

ỨNG DỤNG XỬ LÝ ẢNH CHO TAY MÁY NHẬN DẠNG VÀ GẮP VẬT 3D DI CHUYỂN TRÊN BĂNG CHUYỀN

IMAGE PROCESSING APPLICATION FOR ROBOTIC ARM TO RECOGNIZE AND PICK UP 3D OBJECTS MOVING ON CONVEYOR BELT

Lê Đức Hạnh

Khoa Cơ khí, Trường Đại học Bách Khoa, Đại học Quốc gia Thành phố Hồ Chí Minh

Email: ldhanh@hcmut.edu.vn

TÓM TẮT

Ứng dụng xử lý ảnh để nhận dạng vị trí của vật thể đang di chuyển trên băng chuyền và điều khiển tay máy để gắp vật ra để lắp ráp là một trong những ứng dụng rất phổ biến trong công nghiệp hiện nay. Tuy nhiên hiện nay, hầu hết các ứng dụng đều cho các vật có hình dạng phẳng 2D. Các vật có hình dáng 3D vẫn chưa được nghiên cứu nhiều, nhất là ở Việt Nam do sự phức tạp trong tính toán và thời gian thực hiện lâu bằng phương pháp truyền thống. Nghiên cứu này trình bày phương pháp có thể tăng năng suất cho một dây chuyền lắp ráp bằng cách nhận dạng và gắp vật 3D đang chuyển động trên băng chuyền sử dụng xử lý hình ảnh và tay máy. Hệ thống bao gồm một camera đặt trên cao để phát hiện vật cần lắp ráp bất kỳ hướng và vị trí trên băng chuyền. Sau đó tọa độ 3D và hướng góc của vật trong tọa độ 3D được tính toán. Các dữ liệu tính toán sẽ được truyền xuống cho tay máy để thực thi tác vụ gắp. Quá trình lắp ráp sẽ được thực hiện dựa trên vị trí và hướng của vật đã biết trước để kiểm tra sai số. Qua thực nghiệm kiểm chứng, độ chính xác cũng như thời gian nhận dạng, hệ thống chứng minh là có khả năng áp dụng vào môi trường công nghiệp lắp ráp giúp tăng năng suất cho hệ thống.

Từ khóa: Thị giác máy tính; Lắp ráp; Tay máy; Điều khiển; Xử lý ảnh 3D.

ABSTRACT

The application of image processing to identify the position of moving objects on the conveyor belt and control the manipulator to pick up the object for assembly is one of the very popular applications in industry today. However, most of the applications are for 2D flat objects. 3D objects have not been studied much, especially in Vietnam due to the complexity of calculation and long execution time using traditional methods. This study presents a method that can increase the productivity of an assembly line by identifying and picking up 3D objects moving on the conveyor belt using image processing and manipulators. The system includes a camera placed above to detect the object to be assembled in any direction and position on the conveyor belt. Then the 3D coordinates and angular orientation of the object in the 3D coordinates are calculated. The calculated data will be transmitted to the manipulator to perform the picking task. The assembly process will be performed based on the known position and orientation of the object to check for errors. Through experimental verification, the accuracy as well as the recognition time, the system proves to be applicable to the assembly industrial environment to help increase the productivity of the system.

Keywords: Computer vision; Assembly; Manipulator; Control; Image processing.

1. GIỚI THIỆU

Yêu cầu tăng năng suất và hiệu quả trong công nghiệp hiện đại, đặc biệt là trong dây chuyền lắp ráp đòi hỏi phải áp dụng các công nghệ khác cho cánh tay robot. Hiện nay, hầu hết công việc lắp ráp đều được thực hiện thủ công. Công việc lắp ráp dựa trên thị giác máy tính đã được nghiên cứu cho nhiều hệ thống sản xuất tự động. Hiện nay, các nhà máy tự động hóa có sử dụng tay máy công nghiệp như ABB, Fanuc, Yaskawa, Denso, ở các cụm lắp ráp hay dây chuyền sản xuất, đóng gói các vật thường được di chuyển trên băng truyền. Sau đó, các vật này sẽ được robot gấp ra để đóng gói hay lắp ráp. Hệ thống chứa camera nhận dạng và Scara hoặc robot song song để gấp và đặt đồ vật. Nó đã được sử dụng rộng rãi trong nhiều ngành sản xuất công nghiệp [1-4]. Casado và cộng sự đã sử dụng nhận dạng bằng cách dùng mẫu 2D để nhận dạng và theo dõi đối tượng chuyển động [5]. Weiss và cộng sự đề xuất một phương pháp áp dụng các quan hệ đại số giữa các bất biến của mô hình 3D và các bất biến của hình ảnh 2D của nó dưới phép chiếu xạ tổng quát để ước tính các tư thế 3D của đối tượng [6].

