

**ĐẠI HỌC THÁI NGUYÊN**  
**TRƯỜNG ĐẠI HỌC KỸ THUẬT CÔNG NGHIỆP**



**LÊ THỊ THÚY NGÂN**

**ĐIỀU KHIỂN ROBOT BẮM MỤC TIÊU DỰA  
TRÊN XỬ LÝ ẢNH**

**LUẬN VĂN THẠC SĨ KHOA HỌC**  
**NGÀNH KỸ THUẬT ĐIỆN TỬ**

**Thái Nguyên – Năm 2019**

**ĐẠI HỌC THÁI NGUYÊN**  
**TRƯỜNG ĐẠI HỌC KỸ THUẬT CÔNG NGHIỆP**



**LÊ THỊ THÚY NGÂN**

**ĐIỀU KHIỂN ROBOT BẮM MỤC TIÊU DỰA  
TRÊN XỬ LÝ ẢNH**

**LUẬN VĂN THẠC SĨ KHOA HỌC**

**MÃ NGÀNH: 6520203**

**NGÀNH KỸ THUẬT ĐIỆN TỬ**

**NGƯỜI HƯỚNG DẪN KHOA HỌC :**

**PGS.TS LẠI KHẮC LÃI**

**Thái Nguyên – Năm 2019**

**CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM**  
**Độc lập – Tự do – Hạnh phúc**

**BẢN XÁC NHẬN CHỈNH SỬA LUẬN VĂN THẠC SĨ**

**Họ và tên tác giả luận văn: Lê Thị Thúy Ngân**

**Đề tài luận văn: Điều khiển Robot bám mục tiêu dựa trên xử lý ảnh**

**Chuyên ngành: Kỹ thuật Điện tử**

**Mã số:**

Tác giả, Cán bộ hướng dẫn khoa học và Hội đồng chấm luận văn xác nhận tác giả đã sửa chữa, bổ sung luận văn theo biên bản họp Hội đồng ngày 27 tháng 5 năm 2019 với các nội dung sau:

1. Nội dung thứ nhất về việc gộp nội dung chương 2 và chương 3 với nhau.
2. Nội dung thứ hai về việc bổ sung phần giải thích chương trình lập trình trên Arduino.
3. Nội dung thứ ba về việc chỉnh sửa các lỗi soạn thảo các trang 6, 8, 34, 43, 45, 46, 47.

*Thái Nguyên, ngày      tháng      năm 2019*

**Giáo viên hướng dẫn**

**Tác giả luận văn**

**PGS.TS Lại Khắc Lãi**

**Lê Thị Thúy Ngân**

**CHỦ TỊCH HỘI ĐỒNG**

**PGS.TS Nguyễn Thanh Hà**

**LỜI CAM ĐOAN**

Tên tôi là: **Lê Thị Thúy Ngân**

Sinh ngày: 22 tháng 10 năm 1992

Học viên Cao học Khoá 20 – Lớp Kỹ thuật Điện tử - Trường Đại học Kỹ thuật Công nghiệp – Đại học Thái Nguyên.

Xin cam đoan luận văn “**Điều khiển robot bám mục tiêu dựa trên xử lý ảnh**” do thầy giáo **PGS.TS. Lại Khắc Lãi** hướng dẫn là công trình nghiên cứu của riêng tôi. Tất cả các tài liệu tham khảo đều có nguồn gốc, xuất xứ rõ ràng.

Tôi xin cam đoan tất cả những nội dung trong luận văn đúng như nội dung trong đề cương và yêu cầu của thầy giáo hướng dẫn. Nếu có vấn đề gì trong nội dung của luận văn, tôi xin hoàn toàn chịu trách nhiệm với lời cam đoan của mình.

*Thái Nguyên, ngày 20 tháng 4 năm 2019*

**Học viên**

**Lê Thị Thúy Ngân**

## **LỜI CẢM ƠN**

Sau thời gian nghiên cứu, làm việc khẩn trương và được sự hướng dẫn tận tình giúp đỡ của thầy giáo **PGS.TS. Lại Khắc Lãi**, luận văn với đề tài

**“Điều khiển robot bám mục tiêu dựa trên xử lý ảnh”** đã được hoàn thành.

Tôi xin bày tỏ lòng biết ơn sâu sắc tới:

Thầy giáo hướng dẫn **PGS.TS. Lại Khắc Lãi** đã tận tình chỉ dẫn, giúp đỡ tôi hoàn thành luận văn.

Các thầy cô giáo Trường Đại học kỹ thuật công nghiệp - Đại học Thái Nguyên và các bạn bè đồng nghiệp, đã quan tâm động viên, giúp đỡ tôi trong suốt quá trình học tập để hoàn thành luận văn này.

Mặc dù đã cố gắng hết sức, song do điều kiện thời gian và kinh nghiệm thực tế của bản thân còn ít, cho nên đề tài không thể tránh khỏi thiếu sót. Vì vậy, tôi mong nhận được sự đóng góp ý kiến của các thầy giáo, cô giáo và các bạn bè đồng nghiệp.

**Tôi xin chân thành cảm ơn!**

*Thái Nguyên, ngày 20 tháng 4 năm 2019*

**Học viên**

**Lê Thị Thúy Ngân**

## **MỤC LỤC**

<b>LỜI CAM ĐOAN .....</b>	<b>II</b>
<b>LỜI CẢM ƠN .....</b>	<b>III</b>
<b>MỤC LỤC .....</b>	<b>IV</b>
<b>DANH MỤC CÁC CHỮ VIẾT TẮT.....</b>	<b>VI</b>
<b>DANH MỤC CÁC HÌNH VẼ.....</b>	<b>VII</b>
<b>LỜI NÓI ĐẦU .....</b>	<b>1</b>
1. Tính cấp thiết của luận văn.....	1
2. Mục tiêu nghiên cứu .....	2
3. Đối tượng và phạm vi nghiên cứu.....	2
4. Ý nghĩa của luận văn.....	3
5. Nội dung nghiên cứu .....	3
6. Phương pháp và phương pháp luận.....	4
<b>CHƯƠNG 1: TỔNG QUAN VỀ ROBOT BẮM MỤC TIÊU.....</b>	<b>5</b>
1.1. Nghiên cứu tổng quan về một hệ robot có gắn camera bám mục tiêu.....	5
di động .....	5
1.2. Tổng quan tình hình nghiên cứu trong và ngoài nước.....	6
1.2.1 Tình hình nghiên cứu ngoài nước .....	6
1.2.2 Tình hình nghiên cứu trong nước .....	9
1.3. Giải pháp đề xuất .....	10
1.3.1 Hướng tiếp cận phần cứng phục vụ xử lý ảnh.....	10
1.3.2 Hướng tiếp cận phần mềm phục vụ xử lý ảnh.....	12
<b>CHƯƠNG 2: ẢNH SỐ VÀ HIỂN THỊ ẢNH.....</b>	<b>13</b>
2.1. Ảnh số .....	13
2.1.1 Định nghĩa ảnh số.....	13
2.1.2. Thuộc tính của hình ảnh số hóa.....	13
2.1.3. Các loại hình ảnh.....	14
2.2. Hiển thị ảnh.....	16
2.2.1 Chế độ hiển thị .....	16

