

**ĐẠI HỌC QUỐC GIA HÀ NỘI
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ**



BỘ MÔN: ĐỘNG HỌC VÀ ĐỘNG LỰC HỌC

ĐỀ TÀI: SỬ DỤNG TAY MÁY ABB IRB 140 ĐỂ VẼ HÌNH ẢNH MONG MUỐN

Lớp K67R

Ngành Kỹ thuật robot

Khoa Điện tử - Viễn Thông

Sinh viên thực hiện

Nguyễn Huy Thắng - 22027545

Nguyễn Bảo Long - 22027537

Dương Vũ Việt Thắng - 22027529

Hà Nội, ngày 9 tháng 6 năm 2024

Mục lục

TÓM TẮT	2
PHẦN I: GIỚI THIỆU.....	2
1.1 Phần mềm sử dụng:.....	2
1.2 Xây dựng mô hình VREP:	2
1.3 Xây dựng code MATLAB:	3
PHẦN 2: PHƯƠNG PHÁP THỰC HIỆN	3
2.1 Thiết lập tay máy ABB IRB 140:	3
2.2 Chỉnh sửa code:	3
PHẦN 3: KẾT QUẢ.....	4
PHẦN 4: THẢO LUẬN VÀ KẾT LUẬN	4
4.1 Thảo luận:	4
4.2 Kết luận:	5
TÀI LIỆU THAM KHẢO	5

TÓM TẮT

Trong báo cáo này, nhóm em đã tìm hiểu và thực hiện thành công hệ thống điều khiển cánh tay robot vẽ hình ảnh dựa trên dữ liệu xử lý ảnh.

Nhóm em đã sử dụng phần mềm Vrep để tạo ra môi trường mô phỏng cánh tay robot được lập trình trên Matlab để vẽ theo biên của một hình ảnh nhị phân.

Kết quả thực nghiệm đạt được: nhóm em đã mô phỏng và điều khiển thành công tay máy thực hiện vẽ hình ảnh đầu vào thông qua phần mềm Matlab và Vrep. Kết quả cho thấy rằng tay máy thực hiện thành công tác vụ vẽ lại hình ảnh đầu vào.

PHẦN I: GIỚI THIỆU

1.1 Phần mềm sử dụng:

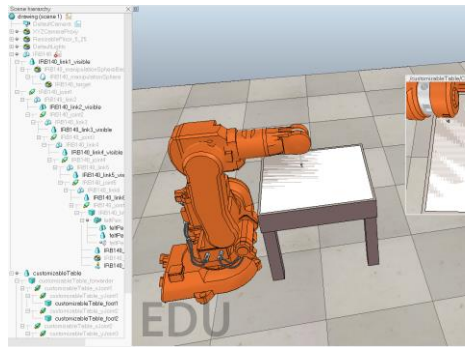
- MATLAB và VREP.

1.2 Xây dựng mô hình VREP:

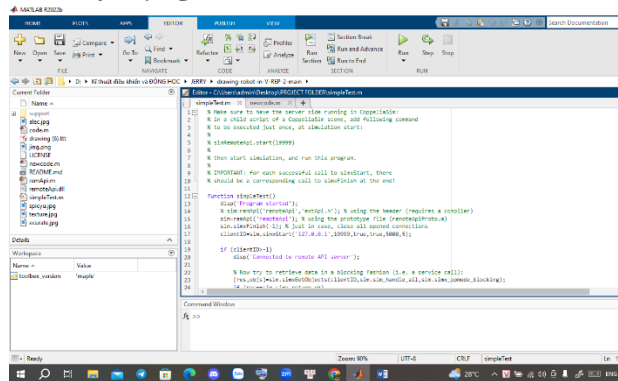
- Sử dụng tay máy ABB IRB 140.

- Add inverse kinematic các khớp của tay máy.
- Add Felt pen để vẽ hình ảnh
- Add custom table.