Tuy nhiên, do bị che lấp hay chồng lấn, nên việc ước tính tư thế sử dụng hình ảnh 2D chỉ hoạt động khi tư thế của chúng bị giới hạn trong một số trường hợp. Để khắc phục các vấn đề này, mô hình cad 3D dựa trên nhận dạng đối tượng được phát triển. Vũ và cộng sự [7] và Choi và cộng sự [8] xây dựng mô hình cad của đối tượng và sau đó sử dụng phương pháp sơ đồ biểu quyết để ước tính tư thế 6DOF và nhận dạng đối tượng của các đối tượng trong thùng. A.K. Jain và cộng sự [9] sử dụng các đặc trưng mô tả đối tượng 3D được tạo trên phần mềm CAD để tạo ra cả biểu đồ quan hệ góc nhìn để sử dụng trong nhận dạng đối tượng. R. Jain và cộng sự [10] đã sử dụng thông tin trong cơ sở

dữ liệu CAD để lựa chọn đặc trưng cho việc nhận dạng đối tượng. Menon và cộng sự [11] đã đề xuất thuật toán dựa trên tìm kiếm dựa trên mô hình 3D để chọn một vật thể đang chuyển động. Thuật toán dựa trên bỏ phiếu và tìm kiếm thường khó do thời gian tính toán lâu và tốn tài nguyên. Có thể thấy, việc xử lý hình ảnh 3D đã bước đầu được ứng dụng thành công. Tuy nhiên, các nghiên cứu trên chỉ áp dụng cho vật tĩnh vì thế ở các cụm lắp ráp, băng chuyền thường phải dừng lại để gấp các vật thể 3D phức tạp.

Bài báo này có sự khác biệt với các nghiên cứu trên. Nghiên cứu này sẽ kết hợp xử lý hình ảnh 3D và dựa vào đó tính toán vị trí cần gấp cho robot. Hệ thống gồm một camera Kinect đặt cố định trên cao dùng để tính toán tọa độ và hướng của vật. Sau đó, dựa vào vận tốc của băng tải, tọa độ điểm cần gấp sẽ được tính toán và truyền về robot để thực thi tác vụ gấp.

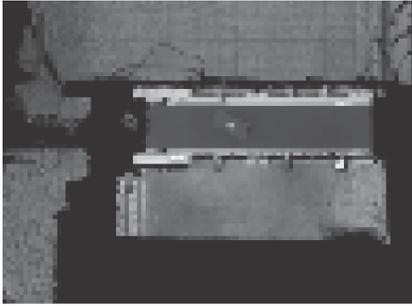
2. NHẬN DẠNG VẬT 3D

Mục đích của quá trình xử lý ảnh là để xác định tâm, hướng và góc xoay của vật trong hệ tọa độ robot. Để ước lượng tâm và vector pháp tuyến, trước tiên ta cần tách đối tượng xác định ra khỏi môi trường xung quanh, sử dụng các bộ lọc để loại bỏ nhiễu và giảm kích thước dữ liệu giúp tính toán nhanh hơn.

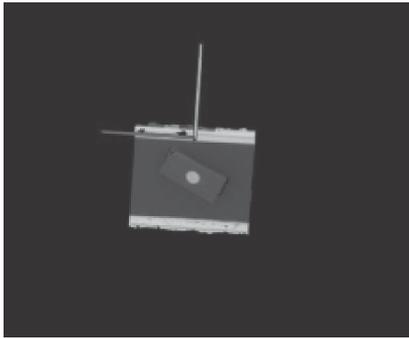
2.1. Tách vật

Hình thu được từ Kinect gồm nhiều vật trong khung hình như hình 1. Để loại bỏ những điểm không cần thiết trong cloud, ta sử dụng bộ lọc PassThrough. Bộ lọc này giúp cắt bỏ các điểm nằm ngoài vùng mong muốn theo 3 phương X, Y và Z như hình 2.





