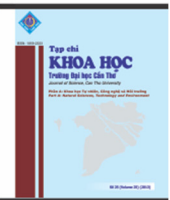




Tạp chí Khoa học Trường Đại học Cần Thơ  
website: [sj.ctu.edu.vn](http://sj.ctu.edu.vn)



## ĐIỀU KHIỂN ROBOT PIONEER P3-DX BẮM SÁT ĐỐI TƯỢNG

Mã Trường Thành<sup>1</sup>, Phạm Nguyên Khang<sup>2</sup>, Lâm Chí Nguyên<sup>3</sup>, Triệu Thanh Ngoan<sup>3</sup>, Hà Duy An<sup>3</sup> và Đỗ Thanh Nghị<sup>3</sup>

<sup>1</sup> Khoa Kỹ thuật - Công nghệ, Trường Cao đẳng Cộng đồng Sóc Trăng

<sup>2</sup> Khoa Công nghệ Thông tin & Truyền thông, Trường Đại học Cần Thơ

<sup>3</sup> Khoa Công nghệ Thông tin & Truyền thông, Trường Đại học Cần Thơ

### Thông tin chung:

Ngày nhận: 03/09/2013

Ngày chấp nhận: 21/10/2013

### Title:

Control of robot Pioneer P3-DX for object tracking

### Từ khóa:

Camshift, meanshift, phát hiện đối tượng, bám sát đối tượng, đặc trưng haar-like, adaboost, robot Pioneer, cấu trúc phân tầng

### Keywords:

Camshift, meanshift, object detection, tracking object, haar-like feature, adaboost, robot Pioneer, cascades of boosted classifiers

### ABSTRACT

In this paper, we present how to combine the machine learning algorithms into Pioneer P3-DX Robot that control the Robot's behaviors, movements, and tracking according to the real time. The machine learning algorithm, Cascades of Boosted Classifiers with Haar-Like Feature, can be used to detect and recognize objects. Next, we used Camshift Algorithms (Continuously Adaptive Meanshift) to monitor the object's activities within a flat image. Finally, based on the object's size and position in the flat image, we control the Robot to move and keep track of the object. The experimental results so that the Robot has been operated as the target objectives such as controlling behaviors, moving and keeping track of an object.

### TÓM TẮT

Trong bài này, chúng tôi trình bày cách kết hợp các giải thuật máy học vào Robot Pioneer P3-DX nhằm điều khiển các hành vi, di chuyển và bám sát của Robot theo thời gian thực. Giải thuật máy học Cascades of Boosted Classifiers sử dụng đặc trưng Haar-Like Feature có thể phát hiện, nhận dạng đối tượng. Tiếp theo chúng tôi sử dụng giải thuật CamShift (Continuously Adaptive MeanShift) để giám sát hoạt động của đối tượng trong phạm vi mặt phẳng ảnh. Bước cuối cùng là tính toán vị trí và kích thước của đối tượng trên mặt phẳng ảnh để điều khiển Robot hoạt động và bám sát theo đối tượng. Kết quả thử nghiệm cho thấy Robot đã hoạt động đảm bảo các mục tiêu đặt ra như điều khiển các hành vi, di chuyển và bám sát đối tượng.

## 1 GIỚI THIỆU

Nghiên cứu điều khiển Robot là bài toán được các nhà khoa học quan tâm, nhằm phục vụ cho con người ở nhiều lĩnh vực ứng dụng như: Robot khuôn hàng hóa theo chủ của nó, Robot dọn rác, lau nhà, hỗ trợ người khuyết tật, Robot trong công nghiệp ô tô, thăm dò khai thác mỏ,... Do có tính ứng dụng cao nên các nhà nghiên cứu đã bắt tay vào phát triển Robot thông minh hơn, phục vụ tốt cho nhu

cầu phát triển kinh tế, xã hội. Để làm được điều đó, cần có sự kết nối giữa “bộ não thông minh” và Robot để tạo nên những Robot thông minh như ngày nay. Máy học chính là nền tảng giúp Robot có thể thông minh, hoạt động tinh vi hơn. Trong các hướng nghiên cứu khác nhau về Robot, nghiên cứu theo dõi, bám sát đối tượng là một trong những bài toán hay và được ứng dụng nhiều trong thực tế. Những Robot này có thể thực hiện nhiều hành động khác nhau thông qua hành vi của đối tượng

để đáp ứng tốt cho con người nhờ vào sử dụng các mô hình máy học tự động. Hiện nay các nghiên cứu liên quan đến phát hiện và bám sát đối tượng bao gồm các hướng tiếp cận như sau: dự đoán chuyển động của đối tượng đang quan sát như bộ lọc Kalman [7][5] kết hợp với thuật toán MeanShift [14] do Fukunaga và Hostetler trình bày nhằm mục đích nâng cao khả năng việc bám sát. Hướng tiếp cận bằng phương pháp xây dựng mô hình cũng được nhiều tác giả thực hiện, chẳng hạn như Anurag Mital và Mikos Paragios [3] đã dùng phương pháp ước lượng mật độ bằng hàm nhân thích nghi (*adaptive kernel density estimation*) để tính toán. Phương pháp này được đánh giá là tốt về mặt không gian lưu trữ nhưng về việc tính toán trên thời gian thực thì còn hạn chế bởi tính toán phức tạp và tốc độ chậm. Ngoài ra, hướng tiếp cận được nhiều người hướng đến là bám sát theo màu sắc như thuật toán CamShift [11], MeanShift [20] hay hướng tiếp cận theo đặc trưng góc cạnh của Cox và Hingorani đã sử dụng MHT [12] (*Multiple Hypothesis Tracking*), hoặc là sự kết hợp của “particle filters” và kỹ thuật tính toán xác suất do MacCormick và Blake [12]. Nhìn chung, các cách tiếp cận đều có những ưu nhược điểm riêng nhằm giải quyết các bài toán khác nhau về giám sát. Với những hướng tiếp cận trên đều tiến tới mục đích là làm cho Robot di chuyển và bám theo đối tượng. Hầu hết các công trình nghiên cứu đều thực hiện về bám theo người [6][9] hay bám sát theo quả bóng [1] và những vật thể khác. Nếu nói về các thuật toán dùng để bám đối tượng trong ảnh rất đa dạng, phải kể đến một số thuật toán như: Thuật toán KLT, Correlation-base template matching, thuật toán Meanshift, thuật toán Camshift.

Trong bài báo này, chúng tôi áp dụng thuật toán Camshift để bám sát đối tượng dựa trên màu sắc đối tượng trên ảnh để Robot di chuyển theo, do tính hiệu quả về độ chính xác, độ phức tạp về thời gian. Chúng tôi đã sử dụng Robot Pioneer P3-DX kết hợp với máy học để giải quyết bài toán: “Điều khiển Robot bám sát đối tượng”. Chúng tôi trích các đặc trưng Haar-Like [18][16] kết hợp với bộ phân tầng Cascades of Boosted Classifiers và Adaboost [16][17] để huấn luyện cho Robot nhận dạng đối tượng - đối tượng được sử dụng để Robot nhận dạng là quả bóng. Sau khi Robot phát hiện và nhận dạng được đối tượng, chúng tôi sẽ sử dụng giải thuật CamShift để Robot bám lấy quả bóng. Từ đó, dựa trên mặt phẳng ảnh có đối tượng đang hiện hữu trong đó sẽ cho Robot di chuyển thông qua việc tính khoảng cách và lấy tỷ lệ cho vận tốc để Robot di chuyển theo đối tượng quan sát. Nhờ

vào việc xám hóa ảnh khi phát hiện và nhận dạng sẽ giúp cho đối tượng có thể linh động, giải thuật CamShift [11] sẽ hiệu quả hơn trong bám sát. Với trường hợp, giải thuật CamShift bám sai đối tượng thì bộ phận phát hiện và nhận dạng đối tượng sẽ liên tục tìm đối tượng trên ảnh (*lúc này Robot sẽ đứng yên hoặc xoay tìm đối tượng*) cho đến khi phát hiện và giải thuật CamShift được thực hiện. Kết quả cho thấy Robot đã hoạt động như mục tiêu đặt ra và hạn chế được tình trạng mất bám trong các môi trường khác nhau cũng như kiểm soát được khoảng cách và vận tốc di chuyển của Robot.

Phần tiếp theo của bài báo được tổ chức như sau: phần 2 trình bày ngắn gọn về Robot di động – Robot Pioneer P3-DX. Phần 3: trình bày điều khiển Robot bám sát đối tượng: phát hiện và nhận dạng đối tượng với Haar-Like feature kết hợp Cascades of Boosted Classifiers và Adaboost, trình bày thuật toán Camshift để bám sát đối tượng và cách di chuyển của Robot trên mặt phẳng ảnh, cách tính khoảng cách tương ứng với vận tốc điều khiển Robot. Kết quả thực nghiệm được trình bày ở phần 4 trước khi kết luận và hướng phát triển trong phần 5.

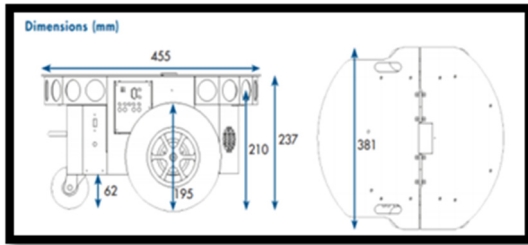
## 2 ROBOT DI ĐỘNG – ROBOT PIONEER P3-DX

Mobile Robot hay còn gọi là Robot di động, Robot tự hành. Đây là một loại Robot có vai trò quan trọng trong ngành Robot học. Cùng với sự phát triển của các hệ thống về lĩnh vực công nghệ thông tin, Robot di động đã góp phần mạnh mẽ trong công nghiệp, y tế và cuộc sống thực tế hiện nay. Robot di động là một loại máy móc tự động có khả năng di chuyển trong một phạm vi môi trường nhất định.

Robot được sử dụng trong bài viết là Robot Pioneer P3-DX [2]. Đây là dòng Robot di động chế tạo bởi hãng Adept Mobile Robot. Robot Pioneer là một trong những Robot được nhiều nhà chuyên gia đa lĩnh vực nghiên cứu đến và được sử dụng phổ biến nhất trên thế giới hiện nay. Robot này được tạo ra và cho phép người nghiên cứu hoàn toàn có thể “*lập trình được*”.