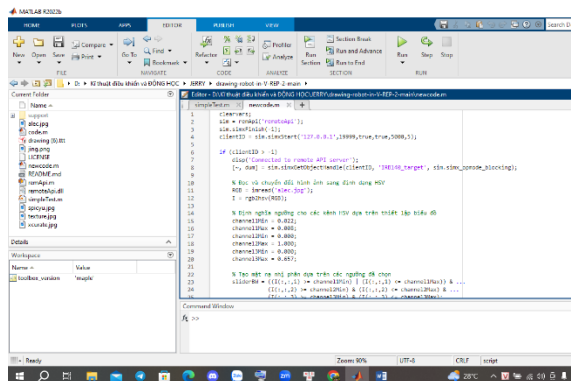
- Add camera để nhìn rõ hình ảnh về.



1.3 Xây dựng code MATLAB:



File simple test dùng để check kết nối giữa qua cổng API giữa 2 phần mềm MATLAB và VREP

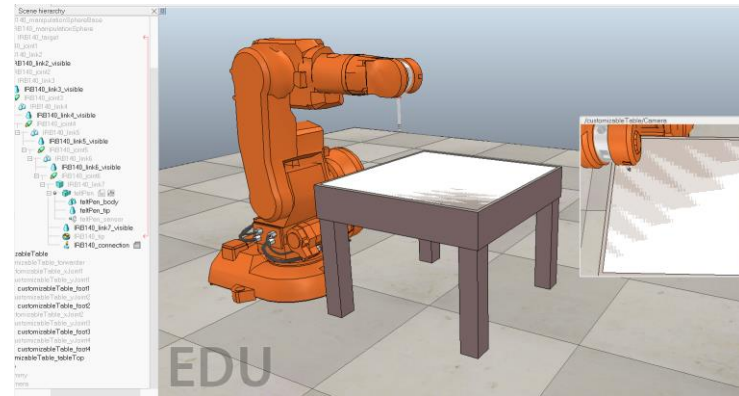


Code thực hiện đọc hình ảnh và vẽ ảnh

PHẦN 2: PHƯƠNG PHÁP THỰC HIỆN

2.1 Thiết lập tay máy ABB IRB 140:

- Sử dụng tay máy ABB IRB 140 có gắn inverse kinematic ở các khớp joint của tay máy theo mẫu tài liệu tham khảo.
 - Thiết lập felt pen gắn với khớp cuối của tay máy.
 - Thiết lập kết nối qua cổng API để link matlab và vrep.
- => Chạy thử tay máy.



2.2 Chỉnh sửa code:

- Dựa trên code mẫu sử dụng ngôn ngữ python, nhóm em đã nghiên cứu và thực hiện code lại theo ngôn ngữ matlab và chia ra các tác vụ nhỏ hơn để xử lý.
- Ý tưởng: Xử lý dữ liệu hình ảnh đầu vào và chuyển đổi thành các dữ liệu nhỏ, dễ tiếp cận hơn để cánh tay robot bám theo và xử lý:
- + Kiểm tra kết nối giữa vrep và matlab thông qua cổng API.
- + Đọc hình ảnh đầu vào với hàm 'imread' trả về dữ liệu ma trận 3 chiều [MxNx3] với 3 màu cơ bản RGB. Sau đó chuyển vào không gian màu HSV.
- + Thiết lập giá trị cho các kênh HSV để tạo mặt nạ nhị phân dựa trên các ngưỡng đã thiết lập.
- + Tạo ảnh nhị phân từ mặt nạ nhị phân để trích xuất biên.
- + Tạo quỹ đạo để cánh tay robot bám theo các biên.
- + Hoàn thành tác vụ, tay máy kết thúc và trở về vị trí ban đầu.

```

10 % Đọc và chuyển đổi hình ảnh sang định dạng HSV
11 RGB = imread('alec.jpg');
12 I = rgb2hsv(RGB);
13
14 % Định nghĩa ngưỡng cho các kênh HSV dựa trên thiết lập biểu đồ
15 channel1Min = 0.022;
16 channel1Max = 0.008;
17 channel2Min = 0.000;
18 channel2Max = 1.000;
19 channel3Min = 0.000;
20 channel3Max = 0.657;
21
22 % Tạo mặt nạ nhị phân dựa trên các ngưỡng đã chọn
23 sliderBW = ((I(:,:,1) >= channel1Min) & (I(:,:,1) <= channel1Max)) & ...
24 (I(:,:,2) >= channel2Min) & (I(:,:,2) <= channel2Max) & ...
25 (I(:,:,3) >= channel3Min) & (I(:,:,3) <= channel3Max);
26 BW = sliderBW;
27
28 % Điều chỉnh hình ảnh để phù hợp với cách vẽ của robot
29 BW = flip(BW, 1);
30 BW = imrotate(BW, -90);
31
32 % Hiển thị ảnh nhị phân
33 figure(1);
34 imshow(BW);
35
36 % Tạo ảnh nhị phân đảo ngược để trích xuất biên
37 ibw_white = 1 - BW;
38
39 % Trích xuất các biên
40 [B, L] = bwboundaries(ibw_white, 'noholes');
41
42
43 % Tạo quỹ đạo di chuyển cho cánh tay robot
44 x = [];
45 y = [];
46 z = [];
47 count = 0;
48
49 for k = 1:length(B)
50     boundary = B(k);
51     for i = 1:length(boundary(:,2))
52         count = count + 1;
53         x(count) = boundary(i,2);
54         y(count) = boundary(i,1);
55         z(count) = 0;
56     end
57     count = count - 1;
58     z(count) = 30;
59 end
60
61 % Hệ số tỷ lệ để điều chỉnh kích thước vẽ
62 scaleFactor = 0.0005; % Tăng hệ số này để vẽ hình lớn hơn
63
64 % Di chuyển cánh tay robot theo quỹ đạo
65 for m = 1:length(x)
66     [returnValue] = sim.simSetObjectPosition(clientID, dum, -1, ...
67         [-0.22 + (x(m) * scaleFactor), -0.175 + (y(m) * scaleFactor), (z(m) * 0.004) + 0.
68     end
69
70 % Di chuyển cánh tay robot ra khỏi khu vực vẽ
71 [returnValue] = sim.simSetObjectPosition(clientID, dum, -1, [-0.4, -0.45, 0.625], sim.sir
72
73 % Kết thúc kết nối

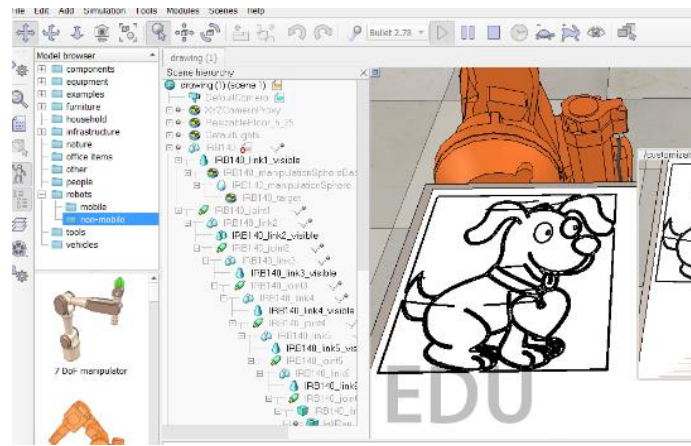
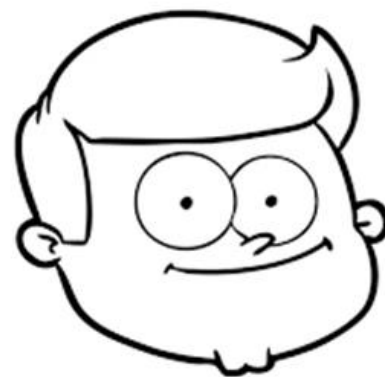
```

PHẦN 3: KẾT QUẢ

- Ảnh đầu vào:



- Ảnh vẽ được:



- Đánh giá: Hình ảnh được vẽ lại rõ nét và có độ chính xác với ảnh đầu vào cao.

Tuy nhiên thời gian vẽ chỉ tiết hơi lâu, cần cải thiện.

PHẦN 4: THẢO LUẬN VÀ KẾT LUẬN

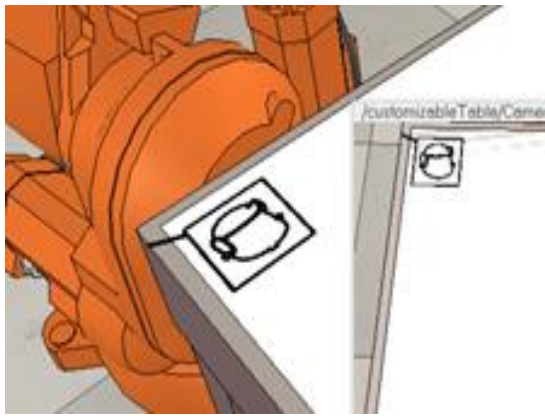
4.1 Thảo Luận:

Tuy nhiên có một số lỗi trong quá trình vẽ gây ra hình ảnh đầu ra không đạt được độ chính xác cao như mong đợi.

- + Lỗi do chất lượng hình ảnh đầu vào kém và thiết lập thông số kênh màu HSV không phù hợp gây ra sai lệch giữa độ màu, độ bão hòa và độ sáng khi đọc hình ảnh vào.
- + Một số ảnh kích thước quá lớn sẽ phải chỉnh lại biến 'scaleFactor' trục x,y khi tay máy thực hiện vẽ trên mặt phẳng là giấy nếu không tay máy sẽ không đủ dài để vẽ gây ra thiếu nét.

- Ví dụ cho lỗi trên:

Ảnh đầu vào



4.2 Kết luận:

- Trong bài tập lớn cuối kì này, chúng em đã trình bày chi tiết về việc sử dụng tay máy để xử lý và vẽ hình ảnh, thông qua việc áp dụng mô phỏng trên phần mềm VREP và thực hiện các chạy code trên phần mềm MATLAB. Kết quả của chúng em tuy gặp phải một số lỗi về hình ảnh khi thử nghiệm nhưng nhìn chung đã hoàn thành được mục tiêu môn học đề ra.
- Ngoài ra bài tập lớn còn giúp chúng em hiểu thêm kiến thức về môn học và tiếp thu những kiến thức, bổ ích lớn để củng cố cho quá trình học tập sau này.
- Cuối cùng nhóm em xin chân thành cảm ơn tới sự giúp đỡ nhiệt tình của thầy Hưng và anh Minh đã giúp chúng em hoàn thành bài tập cuối kì này.

TÀI LIỆU THAM KHẢO

1. <https://www.youtube.com/watch?v=eTd6mOk6Njw> – Inverse kinematics tutorial
2. <https://www.slideshare.net/slideshow/xu-ly-anh/7360853> – Guide about image processing function.
3. <https://github.com/camilof/Robotarm-ABB140-AutoDrawing> – Python code drawing
4. https://github.com/mathworks/getting-started-with-image-processing/blob/main/createMask.m?fbclid=IwZXh0bgNhZW0CMTAAAR2yZjyD37yJD3JjuyGGVvYXcWxJQ8DrMWUqektELYJznnDg7tZlx9mSB6gk_aem_AUe-KLoEx5vb9X49gXIkEb5Nd7fvYeOwAwUhBIGKxbW4uJ2kVqsZ7LJcKRpqo-iVC7fnk61cxKqAZO8eN3bqSia- – Example image processing functions.