Hình 1. Hình ảnh thu được từ Kinect camera



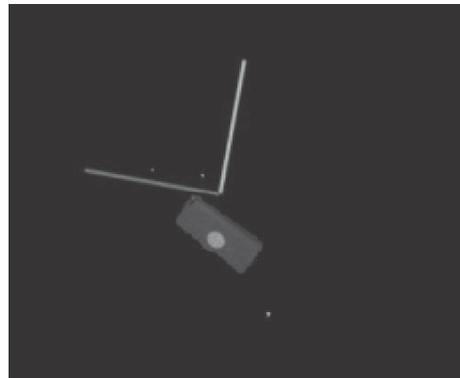
Hình 2. Hình sau khi lọc PassThrough

Sau khi giới hạn khoảng lấy dữ liệu theo 3 phương, vật thể vẫn lẫn lộn trong không gian xung quanh, cloud vẫn chứa dữ liệu điểm của các chi tiết như băng tải hay mặt bàn. Điều này làm cho việc ước lượng tâm và vector pháp tuyến của vật không chính xác. Cần phải tách vật ra khỏi môi trường xung quanh để quá trình tính toán được chính xác và nhanh hơn, ta thực hiện phân đoạn dữ liệu để xóa băng tải ra khỏi cloud. Việc phân đoạn dữ liệu được thực hiện bằng lớp SACSegmentation để tìm ra mặt phẳng chính của khung hình. Để thực hiện SACSegmentation, ta cần phải thiết lập dữ liệu đầu vào (dữ liệu thu được từ Kinect) và phương pháp phân đoạn ảnh (ở đây là RANSAC). Thuật toán RANSAC là thuật toán lặp với mục đích ước lượng các thông số của một mô hình toán học từ một bộ dữ liệu thu thập được bao gồm cả các điểm trong và ngoài. Các điểm trong là những điểm trong bộ dữ liệu phù hợp (hay nằm trong) mô hình toán học đó, còn điểm ngoài

là những điểm không nằm trong mô hình. Quá trình thực hiện của RANSAC như sau:

- Chọn một bộ nhỏ nhất các điểm bất kỳ từ dữ liệu đầu vào, giả thiết các điểm này đều là điểm trong.
- Tính toán các thông số của mô hình chứa các điểm đó.
- Kiểm tra tất cả các điểm còn lại trong dữ liệu đầu vào xem chúng nằm trong hay nằm ngoài mô hình dựa vào mô hình vừa tính được.
- Mô hình vừa xấp xỉ là tốt nếu số điểm trong thỏa mãn yêu cầu.
- Quá trình này được lặp lại để phát hiện các mô hình tốt hơn.

Sau khi thực hiện thuật toán RANSAC, ta sẽ có được dữ liệu chứa bàn, sau đó có thể xóa dữ liệu bàn khỏi Cloud và thu được dữ liệu chỉ chứa vật như hình 3.

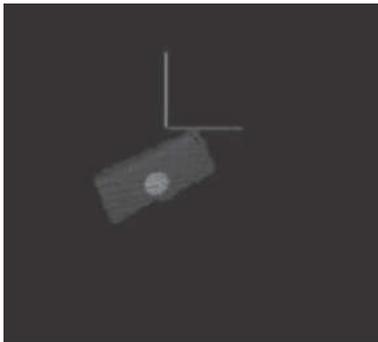


Hình 3. Hình sau khi dùng RANSAC

Khi sử dụng các thiết bị quét 3D như camera RGB-D hoặc máy quét laser, trong dữ liệu thu được thường xuất hiện những điểm nằm lơ lửng, không nằm trong mặt phẳng nào. Những điểm này thường xuất hiện gần nơi giao tiếp giữa các bề mặt, hay gần các cạnh của vật thể. Khi tính toán các đặc trưng của vật thể thì các điểm nhiễu này có thể gây ra sai lệch vì thực ra chúng không phải là một điểm dữ liệu

của vật thể mà chỉ là nhiễu tác động lên cảm biến. Hơn nữa, việc gia tăng số điểm trên đám mây điểm còn làm tăng thời gian tính toán. Do đó trước khi tính toán các đặc trưng, ta cần loại bỏ những điểm nhiễu không nằm trên các bề mặt này. Phương pháp được sử dụng phổ biến để loại bỏ các điểm nhiễu ngoài bề mặt này là StatalOutlierRemoval như hình 4. Đây là phương pháp tính khoảng cách trung bình từ một điểm đến các lân cận của nó để xác định các điểm không nằm trong bề mặt.

Sau khi phân đoạn và qua các bộ lọc thì chỉ còn lại vật và các điểm dữ liệu của vật nhưng mật độ điểm dữ liệu vẫn còn rất nhiều làm cho quá trình tính toán lâu và có khả năng tràn dữ liệu. Để tránh tình trạng đó, ta sử dụng bộ lọc VoxelGrid để giảm mẫu dữ liệu bằng cách đưa đám mây điểm vào một không gian gồm các lưới 3D, với kích thước của lưới lớn hơn so với khoảng cách giữa một điểm và điểm gần nó nhất. Nói cách khác, độ phân giải của Point Cloud của vật phải lớn hơn độ phân giải của lưới Voxel. Sau đó với mỗi lưới Voxel, giải thuật sẽ tính toán tâm trung bình của các điểm trên Point Cloud và thay thế các điểm đó bằng một điểm duy nhất tại trọng tâm trung bình.



Hình 4. Hình lọc StatalOutlierRemoval

2.2. Tính trọng tâm vật

Tâm của một cloud được tính bằng cách

lấy trung bình cộng tọa độ của tất cả các điểm có trong cloud đó.

$$\begin{aligned} X_{centroid} &= \frac{\sum_{i=0}^n x_i}{n} \\ Y_{centroid} &= \frac{\sum_{i=0}^n y_i}{n} \\ Z_{centroid} &= \frac{\sum_{i=0}^n z_i}{n} \end{aligned} \quad (1)$$

Trong đó: Với x_i, y_i, z_i là tọa độ điểm thứ i của cloud, n là tổng số điểm có trong cloud.

2.3. Tính vector pháp tuyến

Cách tính vector pháp tuyến được tóm tắt như sau: vector pháp tuyến của mặt phẳng tiếp tuyến với một lượng điểm lân cận của điểm đang xét. Để xác định một phương trình mặt phẳng trong không gian cần xác định một điểm x nằm trong mặt phẳng đó và vector pháp tuyến \vec{n} của nó. Điểm x được xác định bằng cách lấy trọng tâm các điểm lân cận của điểm đang xét:

$$x = \bar{p} = \frac{1}{k} \sum_{i=1}^k p_i \quad (2)$$

\vec{n} được xác định bằng các trị riêng của ma trận covariance C tính từ tọa độ của tập hợp các điểm lân cận:

$$C = \frac{1}{k} \sum_{i=1}^k \xi_i (p_i - \bar{p})(p_i - \bar{p})^T \quad (3)$$

Trong đó, ξ_i là hàm số tỉ trọng của điểm p_i , để tăng độ chính xác, các điểm càng gần tâm \bar{p} thì hàm số tỉ trọng ξ_i có giá trị càng lớn. Các trị riêng λ của ma trận covarian được tìm bằng cách giải nghiệm đa thức đặc trưng của nó:

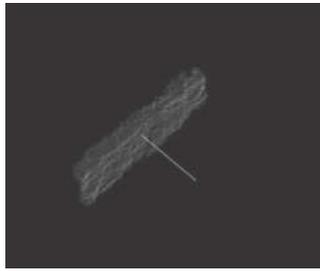
$$P(\lambda) = \det(C - \lambda I) = 0 \quad (4)$$

Các vector riêng được tính bằng công

thức ứng với các trị riêng được tìm bằng công thức:

$$C\vec{v}_j = \lambda_j \vec{v}_j, \quad j \in \{0,1,2\} \quad (5)$$

Nếu thỏa các điều kiện trên, các vector riêng sẽ tạo thành một khung trục giao. Khi đó có thể lấy vector pháp tuyến của điểm đang xét xấp xỉ vector riêng ứng với trị riêng lớn nhất hay $\vec{n} \approx \vec{v}_h$ (h thỏa điều kiện sao cho $\lambda_h = \max \{ \lambda_0, \lambda_1, \lambda_2 \}$).



Hình 5. Kết quả tính trọng tâm và vector pháp tuyến

3. TÍNH CHỈNH CAMERA

3.1. Chuyển hệ tọa độ

Tay máy có hệ tọa độ riêng, camera có hệ tọa độ riêng, do đó để robot có thể hiểu và gặp chính xác vị trí của vật được nhận diện từ camera, ta phải thực hiện tinh chỉnh. Quá trình tinh chỉnh là đi tìm ma trận chuyển đổi bao gồm ma trận xoay và tịnh tiến từ hệ tọa độ camera về hệ tọa độ robot, từ đó ta có thể điều khiển robot gặp chính xác vị trí vật. Trong nghiên cứu này sử dụng phương pháp tìm các điểm trung gian đã biết trước vị trí tọa độ trong cả hệ tọa độ camera và hệ tọa độ robot, từ đó ta áp dụng tính toán ma trận để tìm ra ma trận chuyển đổi tức thông số ngoại của camera.

Một cách tổng quát, để chuyển một điểm từ hệ tọa độ này về một hệ tọa độ khác, ta cần tìm ma trận chuyển M.

$$P_R = MP_C$$

$$\Leftrightarrow \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}_R = \begin{bmatrix} R_{11} & R_{12} & R_{13} & T_X \\ R_{21} & R_{22} & R_{23} & T_Y \\ R_{31} & R_{32} & R_{33} & T_Z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}_C \quad (6)$$

Trong đó:

P_R là tọa độ điểm trung gian đã biết trước trong hệ tọa độ Robot.

P_C là tọa độ điểm trung gian đã biết trước trong hệ tọa độ Camera.

Ta thấy M có tới 12 ẩn cần tìm, vì thế cần 12 phương trình độc lập để giải M. Do đó, số điểm trung gian tối thiểu cần xác định là 4 điểm.

Sử dụng thuật toán Hough Circle trên OpenCV để xác định vị trí tâm hình tròn, tương ứng tại vị trí tâm hình tròn ta thu được các tọa độ thực trong hệ tọa độ camera. Di chuyển Robot đến vị trí tâm hình tròn và đọc về tọa độ tương ứng. Phương pháp này đòi hỏi đầu dò gắn trên tool Robot phải đủ nhỏ, và offset chiều dài đầu dò trong bộ thông số Robot phải chính xác. Như vậy, ta đã tìm được tọa độ các điểm trung gian trong cả hai hệ tọa độ.

Hệ phương trình (6) là một hệ tuyến tính bình thường nên có thể chuyển thành dạng ma trận để thuận tiện tính toán:

$$\begin{bmatrix} X_1 \\ Y_1 \\ Z_1 \\ X_2 \\ Y_2 \\ Z_2 \\ X_3 \\ Y_3 \\ Z_3 \\ X_4 \\ Y_4 \\ Z_4 \end{bmatrix} = \begin{bmatrix} x_1 & y_1 & z_1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & x_1 & y_1 & z_1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & x_1 & y_1 & z_1 & 0 & 0 & 1 \\ x_2 & y_2 & z_2 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & x_2 & y_2 & z_2 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & x_2 & y_2 & z_2 & 0 & 0 & 1 \\ x_3 & y_3 & z_3 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & x_3 & y_3 & z_3 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & x_3 & y_3 & z_3 & 0 & 0 & 1 \\ x_4 & y_4 & z_4 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & x_4 & y_4 & z_4 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & x_4 & y_4 & z_4 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} R_{11} \\ R_{12} \\ R_{13} \\ R_{21} \\ R_{22} \\ R_{23} \\ R_{31} \\ R_{32} \\ R_{33} \\ T_X \\ T_Y \\ T_Z \end{bmatrix} \quad (7)$$

$$\Leftrightarrow A = BK$$

Vậy M có thể được tìm qua các giá trị của vector K:

$$K = B^{-1}A \quad (8)$$

3.2. Tính toán vận tốc di chuyển của vật

Quá trình tính toán vận tốc được kích hoạt qua điều kiện “Check speed”. Chương trình sẽ liên tục lấy dữ liệu vị trí và thời gian của đối tượng trên băng tải. Khi kết thúc quá trình này sẽ tiến hành tính ra vận tốc của băng tải qua hồi quy tuyến tính. Giá trị vận tốc được cập nhật lên hệ thống để dùng cho các mục đích tính toán khác.

Ta giả định việc setup vận tốc băng tải được thực hiện qua núm xoay biến trở và trong quá trình hoạt động, vận tốc băng tải không đổi bởi các điều kiện khác nhau. Ở phần trước, việc tiền xử lý ảnh đưa ra được các đối tượng trên băng tải, ta có thể tìm được tâm của đối tượng đó qua việc tính các moment ảnh ở mỗi thời điểm khác nhau.

Khi một đối tượng di chuyển trên băng tải, với vận tốc v là hằng số, ta có phương trình:

$$X = v.t + x_0 \quad (9)$$

Từ đó, ta thu thập được bảng số liệu sau:

Bảng 1. Bảng số liệu thu thập

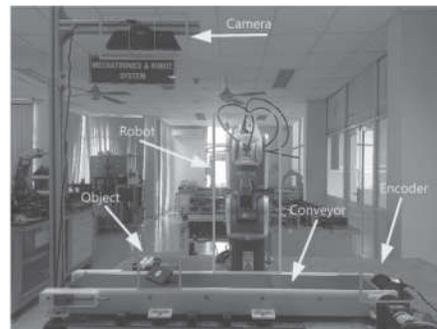
Vị trí	Thời gian
X1	T1
X2	T2
.....
X4	T4

Các giá trị thu được là kết quả quan sát của một đối tượng.

Do hệ thống tách biệt vùng làm việc của camera và vùng làm việc của robot nên việc xác định vận tốc của của băng tải hết sức quan trọng. Nó là một thông số quan trọng để xác định vị trí của vật theo thời gian và phải có sai số nhỏ để đảm bảo robot có thể bắt chính xác được vị trí của vật. Để xác định chỉ tiêu này, ta sử dụng thêm một encoder ngoài hệ thống và cho băng tải chạy ở các mức vận tốc khác nhau để tiến hành so sánh kết quả.

4. THỰC NGHIỆM

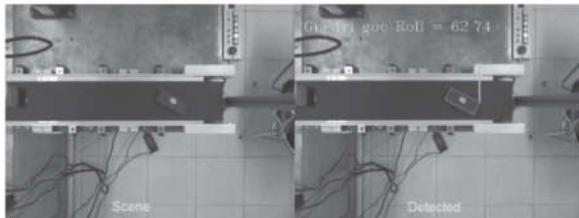
Quá trình thực nghiệm được thực hiện trên robot Nachi 6 bậc tự do, camera Kinect được đặt trên cao. Vật cần gắp di chuyển trên băng tải như hình 6. Tốc độ di chuyển của băng tải là 5mm/s. Băng tải không dừng trong suốt quá trình gắp.



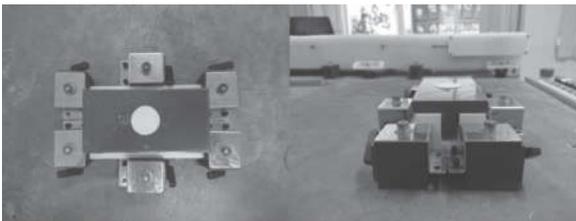
Hình 6. Môi trường thực nghiệm

Cho vật di chuyển trên băng tải, Kinect liên tục thu nhận dữ liệu, đồng thời thực hiện các bước filter để tách vật ra khỏi nền. Khi đủ dữ liệu như yêu cầu, chuyển sang các thuật toán ước lượng vị trí của vật. Hiện thị vật đã nhận diện được cùng vector pháp tuyến của bề mặt gắp. Nhận diện vị trí của vật trên băng tải (các thông số X, Y, Z, Roll, Pitch, Yaw) như hình 7. 

Ghi nhận vị trí ban đầu của vật trên băng tải thông qua Encoder. Di chuyển robot đến điểm đợi tracking. Khi vật di chuyển đến điểm đợi tracking, ra lệnh cho robot thực hiện bám theo theo và gấp vật. Sau khi gấp, di chuyển vật đến vị trí lắp. Xoay vật về đúng góc và lắp vật vào đế như hình 8. Thời gian xử lý của hệ thống được thể hiện ở bảng 2.



Hình 7. Hình thực nghiệm tính toán vị trí và hướng vật di chuyển trên băng chuyên



Hình 8. Kết quả lắp ghép

Bảng 2. Thời gian xử lý của hệ thống

STT	Tính toán vị trí góc (ms)	Robot di chuyển (s)
1	50	30
2	45	27
3	55	29
4	42	32
5	48	35

Nhận xét:

- Hệ thống hoạt động tốt đối với các vật cần gấp di chuyển ở vị trí khác nhau.
- Thời gian xử lý ảnh tổng cộng không vượt quá 55ms, ứng dụng có thể chạy thời gian thực rất tốt.
- Thời gian thực thi của robot trung bình khoảng 32s phù hợp với tác vụ lắp ráp nhanh trên băng chuyên.

5. KẾT LUẬN

Nghiên cứu đã trình bày một phương pháp hiệu quả để tự động hóa quá trình lắp ráp vật trong dây chuyền sản xuất bằng cách sử dụng công nghệ xử lý hình ảnh và tay máy công nghiệp. Qua thực nghiệm có thể thấy phương pháp có thể thay thế việc lắp ráp thủ công hiện nay của người lao động, tăng năng suất cho quá trình lắp ráp cũng như bảo vệ sức khỏe người lao động tốt hơn. Cách làm cũng đơn giản và dễ thực hiện cũng như dễ bảo trì. Để tăng độ chính xác, hệ thống camera có thể dễ dàng thay đổi.

Lời cảm ơn:

Tôi xin cảm ơn Trường Đại học Bách khoa, Đại học Quốc gia Thành phố Hồ Chí Minh đã hỗ trợ thời gian, phương tiện và cơ sở vật chất cho nghiên cứu này.

Xung đột lợi ích:

Tác giả xin cam đoan rằng không có bất kỳ xung đột lợi ích nào trong công bố bài báo. ❖

Ngày nhận bài: **12/11/2025**

Ngày phản biện: **24/11/2025**

Tài liệu tham khảo:

- [1]. Efficient programming and teaching of DENSO robots, <https://www.artiminds.com/robot-applications/robot-programming/denso-robot-programming/>
- [2]. Robotic Solutions for Packaging, <https://www.motoman.com/en-us/applications/picking-packing>
- [3]. Solutions for Robotic Pick and Place and Packaging Automation, <https://www.fanucamerica.com/solutions/applications/picking-and-packing-robots>
- [4]. ABB confectionery palletizing solution enhances collaboration and improves productivity by 53 percent, <https://new.abb.com/news/detail/73220/cstmr-abb-confectionery-palletizing-solution-enhances-collaboration-and-improves-productivity-by-53-percent>
- [5]. F. Casado, Y. L. Lapido, D.P. Losada and A.. S. Alonso, “*Pose estimation and object tracking using 2D image*”. 27th International Conference on Flexible Automation and Intelligent Manufacturing, 27-30 June, 2017, Modena, Italia, pp. 63-71.
- [6]. I. Wesis and M. Ray, “*Model-based recognition of 3D objects from single images*”. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 23, 2001, pp. 116-128.
- [7]. K.T. Song, C. H. Wu and S. Y. Jiang, “*CAD-based Pose Estimation Design for Random Bin Picking Using a RGB-D Camera*”. Journal of Intelligent & Robotics System, Vol. 87, 2017, pp.455-470.
- [8]. C. Choi, Y. Taguchi, O. Tuzel, M. Y. Lin and S. Ramalingam, “*Voting-based pose estimation for robotic assembly using a 3D sensor*”. 2012 IEEE International Conference on Robotics and Automation, 2012, USA, pp. 180-186.
- [9]. A.K. Jain, and P.J. Flynn, “*CAD-based computer vision: from CAD model to relational graphs*”. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 13, 1991, pp.114-132.
- [10]. J. Majumdar and A. G. Seethalakshmy, “*ACAD model based system for Object Recognition*”. Journal of Intelligent and Robotics Systems, 1997, Vol. 18, pp. 352-365.
- [11]. A. Menon, B. Cohen and M. Likhachev, “*Motion planning for smooth pickup of moving objects*”. 2014 IEEE international Conference on Robotics and Automation, 2014, China, pp. 453-460.