Robot Pioneer P3-DX [2] là loại Robot di động nhỏ, trọng lượng nhẹ với 3 bánh xe (2 bánh trước chủ động (*bánh chính*) và bánh sau di chuyển tự do (*bánh phụ - rê*)), bánh xe với đường kính 19.5 cm, thân được bao bọc bởi kim loại nhôm cứng cáp. Phía trước được trang bị 8 cảm biến Sonar, phía sau cũng được trang bị 8 cảm biến Sonar – đây là dạng cảm biến siêu âm để dò tìm vật cản. Robot

Pioneer 3-DX có 3 pin nhằm dự trữ và có thể thay đổi nhanh chóng.



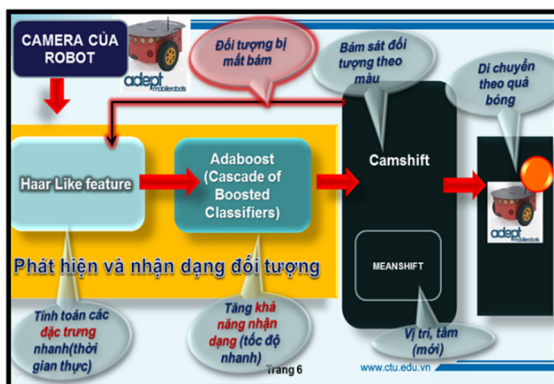
Hình 1: Cấu tạo Robot Pioneer 3-DX [2]

Vì khả năng có thể lập trình được nên hãng đã hỗ trợ bộ SDK (software development kit) dành cho những nhà nghiên cứu Robot, các loại máy tính có thể dễ dàng cài đặt và tích hợp vào để Robot có thể hoạt động nhịp nhàng.

Robot Pioneer 3-DX có thể di chuyển tới và đi chuyển lui với vận tốc tối đa là 1.2 m/s và có tải trọng đồ vật lên đến 23 kg.

### 3 ĐIỀU KHIỂN ROBOT PIONEER 3-DX BẮM SÁT ĐỐI TƯỢNG

Hệ thống đề xuất cho Robot bám sát đối tượng



Hình 2: Mô hình hoạt động của hệ thống

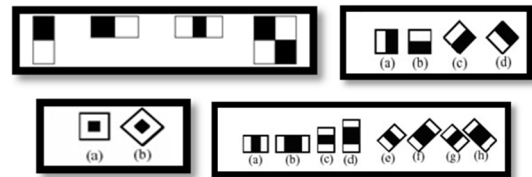
Robot sẽ được tích hợp một Webcam/Camera để quan sát, dựa vào frame ảnh thu được hệ thống sẽ sử dụng Haar-Like features và Adaboost để phát

hiện đối tượng cần nhận dạng trên từng frame ảnh. Dựa vào đối tượng đã được nhận dạng ở trên, hệ thống sẽ sử dụng thuật toán CamShift để bắt bám đối tượng trong quá trình di chuyển (sử dụng MeanShift để lấy vị trí và trọng tâm mới của đối tượng). Sau khi đã bắt, bám được đối tượng thì hệ thống sẽ dựa vào kích thước và mặt phẳng ảnh để điều khiển Robot di chuyển. Trong trường hợp thuật toán CamShift bị mất bám thì bộ phận phát hiện và nhận dạng đối tượng sẽ thực hiện lại việc dò tìm.

#### 3.1 Phát hiện và nhận dạng đối tượng

##### 3.1.1 Trích đặc trưng Haar-Like và Cascades of Boosted Classifiers

Đặc trưng Haar-Like [18][16] là những đặc trưng ảnh số được sử dụng trong nhận dạng đối tượng. Thông thường, những đặc trưng cơ bản Haar-Like dùng để dò khuôn mặt trong thời gian thực, do đó Haar-Like là sự lựa chọn phù hợp cho hệ bám sát đối tượng nhằm nhận dạng nhanh chóng, chính xác trong thời gian thực.

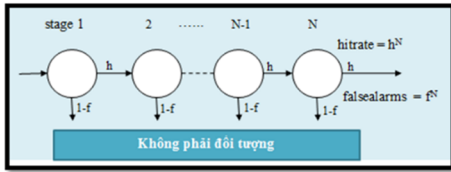


Hình 3: Đặc trưng Haar-Like của Viola và Jones [16]

Đặc trưng Haar-like sẽ diễn đạt được tri thức về các đối tượng trong ảnh (bởi vì nó biểu diễn mối liên hệ giữa các bộ phận của đối tượng), điều mà bản thân từng điểm ảnh không diễn đạt được – dựa trên việc co giãn của khung cửa sổ tìm kiếm. Để tính giá trị các đặc trưng Haar-Like, cần tính sự chênh lệch giữa tổng của các pixel của các vùng đen và các vùng trắng như trong công thức sau:

$$f(x) = \sum_{\text{đen}} (\text{pixel}) - \sum_{\text{trắng}} (\text{pixel}) \quad (1)$$

Lý do đặc trưng Haar-Like được tính toán nhanh chính là nhờ vào khái niệm tích phân ảnh (integration) mà Viola và Jones đã đề xuất. Kết hợp với bộ cấu trúc phân tầng với sự huấn luyện của một tầng (cascade) gồm Ngai đoạn (stages). Ở mỗi giai đoạn, phân lớp yếu (weak classifier) tương ứng sẽ được huấn luyện sao cho độ chính xác của nó là  $h$  (tỷ lệ nhận dạng chính xác đối tượng) và false alarm bằng  $f$  (tỷ lệ nhận dạng lầm).



Hình 4: Cấu trúc phân tầng với N giai đoạn [16]

### 3.1.2 AdaBoost

AdaBoost [19] (*Adaptive Boost*) được Freund và Schapire đề xuất trong những năm 1995 là một bộ phân loại mạnh phi tuyến. AdaBoost hoạt động trên nguyên tắc kết hợp tuyến tính các phân lớp yếu (*weak classifier*). Nghiên cứu của Viola và Jones [16] xây dựng AdaBoost dựa trên dữ liệu đầu vào là các đặc trưng Haar- Like để hình thành một phân lớp mạnh hơn (*strong classifier*) theo nguyên tắc của boosting [19]: “**tăng trọng số của các mẫu bị nhận dạng sai và giảm trọng số của các mẫu được nhận dạng đúng bởi phân lớp yếu (*weak classifier*) vừa xây dựng**”. Ý tưởng chính của Boosting là thực hiện lặp lại nhiều lần quá trình học của bộ phân lớp yếu, nghĩa là tập trung học lại các phần tử bị phân lớp sai. Tại mỗi giai đoạn (stages) của Cascades of Boosted Classifiers được huấn luyện là một AdaBoost.

Các bộ phân lớp yếu  $h_k(x)$  là các bộ phân lớp yếu được lấy từ việc trích các đặc trưng Haar-Like, được biểu diễn như sau [16]:

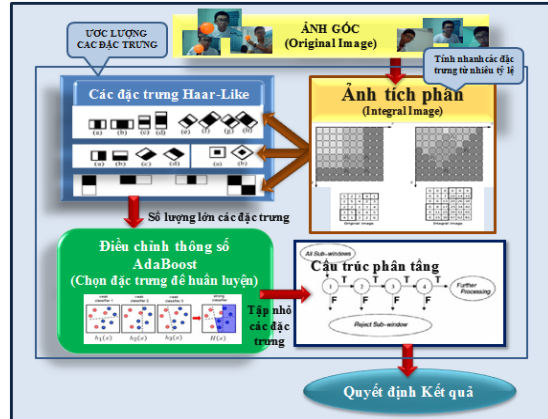
$$h_k(x) = \begin{cases} 1 & \text{nếu } p_k f_k(x) < p_k \theta_k \\ 0 & \text{nếu ngược lại} \end{cases} \quad (2)$$

- $X$  : mẫu hay cửa sổ con cần xét ( $X = (x_1, x_2, \dots, x_n)$  là vector đặc trưng mẫu);
- $\theta_k$  : ngưỡng ( $\theta = \text{teta}$ );
- $f_k$  : giá trị của đặc trưng Haar-like (1);
- $p_k$  : hệ số chiều của bất phương trình (*a parity indicating the direction of the inequality sign*).

Bộ phân lớp mạnh  $h_{\text{mạnh}}(x)$  được tạo ra bằng cách kết hợp tuyến tính N bộ phân loại yếu (2). Vấn đề cốt lõi là thực hiện chọn các trọng số  $\alpha_1, \dots, \alpha_n$  thích hợp [19].

$$h_{\text{mạnh}}(x) = \text{sign} \sum_{k=1}^N \alpha_k \cdot h_k^{\text{yếu}}(x) \quad (3)$$

### 3.1.3 Nhận dạng và phát hiện đối tượng



Hình 5: Mô hình phát hiện và nhận dạng đối tượng

### 3.2 Bám sát đối tượng với thuật toán Camshift

Thuật toán CamShift [11] (*Continuously Adaptive MeanShift*) được xuất phát và dựa trên nền tảng của thuật toán MeanShift, do vậy việc tính trọng tâm của đối tượng và dịch chuyển cửa sổ tìm kiếm sẽ thực hiện bằng MeanShift [14] [theo vết màu MeanShift - Comaniciu 2003] và Moment ảnh. Ý tưởng chính của Camshift là tính lại trọng tâm của đối tượng dựa vào frame ảnh trước đó và kích thước của cửa sổ tìm kiếm được thay đổi động theo đối tượng (*với MeanShift thì cửa sổ tìm kiếm là cố định*).

#### 3.2.1 Thuật toán MeanShift [3] [20]

**Bước 1:** Khởi tạo vị trí của đích trong frame hiện tại với  $y_0$ , tính mô hình hiện tại đang xét (ứng viên) –  $\{p(y_0)\}_{k=1..n}$

$$\rho[p(y_0), q] = \sum_{k=0}^n \sqrt{p_k(y_0) q_k} \quad (4)$$

Tính:

**Bước 2:** Tính được trọng số  $w_i$  theo

$$w_i = \sum_{k=0}^n \sqrt{\frac{q_k}{p_k(y_0)}} \delta(b(x_i) - k) \quad (5)$$

**Bước 3:** Tìm vị trí đích ứng viên kế tiếp theo

$$y_1 = \frac{\sum_i x_i w_i g\left(\frac{\|y_0 - x_i\|^2}{h}\right)}{\sum_i w_i g\left(\frac{\|y_0 - x_i\|^2}{h}\right)} \quad (6)$$



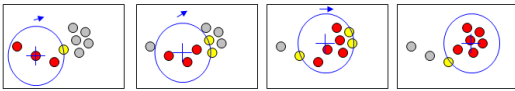
**Bước 4:** Tính ra mô hình mới  $\{p(y_1)\}_{k=1..n}$  và tính được

$$\rho[p(y_1), q] = \sum_{k=0}^n \sqrt{p_k(y_1)q_k} \quad (7)$$

**Bước 5:** Khi (4) < (7) thì ta thực hiện:

$$y_1 = \frac{y_1 + y_0}{2} \quad (8)$$

**Bước 5:** Nếu  $\|y_1 - y_0\| < \varepsilon$  thì cho thuật toán dừng, nếu không nhỏ hơn ta đặt  $y_0 = y_1$  và cho quay lại bước 2 để tìm vị trí.