2.2.2. Phương pháp ánh xạ cho hiển thị hình ảnh 16 bit .....	17
2.2.3. Bảng màu .....	19
<b>CHƯƠNG 3: XỬ LÝ KHỚP MẪU TRÊN PHẦN MỀM LABVIEW .....</b>	<b>22</b>
3.1. Giới thiệu .....	22
3.2. Kỹ thuật ghép mẫu .....	22
3.2.1. Chuẩn hóa chéo tương quan .....	22
3.2.2. Kết hợp tỷ lệ và xoay - bất biến .....	23
3.2.3. Kết hợp kim tự tháp (kết hợp hình chóp) .....	23
3.2.4. Hiệu hình ảnh .....	24
3.3. Thuật toán tìm kiếm .....	24
3.4. Xây dựng, kết nối khối chức năng trên phần mềm LABVIEW .....	26
3.4.1. Giới thiệu phần mềm LABVIEW .....	26
3.4.2. Các khối chức năng .....	27
3.4.3. Kết nối các khối chức năng .....	36
4.1. Mục đích xây dựng mô hình .....	39
4.2. Các thành phần trên mô hình .....	39
4.2.1. Khung mô hình .....	39
4.2.2. Camera .....	39
4.2.3. Vi điều khiển .....	40
4.2.4. Động cơ RC Servo .....	42
4.3. Tính toán góc quay của các servo .....	44
4.4. Chương trình điều khiển trên Arduino Mega 2560 .....	45
4.5. Một số hình ảnh mô hình thực tế .....	47
<b>KẾT LUẬN VÀ KIẾN NGHỊ .....</b>	<b>49</b>
<b>TÀI LIỆU THAM KHẢO .....</b>	<b>50</b>

**DANH MỤC CÁC CHỮ VIẾT TẮT**

Từ Viết Tắt	Tên tiếng anh	Tên tiếng việt
RGB	Red Green Blue	Đỏ Xanh lục Xanh lam
HSL	Hue Saturation Luminance	Màu Độ bão hòa Độ chói
ROI	Regions of Interest	Khu vực quan tâm
VI	Vision	Thị giác
IMAQ	Image Acquisition	Thu nhận ảnh
VISA	Virtual Instrument Software Architecture	Kiến trúc phần mềm ảo



## **DANH MỤC CÁC HÌNH VẼ**

Hình 1.1: Một Số Hệ Thống Tích Hợp Quang Hồng Ngoại, Và Ảnh Nhiệt Trên Các Phương Tiện Di Động .....	8
Hình 1.2: Một Số Hệ Thống Camera Robot Tự Hành, Trong Hàng Không Vũ Trụ.	8
Hình 2.1: Tham Chiếu Không Gian Của Pixel (0, 0).....	13
Hình 3.2: Chuẩn Hóa Tương Quan .....	25
Hình 3.3: Sơ Đồ Kết Nối Các Khối Chức Năng Trên Labview .....	37
Hình 3.4: Giao Diện Người Dùng .....	38
Hình 4.1: Camera C720 Hd Webcam .....	39
Hình 4.2: Bảng Các Thông Số Camera C720 Hd Webcam .....	40
Hình 4.3: Vi Điều Khiển Arduino Mega 2560 R3 .....	40
Hình 4.4: Các Thông Số Của Arduino Mega 2560 R3 .....	41
Hình 4.5: Sơ Đồ Chân Chức Năng Arduino Mega 2560 R3 .....	41
Hình 4.6: Động Cơ Rc Servo .....	42
Hình 4.7: Xung Điều Khiển Servo .....	43
Hình 4.8: Mô Tả Vị Trí Của Đối Tượng Thực .....	44
Hình 4.9: Quy Chiếu Từ Tọa Độ Của Camera Sang Tọa Độ Làm Việc .....	45
Hình 4.10: Một Số Hình Ảnh Mô Hình Thực Tế.....	48

## **LỜI NÓI ĐẦU**

### **1. Tính cấp thiết của luận văn**

Ngày nay việc xây dựng một hệ thống robot bám mục tiêu di động là bài toán được các nhà khoa học rất quan tâm nhằm phục vụ cho con người ở nhiều lĩnh vực khác nhau. Các Robot này thường hoạt động bằng các cảm biến như cảm biến màu sắc, hồng ngoại, cảm biến dò đường, siêu âm... Gần đây các robot loại này được phát triển và tích hợp thêm “thị giác”, việc tích hợp trên Robot hệ thống camera và ứng dụng xử lý ảnh sẽ tăng khả năng tự động, giúp robot thông minh hơn. Các robot có gắn camera và các thiết bị kỹ thuật khác nhằm thực hiện một nhiệm vụ đặt trước, ví dụ như các hệ thống phát hiện lỗi của vật liệu sử dụng camera, các hệ thống dò đường, hệ thống phát hiện lỗi và hàn tự động, vận chuyển hàng hóa trong kho bãi, các hệ phát hiện và bám mục tiêu di động . . . Trong các hệ phát hiện và bám mục tiêu di động, camera được trang bị để có thể nhận biết mục tiêu, kết hợp với các thuật toán xử lý ảnh để xác định chính xác vị trí của mục tiêu để từ đó điều chỉnh cơ cấu chấp hành sao cho tâm của mắt bám đúng mục tiêu. Trong những năm gần đây, các hệ thống bám mục tiêu di động được quan tâm nghiên cứu rất rộng rãi. Luận văn đề cập đến việc nghiên cứu thiết kế và chế tạo robot bám mục tiêu di động trong phòng thí nghiệm với mục đích phục vụ cho công tác nghiên cứu các thuật toán xử lý ảnh và điều khiển hiện đại làm tiền đề cho việc chế tạo các sản phẩm phục vụ đào tạo, y tế, công nghiệp và xa hơn nữa là ứng dụng cho an ninh, quốc phòng. Với sự phát triển của công nghệ điện tử, tốc độ tính toán của vi xử lý năng vượt trội, các chức năng hỗ trợ giao tiếp với các ngoại vi cũng được tích hợp trên một chip, nhờ vậy mà các phương pháp xử lý ảnh hiện đại và các phương pháp điều khiển phức tạp như điều khiển thích nghi bền vững có thể thực thi một cách dễ dàng hơn, chính vì thế các hệ thống bám mục tiêu di động được cải thiện rất nhiều về mặt chất lượng. Vì thế, các nghiên cứu phát triển thuật toán xử lý ảnh hiện đại và các thuật toán điều khiển bám thích nghi, bền vững ngày càng trở nên cấp thiết do khả năng thực hiện các thuật toán này trong thực tế sẽ giúp cho các hệ điều khiển bám mục tiêu ổn định, chính xác và

bền vững hơn khi hệ thống hoạt động trong môi trường thực tế có sự tác động của nhiễu và sự thay đổi mô hình tham số.

Xuất phát từ những yêu cầu và sự cấp thiết đã nêu trên, em đã chọn đề tài cho luận văn của mình là:

### **“Điều khiển Robot bám mục tiêu dựa trên xử lý ảnh”**

## **2. Mục tiêu nghiên cứu**

Mục tiêu của luận văn là tính toán, thiết kế, chế tạo module cảm biến hình ảnh trên cơ sở nhận dạng và xử lý ảnh để điều khiển robot bám mục tiêu di động. Một yếu tố cũng cần chú ý là thuật toán đó phải hướng đến lập trình nhúng trên vi xử lý và chạy thử nghiệm thực tế các thuật toán mới đề xuất, do đó nhiệm vụ là phải thiết kế, chế tạo một robot có gắn camera tự động bám mục tiêu di động trong phòng thí nghiệm để kiểm chứng. Bên cạnh việc tập trung nghiên cứu thuật toán điều khiển mới, việc nghiên cứu và đề xuất thuật toán trong xử lý ảnh để cải thiện chất lượng phát hiện và định vị mục tiêu phải tiến hành song song, bởi tốc độ, độ chính xác tổng thể của toàn hệ thống gắn liền với độ chính xác của “sensor ảnh”.

## **3. Đối tượng và phạm vi nghiên cứu**

### ***Về lý thuyết:***

- + Nghiên cứu tổng quan về robot bám mục tiêu, từ đó rút ra các hướng nghiên cứu thích hợp cho luận văn.
- + Nghiên cứu các thuật toán xử lý ảnh bám bắt mục tiêu di động, đề xuất các phương pháp cải thiện nâng cao chất lượng và tốc độ bám.
- + Thiết kế một cấu trúc phần mềm điều khiển hoàn chỉnh, đồng bộ hóa và có khả năng cài đặt cho robot thực tế để kiểm định các kết quả nghiên cứu lý thuyết.

**Về thực hành:**

+ Thiết kế và chế tạo đồng bộ phần cứng và các thiết bị ngoại vi. Chế tạo các mạch điện tử, điều khiển, giao tiếp ngoại vi với mục tiêu đủ nhanh, mạnh theo hướng có thể mở rộng và nâng cấp.

+ Nghiên cứu, thiết kế cảm biến định vị mục tiêu trên cơ sở công nghệ xử lý ảnh.

+ Thiết kế cấu trúc, lập trình, cài đặt các thuật toán đã nghiên cứu cho robot, chạy thử nghiệm và đánh giá kết quả.

**4. Ý nghĩa của luận văn**

Thiết kế, chế tạo và chạy thử nghiệm thành công robot bám mục tiêu có gắn camera. Sản phẩm này có thể sử dụng trong đào tạo thực tiễn cho sinh viên, các ngành kỹ thuật điện tử, tự động hóa, điều khiển và cơ điện tử . . . hiện đang rất cần tại Việt Nam. Sản phẩm cũng đang được sử dụng để phục vụ đào tạo và nghiên cứu khoa học cho sinh viên ngành kỹ thuật, trường Đại học Kỹ thuật Công nghiệp. Hướng phát triển đến các ứng dụng thực tiễn như chế tạo các robot dò tìm lỗi và hàn tự động, dò đường, robot quan trắc môi trường tại các nơi có điều kiện môi trường độc hại, robot phục vụ chăm sóc y tế...

**5. Nội dung nghiên cứu**

Cấu trúc luận văn bao gồm 4 chương, nội dung tóm tắt của các chương như sau:

**Chương 1: Tổng quan về robot bám mục tiêu**

Tổng quan về hệ thống robot bám mục tiêu có gắn camera tự động bám theo mục tiêu di động , các nghiên cứu liên quan đến lĩnh vực này trong và ngoài nước nhằm định hướng nghiên cứu phát triển cho luận văn, các giải pháp đề xuất để đáp ứng các yêu cầu của luận văn.

**Chương 2: Ảnh số và Hiển thị ảnh**

Trình bày về định nghĩa ảnh số, thuộc tính của hình ảnh số hóa và các loại hình ảnh. Phân tích các chế độ hiển thị hình ảnh, trình bày về các phương pháp ánh xạ cho hiển thị hình ảnh.

**Chương 3: Xử lý khớp mẫu trên phần mềm LABVIEW**

Giới thiệu về kỹ thuật ghép mẫu, các thuật toán tìm kiếm, xây dựng và kết nối các khối chức năng trên phần mềm labview.

#### ***Chương 4: Xây dựng mô hình thực tế***

Thiết kế và chế tạo hệ thống điều khiển camera bám đuổi mục tiêu di động gồm mục đích xây dựng mô hình, các thành phần trên mô hình, xây dựng chương trình điều khiển trên arduino...

Thử nghiệm và đánh giá kết quả thực nghiệm.

#### **6. Phương pháp và phương pháp luận**

- Nghiên cứu lý thuyết để xây dựng thuật toán;
- Tiến hành mô phỏng trên mô hình hệ thống. Đánh giá, so sánh các kết quả lý thuyết, kết quả mô phỏng và thực nghiệm;
- Sử dụng thiết bị thực tế để kiểm chứng.

*Thái Nguyên, ngày 20 tháng 4 năm 2019*

**Học viên**

**Lê Thị Thúy Ngân**

## **CHƯƠNG 1: TỔNG QUAN VỀ ROBOT BẮM MỤC TIÊU**

### **1.1. Nghiên cứu tổng quan về một hệ robot có gắn camera bám mục tiêu di động**

Trong lĩnh vực điều khiển, bài toán điều khiển bám mục tiêu là một trong những bài toán phổ biến nhất. Tùy theo những yêu cầu về tính ổn định và chất lượng điều khiển, ta có nhiều phương án giải quyết khác nhau dựa trên nền tảng lý thuyết điều khiển kinh điển lẫn hiện đại. Hiện nay, để điều khiển bám một đối tượng dịch chuyển có chất lượng tốt, nhiều tác giả đã ứng dụng công nghệ xử lý ảnh trên nền tảng sử dụng camera để theo dõi đối tượng. Đối tượng bám rất đa dạng: có thể là người, xe, bóng, hoặc tùy theo yêu cầu thực tế. Vì vậy, bài toán đầu tiên trong hệ thống điều khiển bám đối tượng dịch chuyển sử dụng camera là phải nhận dạng được đối tượng dịch chuyển trên cơ sở các ảnh phản hồi từ camera, sử dụng các giải thuật xử lý ảnh, giải thuật xử lý bám đối tượng trên nền các ảnh thu được đưa ra tọa độ, vận tốc dịch chuyển, hướng dịch chuyển hiện thời của đối tượng trên nền ảnh và cuối cùng là áp dụng các giải thuật điều khiển động cơ để điều khiển cơ cấu mang camera bám theo đối tượng thực. Trong thực tế, đã có rất nhiều tác giả nghiên cứu chế tạo các mô hình điều khiển hệ thống mang camera bám theo đối tượng dịch chuyển. Chẳng hạn như hệ robot xe - camera, cánh tay máy mang camera, nòng súng mang camera, hệ pan - tilt camera... có thể xoay nhiều hướng. Mục tiêu của mô hình này là điều khiển hai động cơ DC servo hoặc động cơ bước để đảm bảo tầm quan sát và hướng quan sát của camera theo dõi và bám theo đối tượng dịch chuyển

Việc xây dựng hệ thống điều khiển bám mục tiêu di động sử dụng thị giác máy tính không chỉ là nền tảng tri thức cho các nghiên cứu sau này, mà còn có thể ứng dụng cụ thể trong ngành điều khiển robot bám mục tiêu theo những quỹ đạo mong muốn để thực hiện một số nhiệm vụ cụ thể, hoặc có thể ứng dụng trong lĩnh vực quốc phòng như xây dựng hệ thống ngắm bắn tự động bám mục tiêu trên các loại pháo phòng không...

## **1.2. Tổng quan tình hình nghiên cứu trong và ngoài nước**

### **1.2.1 Tình hình nghiên cứu ngoài nước**

Trên thế giới nghiên cứu ứng dụng xử lý và nhận dạng ảnh (Machine Vision) đang là hướng nghiên cứu tập trung của rất nhiều nhà khoa học trong đa số các lĩnh vực. Xử lý ảnh số đã được phát triển và trở thành một lĩnh vực khoa học. Xử lý ảnh số không chỉ nâng cao chất lượng của ảnh mà còn phân tích và lý giải nó phục vụ các mục đích riêng biệt. Các thiết bị ngày nay được ứng dụng công nghệ xử lý và điều khiển theo hình ảnh ngày càng nhiều và cho thấy rõ sự ưu việt của nó, trong đó có rất nhiều ứng dụng mang tính cách mạng như:

*Trong quân sự:*

Các hệ thống tích hợp quang hồng ngoại có khả năng tự động điều khiển dàn hỏa lực (pháo, tên lửa) được lắp đặt cho các trận địa cao xạ, trên xe tăng, tàu chiến, máy bay, tên lửa hoặc vệ tinh (Hình 1). Chúng được thay thế và hỗ trợ các dàn rada dễ bị nhiễu trong việc tự động phát hiện, cảnh giới, bám bắt mục tiêu. Đặc biệt có những loại lắp trên máy bay có khả năng điều khiển hỏa lực đánh phá hàng chục mục tiêu một lúc. Ngoài ra còn phải kể đến các đầu tự dẫn tên lửa và đạn thông minh

*Trong an ninh, phòng chống tội phạm, bảo vệ pháp luật:*

Các hệ thống camera nhận dạng khuôn mặt vân tay tự động cũng như phát hiện, theo dõi, cảnh báo các âm mưu và hoạt động khủng bố. Các xe robot tự hành có gắn các camera cũng được ứng dụng trong các môi trường độc hại, dò phá bom mìn (Hình 1.1).

*Trong lĩnh vực hàng không vũ trụ:*

Các hệ thống ống kính chụp ảnh viễn thám (remote sensing) lắp trên các vệ tinh bay quanh trái đất có thể chụp và quan sát được các vật kích cỡ 0,5 m từ độ cao 750 km trong mọi điều kiện thời tiết (Hình 1.1). Việc nối ghép các ống kính này với hệ thống GPS sẽ cho phép xây dựng các bản đồ số có những lĩnh vực ứng dụng cực kỳ quan trọng trong quốc phòng an ninh, phát triển kinh tế, xã hội...

*Trong công nghiệp, giao thông, xây dựng :*



Hệ thống quang điện tử đóng vai trò của các thị giác máy (machine vision) có khả năng tự động đo đạc kiểm tra chất lượng sản phẩm trong các dây chuyền sản xuất: phân loại hạt ngũ cốc, cà phê; tìm lỗi lắp ráp linh kiện các bản vi mạch và khuyết tật các mối hàn và động cơ... Các hệ thống quang điện tử cũng được ứng dụng ngày càng nhiều trong giao thông như đo tốc độ, tự động kiểm soát điều khiển và phân luồng giao thông (Hình 1.2).

*Trong nghiên cứu y sinh được học:*

Các kính hiển vi có khả năng tự động nhận dạng và đo đếm các tế bào với độ chính xác cao. Các kính hiển vi có hệ thống dẫn đường laser cho phép thực hiện những phẫu thuật rất phức tạp như mổ u não, nơi mà một sự không chính xác cỡ  $\mu\text{m}$  cũng gây tổn hại đến các dây thần kinh chằng chịt xung quanh...

*Trong công nghiệp giải trí truyền hình:*

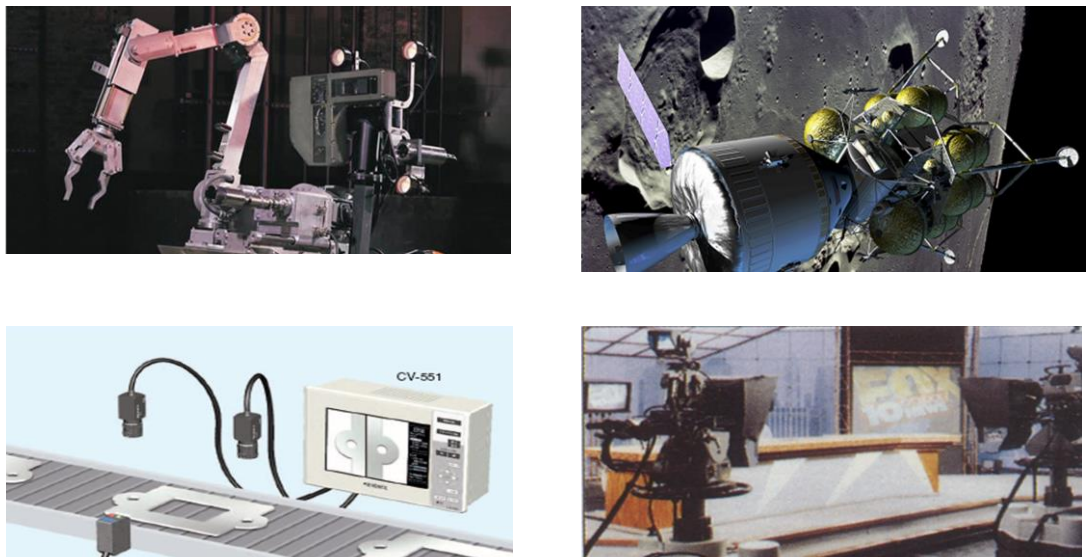
Các hệ thống tích hợp có thể điều khiển các camera kích thước và khối lượng lớn dễ dàng tự động bám theo các đối tượng chuyển động nhanh như bóng đang bay, đua xe...

Một số hình ảnh hệ thống sử dụng công nghệ xử lý ảnh:





*Hình 1.1: Một số hệ thống tích hợp quang hồng ngoại, và ảnh nhiệt trên các phương tiện di động*



*Hình 1.2: Một số hệ thống camera robot tự hành, trong hàng không vũ trụ*

Nhìn chung, các bộ điều khiển này rất đa dạng về chủng loại, mục đích sử dụng, hình dáng, kích thước và tính năng, có thể rút ra một số đặc điểm chính của các bộ điều khiển đó như sau:

- Về phần cứng:

+ Hình thức: rất đa dạng, gọn, chắc chắn, dễ lắp đặt tùy từng loại ứng dụng, tầm quan sát, và đảm bảo các yêu cầu về an ninh, quân sự: kín khít, chống ăn mòn, chịu được thời tiết khắc nghiệt như mưa, gió, khí hậu biển . . .

+ Mạch in được thiết kế nhiều lớp, được nhiệt đới hóa để bảo đảm hoạt động an toàn tin cậy, có khả năng chống nhiễu;

+ Các bộ điều khiển sử dụng nguồn điện áp một chiều 24V hoặc 12V. Nguồn điện luôn luôn sử dụng các mạch bảo vệ quá điện áp quá dòng;

+ Vi xử lý và tất cả các linh kiện điện tử đều là loại đặc chủng dành cho quân sự.

- *Về phần mềm:*

Đa số đều giống nhau, có chức năng chính như sau:

+ Đọc ảnh, video từ camera truyền về, hiển thị, lưu trữ ảnh thu được, truyền hình link qua cáp, mạng truyền thông không dây, Wifi . . .

+ Có các chức năng điều khiển hệ để xoay, các thiết bị ngoại vi;

+ Giao tiếp với Module Joystick để điều khiển;

+ Hiển thị vị trí, vận tốc và góc quay;

+ Có khả năng bám mục tiêu tự động.

Tác giả Comaniciu, Ramesh và Meer có các công trình nghiên cứu được đăng trên IEEE đã sử dụng thuật toán Meanshift và bộ lọc Kalman để dự báo tối ưu hướng dịch chuyển của đối tượng quan sát để xây dựng hệ thống bám theo đối tượng. Tuy nhiên, kết quả cũng chỉ dừng lại ở tốc độ bám ảnh còn chậm do thời gian xử lý các thuật toán trên máy tính và độ phân giải của camera khi đọc dữ liệu về máy tính.

### **1.2.2 Tình hình nghiên cứu trong nước**

Từ nhiều năm nay, một số cơ quan nghiên cứu chủ yếu thuộc Bộ Quốc phòng đã đặt vấn đề nghiên cứu xây dựng hệ thống, có thể kể ra:

- Viện Nghiên cứu Tự động hóa Quân sự đã nghiên cứu và chế tạo thành công hệ quang điện tử tích hợp ứng dụng cho hỏa lực tên lửa phòng không không quân. Đây là Đề tài đã đoạt giải thưởng Hồ Chí Minh về Khoa học công nghệ. Đây cũng là công trình mang tính chất tuyệt mật;

- Viện Vũ khí và Phân Viện Vật lý Kỹ thuật thuộc Viện Khoa học Công nghệ và Kỹ thuật Quân sự - Bộ Quốc phòng cũng đã nghiên cứu triển khai hệ tích hợp ứng dụng cho pháo phòng không;

- Viện Nghiên cứu ứng dụng công nghệ Bộ Khoa học và Công nghệ đã đặt vấn đề nghiên cứu chế tạo hệ thống quang điện tử tích hợp tự động bám mục tiêu dựa trên

hình ảnh và đã đạt được một số kết quả nhất định, và đây cũng là hệ thống ứng dụng cho Phòng không không quân, tầm quan sát xa;

Đánh giá chung là đa số các hệ thống này đều là các sản phẩm phục vụ bám mục tiêu bay, là các hệ quang điện tử tích hợp đặt cố định, phục vụ cho phòng không, có tầm quan sát bám bắt xa ( $>2$  km). Mặc dù các nghiên cứu đã có các kết quả đã được khẳng định, tuy nhiên vì đây là một hệ thống đòi hỏi sự kết hợp của rất nhiều lĩnh vực khoa học công nghệ cao và đặc biệt là lĩnh vực ứng dụng cho quốc phòng do đó các thông tin là rất hạn chế.

- Ngoài ra có tác giả Ngô Mạnh Tiến (Viện Vật lý, Viện KH & CN Việt Nam) đã sử dụng thuật toán bám theo một tập hợp các điểm đặc trưng ảnh của Lucas, Kanade và Tommasini KLT, đây là phương pháp bám sử dụng rất có hiệu quả đối với ảnh thu được là ảnh đen trắng, ảnh của camera hồng ngoại nhìn đêm. Thuật toán này dùng để tính toán sai lệch vị trí mục tiêu, bao gồm các bước: tiền xử lý, xác định và bám các đặc tính (sử dụng 100 đặc tính), bù tự chuyển động của camera, sai phân ảnh, định nghĩa và phân đoạn chuyển động, trước lượng vị trí của đối tượng, xác định LOS và bù vào vòng bám, xây dựng cửa sổ và hiển thị ảnh. Đối với các đối tượng thu được hình ảnh từ camera là ảnh màu đặc trưng so với nền ảnh, tác giả sử dụng thuật toán Camshift (Continuously Adaptive Meanshift), thuật toán này dựa trên cơ sở của thuật toán MeanShift, nó gồm các bước: Chọn vị trí ban đầu của Search Window, dùng thuật toán Meanshift lu ra một vùng moment thứ 0, đặt kích cỡ của cửa sổ tìm kiếm bằng một hàm của Oment thứ 0 tìm được trong bước 2, lặp lại bước và bước 3 cho đến khi hội tụ.

### **1.3. Giải pháp đề xuất**

#### **1.3.1 Hướng tiếp cận phần cứng phục vụ xử lý ảnh**

##### *1.3.1.1 Các giải pháp trong thực tế*

##### **❖ Máy tính PC, laptop**

Đây là một trong những phương pháp đơn giản nhất. Có thể tận dụng các mainboard máy tính hay thậm chí các máy tính xách tay với chức năng là một đơn vị xử lý ảnh, và đưa ra quyết định. Với việc kết nối một camera hay webcam ta hoàn

toàn chủ động trong quá trình nhận/xử lý ảnh. Các giao tiếp ngoại vi phổ biến như UART, Parallel, USB hay Keyboard. Việc sử dụng PC, laptop sẽ có những ưu nhược điểm sau:

- Ưu điểm:

+ Rất dễ dàng phát triển các ứng dụng dựa trên các phần mềm lập trình như C, VisualC, VisualBasic . . . , rất phù hợp cho những người mới bắt đầu nghiên cứu về lĩnh vực xử lý ảnh đặc biệt là hiện nay có rất nhiều thư viện mở phục vụ cho xử lý ảnh, do đó rất thuận tiện cho người mới bắt đầu tìm hiểu về xử lý ảnh;

+ Dễ dàng lập trình, kiểm tra lỗi;

+ Các công cụ lập trình/biên dịch phổ biến (C, C++, VisualC...)

- Nhược điểm:

+ Kích thước, khối lượng lớn;

+ Dễ hư hỏng do va đập hay các tác nhân khác;

+ Chỉ có thể giao tiếp với ngoại vi thông qua các chuẩn phổ biến như UART, USB.

❖ Main công nghiệp, máy tính nhúng PC 104

Ta có thể sử dụng mainboard PC104 với các chức năng như một máy tính thông thường. Việc sử dụng PC104 sẽ có những ưu nhược điểm sau:

- Ưu điểm:

+ Có tốc độ xử lý cao;

+ Dễ Dàng lập trình, kiểm lỗi;

+ Hệ điều hành quen thuộc (windows / linux);

+ Các công cụ lập trình / biên dịch phổ biến (MSVC, C + + ...).

- Nhược điểm:

+ Kích thước lớn;

+ Có quá nhiều thành phần không sử dụng đến. Giá thành đắt ( > 300 \$ );

+ Chỉ có thể giao tiếp với ngoại vi thông qua các chuẩn phổ biến như UART, USB.

❖ FPGA

Đây là giải pháp về phần cứng mà hay được sử dụng hiện nay. Tận dụng đặc tính của FPGA là linh hoạt và tốc độ xử lý nhanh. Ta có thể kết nối FPGA với các CCD sensor để điều khiển và nhận các ảnh thông qua bus dữ liệu tốc độ cao.

#### *1.3.1.2 Giải pháp phần cứng sử dụng trong đề tài*

Phần cứng được sử dụng trong đề tài bao gồm một máy tính, thị giác máy là camera, sử dụng vi điều khiển, 2 động cơ RC servo, khung mô hình, bộ truyền dữ liệu RS232 với đèn phát Laser.

#### **1.3.2 Hướng tiếp cận phần mềm phục vụ xử lý ảnh**

Để giải quyết bài toán xử lý ảnh – thị giác máy tính nói chung và bài toán điều khiển bám đối tượng nói riêng, trong thực tế ta hay dùng các thư viện mã nguồn mở trên cơ sở các thuật toán được tích hợp sẵn trong các hàm cơ sở. Thông qua các khối chức năng, ghép nối và đưa ra thông số hợp lý cho từng khối trên các phần mềm chuyên dụng. Phần mềm được sử dụng trong đề tài là phần mềm Labview. Chương trình điều khiển ngoại vi được lập trình trên arduino mega 2560.

## CHƯƠNG 2: ẢNH SỐ VÀ HIỂN THỊ ẢNH

### 2.1. Ảnh số

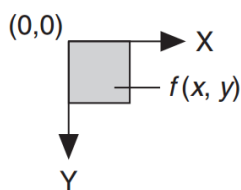
#### 2.1.1 Định nghĩa ảnh số

Một hình ảnh là một mảng 2D (2 chiều) của các giá trị đại diện cho cường độ ánh sáng. Với mục đích xử lý hình ảnh, thuật ngữ hình ảnh đề cập đến một hình ảnh kỹ thuật số.

Một hình ảnh là một hàm của cường độ ánh sáng:

$$f(x, y)$$

Trong đó  $f$  là độ sáng của điểm  $(x, y)$  và  $x$  và  $y$  đại diện cho không gian tọa độ của một yếu tố hình ảnh, hoặc pixel. Theo quy ước, tham chiếu không gian của pixel với tọa độ  $(0, 0)$  nằm ở trên cùng, góc trái của hình ảnh. Lưu ý trong hình 2.1 rằng giá trị của  $x$  tăng khi di chuyển từ trái sang phải và giá trị của  $y$  tăng từ trên xuống dưới.



Hình 2.1: Tham chiếu không gian của Pixel  $(0, 0)$

Trong xử lý hình ảnh kỹ thuật số, một cảm biến hình ảnh chuyển đổi một hình ảnh thành một số lượng pixel rời rạc. Cảm biến hình ảnh gán cho từng pixel một vị trí số và mức màu xám hoặc giá trị màu chỉ định độ sáng hoặc màu của pixel.

#### 2.1.2. Thuộc tính của hình ảnh số hóa

Một hình ảnh số hóa có ba thuộc tính cơ bản: độ phân giải (resolution), định nghĩa (definition) và số lượng mặt phẳng (number of planes).

##### 2.1.2.1. Độ phân giải

Độ phân giải không gian của hình ảnh được xác định bởi số lượng hàng của nó và các cột pixel. Một hình ảnh bao gồm  $m$  cột và  $n$  hàng có độ phân giải của  $m \times n$ . Hình ảnh này có  $m$  pixel dọc theo trục ngang của nó và  $n$  pixel dọc theo trục dọc của nó.

#### **2.1.2.2. Định nghĩa**

Định nghĩa của một hình ảnh cho biết số lượng sắc thái mà ta có thể thấy trong hình ảnh. Độ sâu bit của hình ảnh là số bit được sử dụng để mã hóa giá trị của một pixel. Đối với độ sâu bit đã cho của  $n$ , hình ảnh có định nghĩa  $2^n$ , nghĩa là một pixel có thể có  $2^n$  giá trị khác nhau. Ví dụ, nếu  $n$  bằng 8 bit, một pixel có thể có 256 giá trị khác nhau từ 0 đến 255. Nếu  $n$  bằng 16 bit, một pixel có thể có 65.536 giá trị khác nhau trong khoảng từ 0 đến 65.535 hoặc từ Nhận -32.768 đến 32.767.

#### **2.1.2.3. Số lượng mặt phẳng màu**

Số lượng mặt phẳng trong một hình ảnh tương ứng với số lượng mảng pixel tạo thành hình ảnh. Một hình ảnh thang độ xám hoặc giả màu gồm một mặt phẳng. Một hình ảnh màu sắc thật bao gồm ba mặt phẳng: một cho mỗi thành phần màu đỏ, thành phần màu xanh lục và thành phần xanh dương. Trong ảnh màu trung thực, cường độ thành phần màu của pixel được mã hóa thành ba giá trị khác nhau. Một hình ảnh màu là sự kết hợp của ba mảng pixel tương ứng với các thành phần màu đỏ, xanh lục và xanh lam trong hình ảnh RGB. Hình ảnh HSL được xác định bởi màu sắc, độ bão hòa và độ chói của chúng.

#### **2.1.3. Các loại hình ảnh**

##### **2.1.3.1. Hình ảnh thang độ xám**

Một hình ảnh thang độ xám bao gồm một mặt phẳng các pixel. Mỗi pixel là được mã hóa bằng một trong các kiểu sau: Số nguyên không dấu 8 bit biểu thị các giá trị thang độ xám giữa 0 và 255 Số nguyên có chữ ký 16 bit biểu thị các giá trị thang độ xám giữa -32.768 và +32.767. Số dấu phẩy động được mã hóa bằng bốn byte, đại diện cho các giá trị thang độ xám từ phạm vi  $-\infty$  đến  $\infty$ .

##### **2.1.3.2. Ảnh màu**

Một hình ảnh màu được mã hóa trong bộ nhớ dưới dạng đỏ, lục và lam (RGB) hoặc hình ảnh chứa màu sắc, độ bão hòa và độ chói (HSL). Pixel hình ảnh màu là tổng hợp của bốn giá trị. Hình ảnh RGB lưu trữ thông tin màu bằng cách sử dụng Mỗi bit 8 bit cho các mặt phẳng đỏ, lục và lam. Màu sắc lưu trữ hình ảnh HSL thông tin sử dụng mỗi 8 bit cho màu sắc, độ bão hòa và độ chói. Hình ảnh RGB U64 lưu trữ



thông tin màu bằng cách sử dụng 16 bit cho mỗi màu đỏ, xanh lá cây, và các mặt phẳng màu xanh. Trong các mô hình màu RGB và HSL, thêm 8 bit giá trị không được sử dụng. Đại diện này được gọi là  $4 \times 8$  bit hoặc 32 bit mã hóa. Trong mô hình màu RGB U64, giá trị 16 bit bổ sung sẽ tang không sử dụng Đại diện này được gọi là mã hóa  $4 \times 16$  bit hoặc 64 bit.

### **2.1.3.3. Hình ảnh phức tạp**

Một hình ảnh phức tạp chứa thông tin tần số của hình ảnh thang độ xám. ta có thể tạo một hình ảnh phức tạp bằng cách áp dụng biến đổi Fast Fourier (FFT) đến một hình ảnh thang độ xám. Sau khi ta chuyển đổi hình ảnh thang độ xám thành một hình ảnh phức tạp, ta có thể thực hiện các hoạt động miền tần số trên hình ảnh. Mỗi pixel trong một hình ảnh phức tạp được mã hóa thành hai độ chính xác co giá trị kiểu dấu phẩy động, đại diện cho các thành phần thực và ảo của pixel phức tạp. ta có thể trích xuất bốn thành phần sau từ một hình ảnh phức tạp: phần thực, phần ảo, cường độ và pha.

### **2.1.3.4. File ảnh**

Một tệp hình ảnh bao gồm một tiêu đề theo sau là các giá trị pixel. Tùy trên định dạng tệp, tiêu đề chứa thông tin hình ảnh về độ phân giải ngang và dọc, định nghĩa pixel và bảng màu gốc. Tệp hình ảnh cũng có thể lưu trữ thông tin về hiệu chuẩn, khớp mẫu mẫu và lớp phủ. Sau đây là các định dạng tệp hình ảnh phổ biến:

- Bitmap (BMP)
- Định dạng tệp hình ảnh được gắn thẻ (TIFF)
- Đồ họa mạng di động (PNG) cung cấp khả năng lưu trữ thông tin hình ảnh về hiệu chuẩn không gian, khớp mẫu mẫu và lớp phủ
- Định dạng nhóm ảnh chụp chung (JPEG)
- Định dạng tệp hình ảnh bên trong của National (AIPD) lưu hình ảnh dấu phẩy động, phức tạp và HSL .

Các định dạng tiêu chuẩn cho hình ảnh màu xám 8 bit và RGB là BMP, TIFF, PNG, JPEG và AIPD. Các định dạng chuẩn cho thang độ xám 16 bit, 64 bit RGB và hình ảnh phức tạp là PNG và AIPD.



## **2.2. Hiển thị ảnh**

Hiển thị ảnh là một thành phần quan trọng của ứng dụng thị giác bởi vì nó cung cấp cho bạn khả năng trực quan hóa dữ liệu của bạn. Đang xử lý hình ảnh và hình ảnh trực quan là các yếu tố riêng biệt và riêng biệt. Hình ảnh xử lý đề cập đến việc tạo, thu nhận và phân tích hình ảnh. Hình ảnh trực quan đề cập đến cách trình bày dữ liệu hình ảnh và cách bạn có thể tương tác với các hình ảnh trực quan. Một ứng dụng hình ảnh điển hình sử dụng nhiều hình ảnh trong bộ nhớ mà ứng dụng không bao giờ hiển thị.

Sử dụng các chức năng hiển thị để trực quan hóa dữ liệu hình ảnh của bạn, truy xuất được tạo sự kiện và dữ liệu liên quan từ môi trường hiển thị hình ảnh, chọn ROI từ một hình ảnh tương tác và chú thích hình ảnh với bổ sung thông tin.

Tùy thuộc vào môi trường phát triển ứng dụng của bạn, bạn có thể hiển thị hình ảnh trong các môi trường hiển thị hình ảnh sau: cửa sổ bên ngoài (LabVIEW và LabWindows/CVI), điều khiển hiển thị hình ảnh LabVIEW (LabVIEW 7.0 trở lên) và điều khiển ActiveX CWIMAQViewer (Ngôn ngữ lập trình). Hiển thị chức năng hiển thị hình ảnh, đặt thuộc tính của môi trường hiển thị hình ảnh, gán bảng màu cho hiển thị hình ảnh môi trường, đóng môi trường hiển thị hình ảnh, và thiết lập và sử dụng trình duyệt hình ảnh trong môi trường hiển thị hình ảnh. Một số chức năng ROI tập hợp con của các chức năng hiển thị bộ ba tương tác xác định ROI trong hiển thị hình ảnh môi trường. Các hàm ROI này cấu hình và hiển thị khác nhau công cụ vẽ, phát hiện sự kiện vẽ, lấy thông tin về khu vực được vẽ trên môi trường hiển thị hình ảnh, và di chuyển và xoay ROI. Lớp phủ không phá hủy hiển thị thông tin quan trọng trên đầu hình ảnh mà không thay đổi giá trị của các pixel hình ảnh.

### **2.2.1 Chế độ hiển thị**

Một trong những thành phần quan trọng của việc hiển thị hình ảnh là chế độ hiển thị bộ điều hợp video. Chế độ hiển thị cho biết có bao nhiêu bit chỉ định màu của một pixel trên màn hình hiển thị. Nói chung, màn hình chế độ có sẵn từ bộ điều hợp video dao động từ 8 bit đến 32 bit mỗi pixel, tùy thuộc vào dung lượng bộ nhớ video

có sẵn trên bộ điều hợp video và độ phân giải màn hình ta chọn. Nếu ta có chế độ hiển thị 8 bit, pixel có thể là một trong 256 màu khác nhau. Nếu ta có chế độ hiển thị 16 bit, pixel có thể là một trong 65.536 màu. Ở chế độ hiển thị 24 bit hoặc 32 bit, màu của pixel trên màn hình là được mã hóa bằng cách sử dụng 3 hoặc 4 byte tương ứng. Trong các chế độ này, thông tin là được lưu trữ bằng cách sử dụng 8 bit cho mỗi thành phần màu đỏ, xanh lục và xanh lam của pixel. Các chế độ này cung cấp khả năng hiển thị khoảng 16.7 triệu màu. Hiệu chế độ hiển thị là rất quan trọng, xử lý hình ảnh chức năng thường sử dụng hình ảnh thang độ xám. Bởi vì pixel màn hình hiển thị là được làm bằng các thành phần màu đỏ, xanh lá cây và xanh dương, các pixel của hình ảnh thang độ xám không thể được kết xuất trực tiếp. Ở chế độ hiển thị 24 bit hoặc 32 bit, bộ điều hợp hiển thị sử dụng 8 bit để mã hóa một giá trị thang độ xám, cung cấp 256 màu xám. Độ phân giải màu này là đủ để hiển thị hình ảnh thang độ xám 8 bit. Tuy nhiên, độ sâu bit cao hơn hình ảnh, chẳng hạn như hình ảnh thang độ xám 16 bit, không được thể hiện chính xác trong chế độ hiển thị 24 bit hoặc 32 bit. Để hiển thị hình ảnh thang độ 16 bit, một trong hai bỏ qua các bit quan trọng nhất hoặc sử dụng chức năng ánh xạ để chuyển đổi 16 bit đến 8 bit.

### **2.2.2. Phương pháp ánh xạ cho hiển thị hình ảnh 16 bit**

Các kỹ thuật sau đây mô tả cách chuyển đổi hình ảnh 16 bit sang hình ảnh 8 bit và hiển thị chúng bằng các chức năng ánh xạ. Lập bản đồ các chức năng phân phối đồng đều dải động của hình ảnh 16 bit đến một hình ảnh 8 bit.

- Full Dynamic Combo Giá trị cường độ tối thiểu của hình ảnh 16 bit là ánh xạ thành 0 và giá trị cường độ tối đa được ánh xạ tới 255. Tất cả các giá trị khác trong hình ảnh được ánh xạ từ 0 đến 255 bằng cách sử dụng phương trình hiển thị dưới đây. Phương pháp ánh xạ này là mục đích chung bởi vì nó đảm bảo hiển thị phạm vi động hoàn chỉnh của hình ảnh. Bởi vì các giá trị pixel tối thiểu và tối đa trong một hình ảnh được sử dụng để xác định toàn bộ dải động của hình ảnh đó, sự hiện diện của các pixel nhiễu hoặc khiếm khuyết (đối với các cảm biến không thuộc loại A) với giá trị tối thiểu hoặc tối đa có thể ảnh hưởng đến sự xuất hiện của hình ảnh hiển thị.

$$z = \frac{x - y}{v - y} \times 255$$

z là giá trị pixel 8 bit  
x là giá trị 16 bit  
y là giá trị cường độ tối thiểu  
v là giá trị cường độ tối đa

90% cường độ tương ứng với 5% tích lũy biểu đồ được ánh xạ đến 0, cường độ tương ứng với 95% biểu đồ tích lũy được ánh xạ tới 255. Các giá trị trong phạm vi 0 đến 5% là ánh xạ thành 0, trong khi các giá trị trong phạm vi 95 đến 100% được ánh xạ tới 255. Phương pháp ánh xạ này mạnh hơn phương thức động đầy đủ và không nhạy cảm với quang sai nhỏ trong ảnh. Phương pháp này đòi hỏi tính toán của biểu đồ tích lũy hoặc ước tính của biểu đồ.

- Đưa ra tỷ lệ phần trăm phạm vi phương thức này tương tự với 90% cường độ động tương ứng, ngoại trừ tỷ lệ phần trăm tối thiểu và tối đa của biểu đồ tích lũy mà phần mềm ánh xạ tới 8 bit được người dùng xác định.

- Kỹ thuật này tương tự như phương pháp Full Dynamic, ngoại trừ các giá trị tối thiểu và tối đa được ánh xạ tới 0 và 255 được xác định bởi người dùng. Bạn có thể sử dụng phương pháp này để tăng cường độ tương phản của một số vùng của hình ảnh bằng cách tìm mức tối thiểu và giá trị tối đa của các vùng đó và tính toán biểu đồ của những vùng đó. Một biểu đồ của khu vực này cho thấy tối thiểu và cường độ tối đa của các pixel. Những giá trị này được sử dụng để kéo dài phạm vi năng động của toàn bộ hình ảnh.

- Downshifts kỹ thuật tinh tế này dựa trên sự dịch chuyển của các giá trị pixel. Điều này phương pháp áp dụng một số lần dịch chuyển phải cho giá trị pixel 16 bit và hiển thị bit ít quan trọng nhất. Kỹ thuật này cắt một số các bit thấp nhất, không được hiển thị. Phương pháp này rất nhanh, nhưng nó làm giảm tính năng động thực sự của cảm biến xuống khả năng cảm biến 8 bit. Nó đòi hỏi kiến thức về độ sâu bit của cảm biến hình ảnh có đã được sử dụng. Ví dụ: hình ảnh thu được bằng máy ảnh 12 bit nên được hình dung bằng cách sử dụng bốn ca phải để hiển thị tám bit quan trọng nhất có được với máy ảnh. Nếu bạn đang sử dụng một thiết bị thu nhận hình

ảnh quốc gia, kỹ thuật này là mặc định được sử dụng bởi Đo lường & Tự động hóa Explorer (MAX).

### **2.2.3. Bảng màu**

Tại thời điểm hình ảnh thang độ xám được hiển thị trên màn hình, NI Vision chuyển đổi giá trị của từng pixel của hình ảnh thành màu đỏ, xanh lá cây và xanh dương cường độ cho pixel tương ứng hiển thị trên màn hình. Quá trình này được gọi là bảng màu, liên kết một màu với từng màu có thể giá trị thang độ xám của một hình ảnh. Tầm nhìn NI cung cấp khả năng tùy chỉnh bảng màu được sử dụng để hiển thị hình ảnh thang độ xám 8 bit.

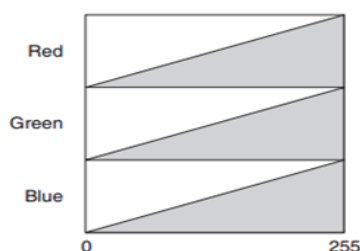
Với bảng màu, bạn có thể tạo các hình ảnh đại diện khác nhau của hình ảnh mà không thay đổi dữ liệu pixel. Bảng màu có thể tạo hiệu ứng, chẳng hạn như hiển thị ánh sáng hoặc hiển thị mã màu. Trong trường hợp sau, bảng màu rất hữu ích để chi tiết các thành phần hình ảnh cụ thể trong đó tổng số số lượng màu sắc có hạn. Hiển thị hình ảnh trong các bảng màu khác nhau giúp nhấn mạnh các vùng với cường độ cụ thể, xác định các biến thể cấp độ xám trơn tru hoặc đột ngột và truyền đạt các chi tiết có thể khó nhận biết trong ảnh thang độ xám. Ví dụ, mắt người nhạy cảm hơn với cường độ nhỏ biến đổi trong một khu vực sáng hơn trong một khu vực tối. Sử dụng bảng màu có thể giúp bạn phân biệt những thay đổi nhỏ này.

Một bảng màu là một mảng được xác định trước hoặc do người dùng xác định các giá trị RGB. Nó định nghĩa đối với mỗi giá trị cấp độ xám có thể có một giá trị màu tương ứng để hiển thị pixel. Giá trị cấp độ xám của pixel hoạt động như một địa chỉ được lập chỉ mục thành bảng, trả về ba giá trị tương ứng với một màu đỏ, xanh lá cây và xanh dương (RGB) cường độ. Tập hợp các giá trị RGB này xác định một bảng màu trong đó thay đổi số lượng màu đỏ, xanh lá cây và xanh dương được trộn lẫn để tạo ra một đại diện màu sắc của phạm vi giá trị. Trong trường hợp hình ảnh thang độ xám 8 bit, pixel có thể lấy 28 hoặc 256, giá trị trong khoảng từ 0 đến 255. Bảng màu bao gồm 256 phần tử RGB. Một màu cụ thể là kết quả của việc áp dụng giá trị từ 0 đến 255 cho mỗi màu của ba thành phần màu: đỏ, xanh lá cây và xanh dương. Nếu màu đỏ, xanh lá cây và các thành phần màu xanh có giá trị giống hệt nhau, kết quả là

một pixel mức xám giá trị. Một bảng màu xám liên kết các sắc thái khác nhau của màu xám với mỗi giá trị để tạo ra sự chuyển màu tuyến tính và liên tục của màu xám, từ đen sang trắng. Bạn có thể thiết lập bảng màu để gán màu đen cho giá trị 0 và trắng đến 255, hoặc ngược lại. Các bảng màu khác có thể phản ánh tuyến tính hoặc phi tuyến chuyển màu từ đỏ sang xanh, nâu nhạt đến nâu sẫm, vân vân. NI Vision có năm bảng màu được xác định trước. Mỗi bảng màu nhấn mạnh sắc thái khác nhau của màu xám. Bao gồm năm bảng màu dưới đây được xác định trước. Các biểu đồ trong mỗi phần biểu thị các bảng màu được sử dụng bởi từng bảng màu. Các trục ngang của đồ thị đại diện cho đầu vào phạm vi mức xám  $[0, 255]$  và trục dọc đại diện cho RGB cường độ được gán cho một giá trị cấp độ xám nhất định.

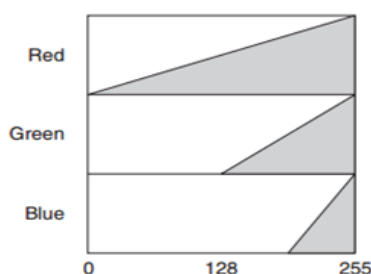
### **2.2.3.1 Bảng màu xám(Gray Palette)**

Bảng màu này có sự thay đổi cấp độ xám dần dần từ đen sang trắng. Mỗi giá trị được gán cho một lượng bằng nhau màu đỏ, xanh lá cây và xanh dương theo thứ tự để tạo ra một mức độ màu xám.



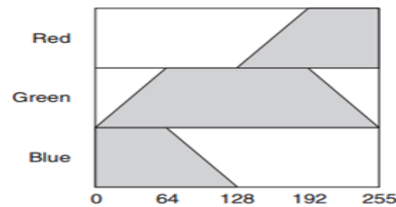
### **2.2.3.2 Bảng nhiệt độ**

Bảng màu này có sự chuyển màu từ nâu nhạt sang nâu sẫm. 0 là màu đen và 255 là màu trắng.



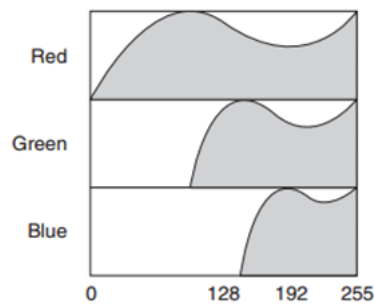
### **2.2.3.3 Bảng màu cầu vồng**

Bảng màu này có sự chuyển màu từ xanh sang đỏ với một phạm vi nổi bật của rau xanh trong phạm vi giá trị trung bình. 0 là màu xanh và 255 là màu đỏ.



#### **2.2.3.4 Bảng màu Gradient**

Bảng màu này có sự chuyển màu từ đỏ sang trắng với một phạm vi nổi bật là màu xanh nhạt trong phạm vi giá trị trên. 0 là màu đen và 255 là màu trắng.



## **CHƯƠNG 3: XỬ LÝ KHỚP MẪU TRÊN PHẦN MỀM LABVIEW**

### **3.1. Giới thiệu**

Khớp mẫu nhanh chóng định vị các vùng của ảnh thang độ xám phù hợp một mẫu tham chiếu đã biết, cũng được gọi là một mô hình hoặc mẫu.

Lưu ý: Mẫu là một đại diện lý tưởng hóa của một tính năng trong hình ảnh.

Khi sử dụng khớp mẫu, ta tạo một mẫu đại diện cho đối tượng mà ta đang tìm kiếm. Thuật toán tìm kiếm các phiên bản của mẫu trong mỗi hình ảnh thu được, tính toán đến mức độ gần giống với mẫu chuẩn.

Khớp mẫu tìm thấy mẫu bất kể biến thể ánh sáng, làm mờ, nhiễu và biến đổi hình học như dịch chuyển, xoay hoặc chia tỷ lệ của mẫu.

Các thuật toán khớp mẫu là một số chức năng quan trọng nhất trong thị giác máy vì sử dụng chúng trong các ứng dụng khác nhau. Ta có thể dùng khớp mẫu trong ba ứng dụng chung sau:

- Sắp xếp thứ hạng Xác định vị trí và hướng của một cái đã biết đối tượng bằng cách định vị các fiducials. Sử dụng các fiducials làm điểm tham chiếu trên đối tượng.
- Đo độ dài đo, đường kính, góc và các chỉ số quan trọng khác kích thước. Nếu các phép đo nằm ngoài mức dung sai, thành phần bị từ chối. Sử dụng khớp mẫu để xác định vị trí đối tượng ta muốn đánh giá.
- Kiểm tra phát hiện các lỗi đơn giản, chẳng hạn như các phân bị thiếu hoặc không thể đọc được in.

### **3.2. Kỹ thuật ghép mẫu**

Các kỹ thuật khớp mẫu bao gồm chuẩn hóa chéo tương quan, kết hợp hình chóp, khớp tỷ lệ và xoay bất biến, và hiệu hình ảnh.

#### **3.2.1. Chuẩn hóa chéo tương quan**

Chuẩn hóa chéo tương quan là phương pháp phổ biến nhất để tìm mẫu trong một hình ảnh. Bởi vì cơ chế cơ bản cho mỗi tương quan là dựa trên một loạt các phép toán nhân, quá trình tương quan là mất thời gian. Các công nghệ như MMX cho phép song song phép nhân và giảm thời gian tính toán tổng thể. Để tăng tốc độ của quá trình khớp, giảm kích thước của hình ảnh và hạn chế vùng của hình ảnh trong đó sự



phù hợp xảy ra. Phù hợp với hình kim tự tháp và hình ảnh hiểu biết là hai cách để tăng tốc độ của quá trình khớp.

### **3.2.2. Kết hợp tỷ lệ và xoay - bất biến**

Tương quan chéo chuẩn hóa là một kỹ thuật tốt để tìm mẫu trong một hình ảnh khi các mẫu trong hình ảnh không được thu nhỏ hoặc xoay. Thông thường, tương quan chéo có thể phát hiện các mẫu có cùng kích thước lên đến một vòng quay từ 5 ° đến 10 °. Mở rộng mối tương quan để phát hiện các mẫu bất biến để thay đổi quy mô và xoay vòng là khó khăn.

Để khớp tỷ lệ bất biến, ta phải lặp lại quá trình chia tