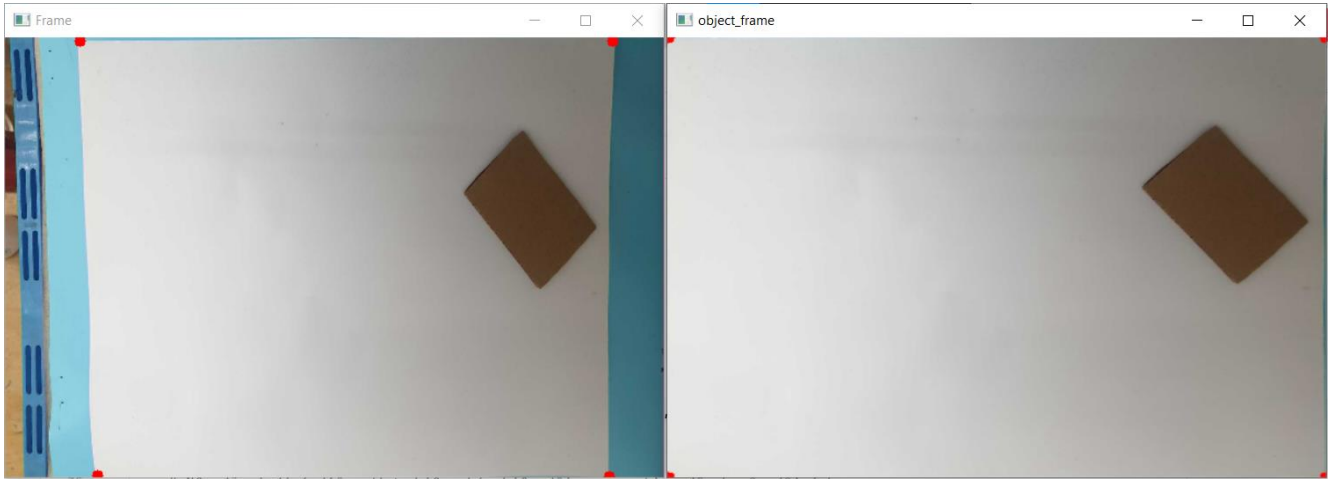
3.2.1 Tách hệ quy chiếu

Do dễ hạn chế ảnh hưởng của các vật không mong muốn trong việc xác định hệ quy chiếu, em đã sử dụng hàm `cv2.EVENT_LBUTTONDOWN` để gọi ra 4 điểm góc của hệ quy chiếu (ở trường hợp này là tờ A4 có kích thước 29.7cm x 20cm) bằng cách click chuột trái.

```
5 # Hàm callback để vẽ các điểm trên màn hình khi người dùng nhấp chuột
6 def mouse_drawing(event, x, y):
7     if event == cv2.EVENT_LBUTTONDOWN:
8         circles.append((x, y))
9 # Mở video stream từ camera IP
10 video = cv2.VideoCapture('http://192.168.2.7:8080/video')#http://192.168.2.7:8080/: đây là IP của Camera
11 # Tạo cửa sổ để hiển thị video và gắn callback để vẽ các điểm khi chuột được nhấp
12 cv2.namedWindow("Frame")
13 cv2.setMouseCallback("Frame", mouse_drawing)
```

Hình 4. Hàm gọi ra 4 điểm góc bằng click chuột

Sau đó, em tính toán và chuyển đổi 4 đỉnh của hệ quy chiếu để tách hệ này sang một frame mới – gọi là `object_frame` có kích thước 600*400 theo góc nhìn trực tiếp của mình với 4 điểm đó là 4 điểm được xác định bằng cách click chuột trái vào 4 góc của tờ A4 với thứ tự góc trên bên trái, góc trên bên phải, góc dưới bên trái và góc dưới bên phải.



Hình 5. Hệ quy chiếu được tách ra.

3.2.2 Xác định vị trí tâm vật

Trước khi xác định vị trí tâm vật, hệ quy chiếu được tách ra sẽ đi qua các bộ lọc ảnh để giảm bớt nhiễu.

```

45     ## Chuyển ảnh sang ảnh grayscale và làm mờ để chuẩn bị cho việc xử lý contour
46     gray = cv2.cvtColor(object_frame, cv2.COLOR_BGR2GRAY)
47     gray = cv2.GaussianBlur(gray, ksize: (5, 5), sigmaX: 0)
48     gray = cv2.medianBlur(gray, ksize: 3)
49
50     # Áp dụng ngưỡng để tạo ảnh nhị phân
51     ret, thresh = cv2.threshold(gray, thresh: 150, maxval: 255, cv2.THRESH_BINARY)
52     thresh = cv2.morphologyEx(thresh, cv2.MORPH_GRADIENT, kernel)
53     thresh = cv2.dilate(thresh, kernel, iterations=5)

```

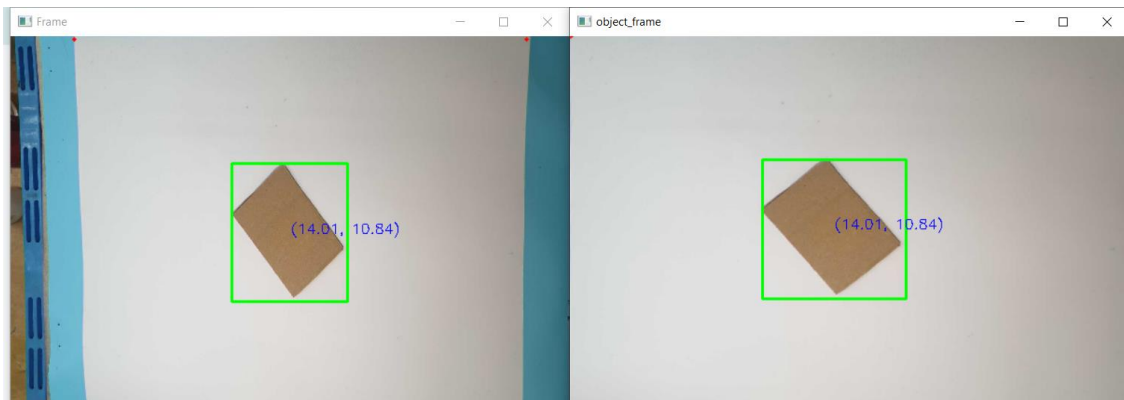
Hình 5. Các hàm xử lý nhiễu

Sau đó, vật sẽ được xác định các cạnh viền bằng hàm `cv2.findContours` và ta sẽ xác định được kích thước của đường viền lớn nhất sẽ bao quanh vật với các giá trị thu được là x , y , w , h với:

- + (x, y) là tọa độ điểm trên bên trái của đường viền
- + w là độ rộng của đường viền
- + h là độ cao đường viền

Khi đó, tọa độ tâm của vật trên ảnh là $\text{center} = (x + \frac{w}{2}, y + \frac{h}{2})$

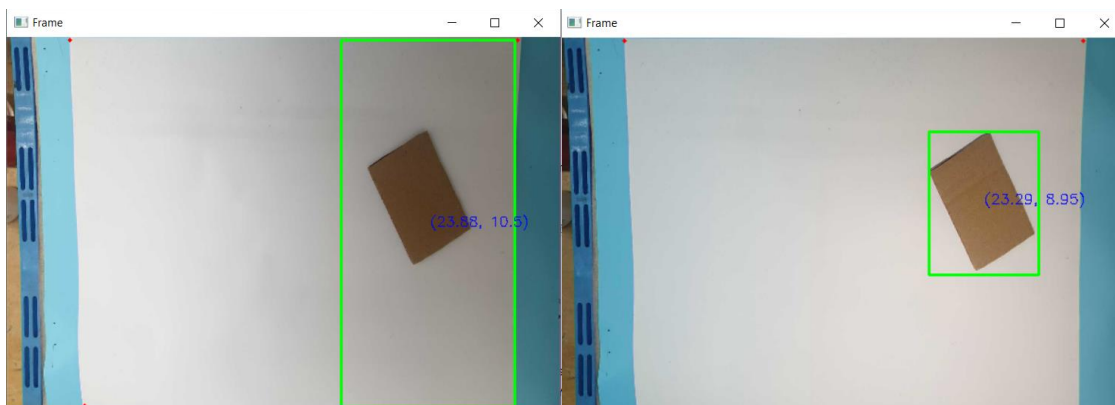
Tiếp theo bằng các thiết lập tỉ lệ ảnh và độ dài thực tế ta tính được vị trí thực tế của vật thể hệ quy chiếu, sau đó in vị trí chính xác của vật trên `object_frame` và `frame` như hình vẽ.



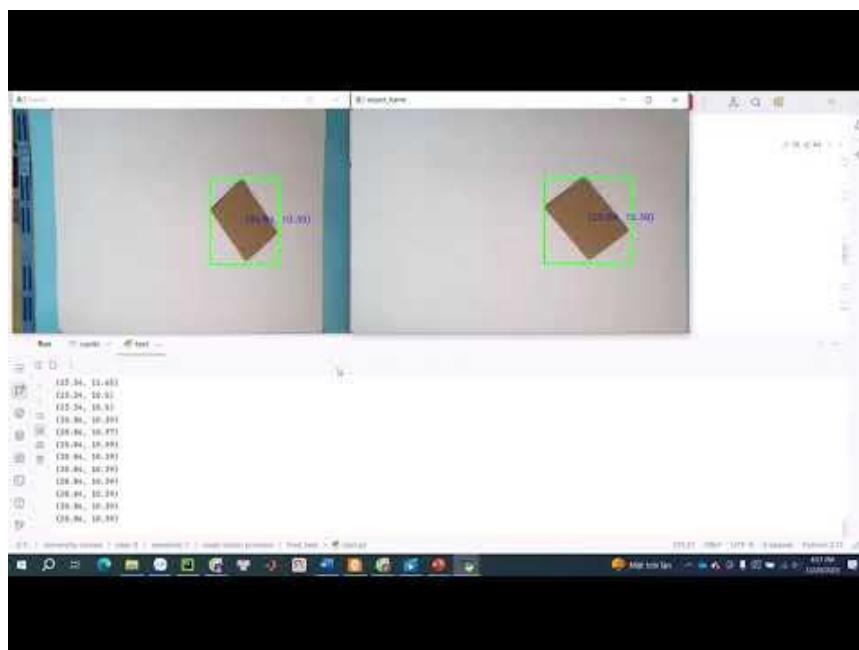
Hình 6. Tọa độ thực tế của vật

4. Kết quả và đánh giá

Trên thực tế, trong lúc thử nghiệm, nếu ảnh sáng cung cấp không đủ sẽ làm cho hệ thống bị tối và không xác định được đường viền như hình dưới đây.



Hình 7. Thử nghiệm nếu cung cấp đủ sáng và không đủ sáng



Hình 8. Video thực nghiệm

5. Kết luận và định hướng phát triển

5.1 Kết luận

Từ những nỗ lực và kết quả thu được trong quá trình thực hiện bài toán xác định tâm vật thể 2D trên mặt phẳng, đây là một bước đầu tiên quan trọng trong việc áp dụng các kiến thức về xử lý ảnh và thuật toán vào thực tế. Hệ thống đã đạt được khả năng nhận diện và xác định tâm vật thể 2D một cách tương đối chính xác dựa trên mô hình camera cố định và các điều kiện môi trường đã được thiết lập.

Tuy nhiên, còn nhiều hướng phát triển cần được nghiên cứu và cải tiến hơn. Vấn đề về ánh sáng môi trường vẫn là một thách thức lớn, cần phải tối ưu hóa và cải thiện để đảm bảo tính ổn định và độ chính xác của hệ thống trong các điều kiện ánh sáng khác nhau. Ngoài ra, việc tăng cường khả năng phát hiện vật thể trong các trường hợp đặc biệt như khi có sự che khuất, đa vật thể hoặc vật thể nhỏ cũng là những điểm cần được tiếp tục nghiên cứu và phát triển.

Với những hướng phát triển này, bài toán xác định tâm vật thể 2D trên mặt phẳng sẽ trở nên có ứng dụng rộng rãi hơn trong lĩnh vực tự động hóa, kiểm soát chất lượng, an ninh và giám sát. Đồng thời, việc nghiên cứu và phát triển các thuật toán xử lý ảnh cũng sẽ góp phần quan trọng vào sự phát triển của ngành Thị giác Robot.

5.2 Định hướng phát triển

Từ vấn đề đặt ra ở trên, em nhận thấy có những hướng phát triển thêm cho sản phẩm như:

- + Tối ưu hóa hiệu suất: Đảm bảo rằng mã đang thực hiện các phép toán một cách hiệu quả, đặc biệt khi xử lý video trực tiếp. Sử dụng các kỹ thuật tối ưu hóa như caching, việc tái sử dụng biến, và sử dụng thư viện tối ưu hóa để cải thiện hiệu suất.

- + Xử lý đa vật thể: Mở rộng mã để có thể phát hiện và theo dõi nhiều vật thể khác nhau trong cùng một khung hình.

- + Xử lý chuyển động: Thêm chức năng xử lý chuyển động để theo dõi sự chuyển động của vật thể qua các khung hình, cung cấp thông tin về vận tốc và hướng di chuyển.

- + Tích hợp với hệ thống điều khiển hoặc IoT: Nếu đang phát triển một ứng dụng thực tế, bạn có thể tích hợp mã của mình với hệ thống điều khiển hoặc IoT để có thể thực hiện hành động dựa trên dữ liệu từ việc phân tích hình ảnh.

Các đề xuất trên đây có thể giúp em phát triển hệ thống của mình thành một ứng dụng hoặc hệ thống phức tạp hơn.