**Hình 6:** Kỹ thuật dịch chuyển cửa sổ tìm kiếm của MeanShift

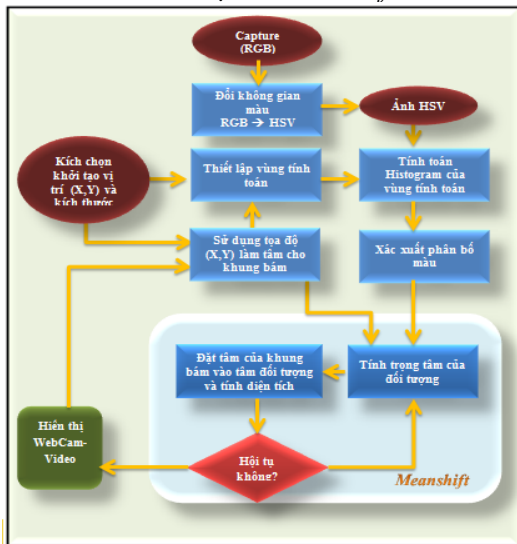
### 3.2.2 Tính Moment ảnh

Những thuộc tính cơ bản của ảnh sẽ được tìm thấy thông qua Moment ảnh [15][11] bao gồm: vùng ảnh (area), trọng tâm của đối tượng và một số thông tin đi kèm với đối tượng đang xét. Đây là việc tính toán những đặc trưng Moment và ước lượng chiều dài, chiều rộng của đối tượng dựa vào

$$\{x_c, y_c\} = \left\{ \frac{M_{10}}{M_{00}}, \frac{M_{01}}{M_{00}} \right\} \quad \text{REF So14} \quad \text{MMercimekKGulezandTVMumc} \setminus h \setminus * \text{MERGEFORMAT} \quad (13)$$

[15] [11]

### 3.2.3 Lưu đồ thuật toán CamShift



**Hình 7:** Lưu đồ hoạt động của giải thuật CamShift [11]

Moment thứ 0.

Moment 2 chiều của bậc (i+j) của một ảnh  $f(x, y)$  có kích thước  $M \times N$  sẽ được định nghĩa như sau [15]:

$$M_{ij} = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} x^i y^j f(x, y) \quad (9)$$

Trong đó  $i, j$  là các số nguyên  $0, 1, 2, 3, \dots$

– Tính Moment thứ 0 [11]:

$$M_{00} = \sum_x \sum_y f(x, y) \quad (10)$$

– Tính Moment thứ nhất của  $x, y$  **Error!**  
Reference source not found.:

$$M_{10} = \sum_x \sum_y x f(x, y) \quad (11)$$

$$M_{01} = \sum_x \sum_y y f(x, y) \quad (12)$$

Một số tính chất cần quan tâm trong moment ảnh là:

- Diện tích (cho ảnh nhị phân) hoặc tổng mức xám (ảnh xám)  $M_{00}$ ;
- Tâm của ảnh (đối tượng):

### 3.3 Tính khoảng cách và di chuyển

#### 3.3.1 Tính khoảng cách của đối tượng đến Camera

$$Z = f \times \frac{X}{x'} \Rightarrow f = \frac{Z \cdot x'}{X} \quad (14)$$

[Nguồn: [10] -chapter 11.1]

- $Z$ : Khoảng cách vật đến Webcam;
- $f$ : tiêu cự (17,63-Webcam);
- $X$ : Kích thước vật (10 cm);
- $x'$ : Kích thước ảnh.

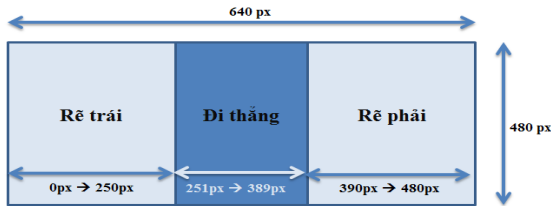
Với 1 pixel ( $X$ ) = 0.0264583333333334 centimeter [cm].

$$Z \text{ (Khoảng cách (cm))} = \frac{17.63 \times X}{\text{Kích thước ảnh (cm)}} \quad (15)$$

$$\text{Vận tốc Robot} = \frac{V_d \times Z}{100} \quad (16)$$

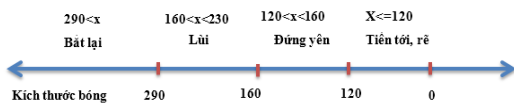
Với:  $V_d$  là vận tốc ban đầu.

### 3.3.2 Robot di chuyển theo mặt phẳng ảnh



**Hình 8: Vị trí của quả bóng trên mặt phẳng ảnh ứng với hoạt động Robot**

Robot chỉ thực hiện việc di chuyển khi quả bóng có kích thước nằm trong khoảng cho phép (Hình 9). Khi kích thước ảnh của quả bóng quá lớn thì sẽ cho thực hiện bắt lại theo ngưỡng.



**Hình 9: Kích thước của quả bóng ứng với hành vi của Robot**

Nếu di chuyển qua vùng có cùng màu sắc với màu sắc mà thuật toán CamShift đang bắt thì hệ thống sẽ cho bắt lại (vì mảng màu lớn  $x > 290px$ ).

## 4 KẾT QUẢ THỰC NGHIỆM

### 4.1 Mô tả thực nghiệm

– Bài báo nghiên cứu trên Robot Pioneer P3-DX của hãng Adept Mobile Robot được trình bày ở phần 2.

– Thông tin máy tính và ứng dụng cài đặt trong nghiên cứu:

Loại	Tên và kích thước sử dụng	Ghi chú
Hiệu máy tính	Laptop HP DV6T-2000;	
Bộ xử lý (CPU)	Intel Core i7 CPU Q720 @1.60Ghz (8 CPUs), 1.6GHz;	
Bộ nhớ (RAM)	4096MB – DDR3;	
Hệ điều hành	Ubuntu 12.04 OS (64 bit);	
Webcam	HP Webcam 1.3 MP;	
Tốc độ Webcam	20 frame/s;	
Thư viện cài đặt	OpenCV 2.4.2[10]; (gói thư viện cài đặt nhận dạng và bám sát)	
Thư viện Robot	SDK ARIA 2.7.2 , MobileSim 0.5;	
Ngôn ngữ LT	C++, biên dịch g++ Linux	

### 4.2 Thiết lập thông số (giá trị) thực nghiệm

- Vận tốc ban đầu của Robot khi di chuyển là  $V_0 = 200 \text{ mm/s}$ ;
- Khoảng cách dừng của Robot trong khoảng  $L_0 = (40; 55) \text{ cm}$ ;
- Khoảng cách Robot hoạt động:  $L_t = 56 \text{ cm}$  trở lên;
- Khoảng cách Robot lùi  $L_b = (28; 40) \text{ cm}$ ;
- Khoảng cách xa nhất mà Robot không thể dò tìm đối tượng là: 333 cm (vì bộ huấn luyện nhận dạng với thông số min là  $20 \times 20 \text{ px} \rightarrow$  tương ứng với kích thước quả bóng);
- Khoảng cách gần nhất mà Robot không thể dò tìm đối tượng là: nhỏ hơn 22 cm (290px);
- Tốc độ ảnh xử lý được thiết lập: 20 frame/s;
- Tiêu cự Webcam được tính trung bình là: 17.6 cm;

- Kích thước giao diện Webcam: 640-480px.

### 4.3 Kết quả thực nghiệm

#### 4.3.1 Huấn luyện nhận dạng và phát hiện đối tượng

- Thông số huấn luyện HaarTraining:

“-nstages 20 -nsplits 2 -minhitrate 0.999-maxfalsealarm 0.5 -npos 1600 -nneg 4200 -w 20 -h 20 -nonsym -mem 1100 -mode ALL”

- **-nstages 20:** Số lượng tầng của mô hình huấn luyện;
- **-nneg 4200:** Số lượng ảnh không có đối tượng (quả bóng) là 4200 ảnh;
- **-npos 1600:** Số lượng ảnh có đối tượng (quả bóng) là 1600 ảnh;
- **-nsplits:** Số lần phân hoạch dữ liệu là 2;
- **-mem 1100:** sử dụng 1.1 GB Ram (bộ nhớ) để training;
- **-minhitrate 0.999:** tỷ lệ nhận dạng đối tượng (quả bóng) tối thiểu là 0.999;
- **-maxfalsealarm 0.5:** tỷ lệ nhận dạng sai đối tượng tối đa là 50%;
- **-mode ALL:** sử dụng tất cả các đặc trưng bao gồm cả đặc trưng mở rộng;
- **-w 20:** kích thước chiều rộng; **-h 20:** kích thước chiều cao.

– Kết quả nhận dạng với độ chính xác (recall) 98% khi kiểm thử với 200 ảnh có đối tượng quả bóng (Dương) và 200 ảnh không có đối tượng quả bóng (Âm):

Dự đoán =>	Dương	Âm
Dương	196	4
Âm	1	199

– Khi sử dụng công cụ **performance** trong OpenCV để ước lượng độ chính xác của mô hình huấn luyện, chúng tôi thu được độ chính xác (*recall*) là **97.4 %**.

#### 4.3.2 Robot di chuyển

Dựa trên mặt phẳng ảnh (mục 3.3.2) và các thông số liên quan (mục 4.2) để điều khiển Robot, kết quả Robot đã di chuyển như sau (phần phụ lục):

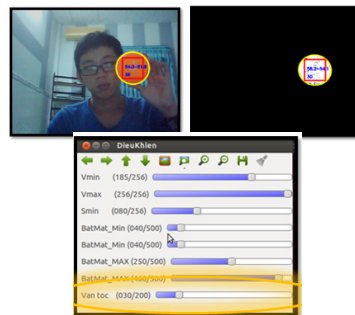
- Robot di chuyển thẳng tới (Hình 10);
- Robot di chuyển rẽ phải (Hình 11);
- Robot di chuyển rẽ trái (Hình 12);
- Robot di chuyển lùi (Hình 13).

### 5 KẾT LUẬN

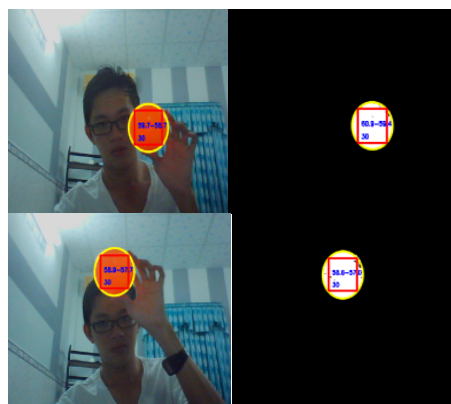
Chúng tôi vừa trình bày sự kết hợp giữa Robot Pioneer P3-DX và máy học để điều khiển các hành vi, di chuyển và bám sát của Robot theo thời gian thực. Robot có khả năng tự di chuyển đúng như yêu cầu đặt ra cho bài toán (*tiến tới, lùi, rẽ trái, rẽ phải*), giảm thiểu khả năng mất bám trong thời gian ngắn. Nhờ sự kết hợp của việc phát hiện đối tượng liên tục khi bị mất bám (*mất quan sát*) đã giúp Robot có thể nhanh chóng tìm lại đối tượng và di chuyển đúng yêu cầu. Robot di chuyển với vận tốc phụ thuộc vào khoảng cách từ đối tượng đến Robot. Tốc độ bắt bám đối tượng dao động 25÷30 ms.

Hướng nghiên cứu tiếp theo của bài báo là việc nâng cao khả năng bám sát đối tượng (*tracking object*) với việc tích hợp thêm bộ lọc Kalman để dự đoán đối tượng trên mặt phẳng ảnh. Robot trong nghiên cứu di chuyển động không theo một bản đồ cụ thể, do đó có thể tích hợp thêm một số bản đồ tĩnh nhằm giúp Robot có thể di chuyển trong một phạm vi nhất định và bám sát theo đối tượng (ví dụ: Robot đá bóng) hoặc Robot sẽ tránh được những chướng ngại vật trước mắt mà không làm ảnh hưởng khả năng bám sát.

Nghiên cứu các ứng dụng đáp ứng nhu cầu thực tế như: Robot chờ hàng theo chủ nhân (*đối tượng nhận dạng là chủ nhân Robot*), Robot làm phương tiện chuyên chở người khuyết tật (*đối tượng được người khuyết tật sử dụng để Robot di chuyển*), Robot lau dọn vệ sinh (*đối tượng là các vết bẩn*), Robot dập lửa trong khu vực nhỏ (*sử dụng kết hợp camera hồng ngoại và nhận dạng khu vực nhiệt độ cao trong ảnh hồng ngoại*).



**Hình 8: Bảng điều khiển và đối tượng được bám sát**



**Hình 9: Bắt bám theo đối tượng quả bóng**

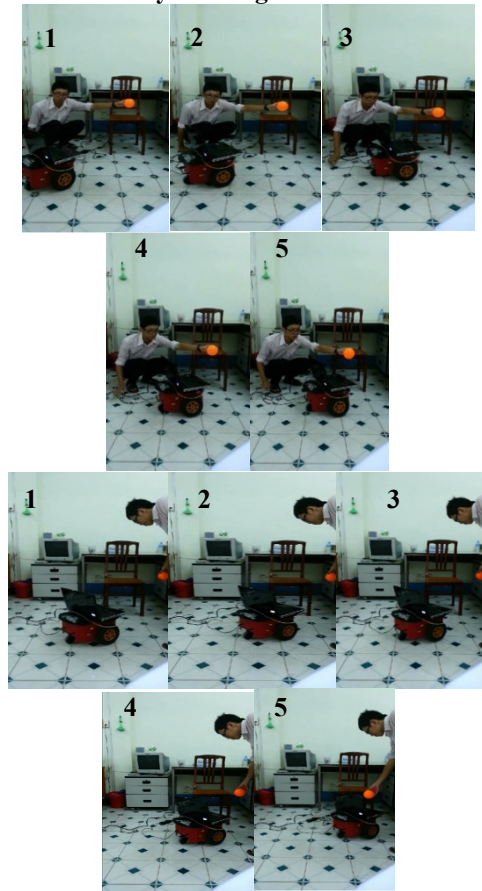
### TÀI LIỆU THAM KHẢO

1. A. Treptow, A. Masselli and A. Zell (2003). *Real-Time Object Tracking for Soccer-Robots without Color Information*. University of Tuebingen Department of Computer Science, Germany.
2. A. WhitBrook (2010). *Programming Mobile Robot with Aria and Player*. Springer.
3. A. Mittal and M. Paragios (2004). *Motion Based Background Subtraction using Adaptive Kernel Density Estimation*. pp. 302-309.
4. Artner (2008). *A Comparison of Mean Shift Tracking Methods*. Digital Media, Upper Austria University of Applied Sciences.
5. C. Zhang, Y. Qiao, E. Fallon, and C. Xu (2009). *An improved camshift algorithm for target tracking in video surveillance*. In Proceedings of 9th. IT and T Conference, number 12.
6. D. Beymer and Konolige K (2001).

- Tracking people from a mobile platform.* In IJCAI-2001 Workshop on Reasoning with Uncertainty in Robotics.
7. D. Comaniciu and V. Ramesh (2009). "Mean Shift and Optimal Prediction for Efficient Object Tracking," Proc. Int'l Conf. Image Processing, vol. III, pp. 70-73, 2000. 20 Sept.
  8. D. Comaniciu, Dorin, Peter Meer (2002). "Mean Shift: A Robust Approach Toward Feature Space Analysis". IEEE Transactions on Pattern Analysis and Machine Intelligence.
  9. D. Kortenkamp, E. Huber, and R. P. Bonasso (1996). *Recognizing and interpreting gestures on a mobile robot.* InProc. of the American Conference on Artificial Intelligence.
  10. G. Bradski and A. Kaehler (2012). *Learning OpenCV*. O'Reilly Media.
  11. Gary R. Bradski (1998). *Computer Vision Face Tracking For Use in a Perceptual User Interface*. Microcomputer Research Lab, Santa Clara, CA, Intel Corporation.
  12. I.J. Cox and S.L (1996). Hingorani. *An efficient implementation of reids multiple hypothesis tracking algorithm and its evaluation for the purpose of visual tracking.* IEEE Transactions on PAMI, 18(2):138–150.
  13. J. MacCormick and A. Blake (1999). *A probabilistic exclusion principle for tracking multiple objects.* InProc. of 7th International Conference on Computer Vision (ICCV), pages 572–587.
  14. K. Fukunaga and L.D. Hostetler (1975). *The Estimation of the Gradient of a Density Function, with Applications in Pattern Recognition.* IEEE Trans. Information Theory, vol. 21, pp. 32-40.
  15. M. Mercimek, K. Gulez and T.V. Mumcu (2005). *Real object recognition using moment invariants.* Yildiz Technical University, Electrical-Electronics Faculty, Electrical Engineering Department, 34349 Besiktas-Istanbul, Turkey.
  16. P. Viola and M. Jones (2001). *Robust Real-time Object Detection.* International Journal of Computer Vision.
  17. Phạm Hồng Ngự (2009). *Nhận dạng đối tượng sử dụng thuật toán Adaboost.* Luận văn Thạc sỹ. Đại học Huế.
  18. R. Lienhart, J. Maydt (2002). *An Extended Set of Haar-like Features for Rapid Object Detection.* Intel Labs, Intel Corporation, Santa Clara, CA 95052, USA.
  19. Y. Freund, and R. Shapire (1995). *A decision-theoretic generalization of on-line learning and an application to boosting.* Proceedings of the Second European Conference on Computational Learning Theory. pp. 23-37.
  20. Y.Cheng (1995). *Mean Shift, Mode Seeking, and Clustering.* IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 17, no. 8. August.

## PHỤ LỤC

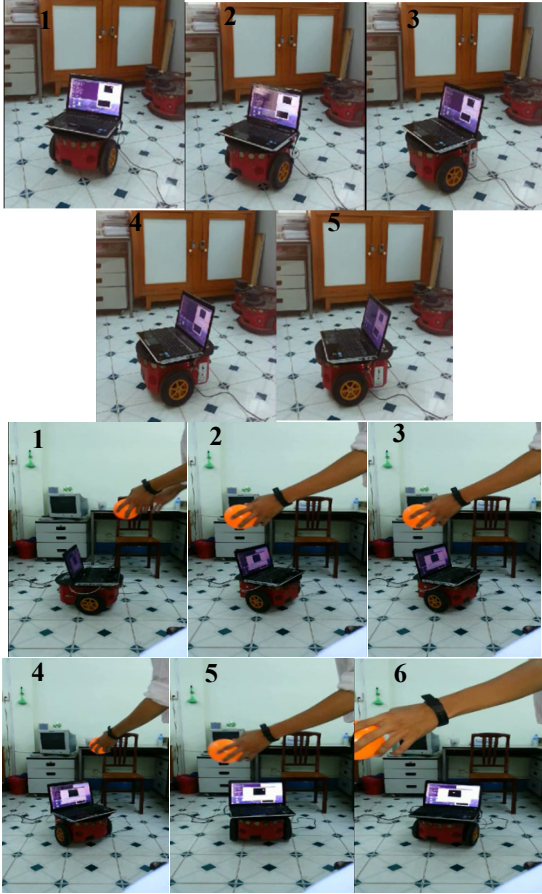
### 1. Robot di chuyển thẳng tới



Hình 10: Robot di chuyển thẳng tới

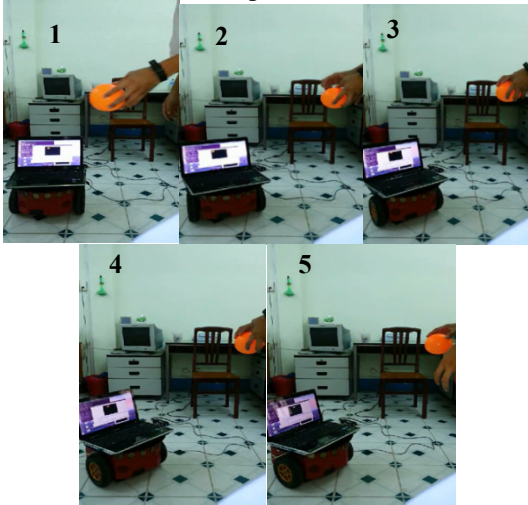


## 2. Robot di chuyển rẽ trái



Hình 12: Robot di chuyển rẽ trái

## 3. Robot di chuyển rẽ phải



Hình 11: Robot di chuyển rẽ phải

## 4. Robot di chuyển lùi



Hình 13: Robot di chuyển lùi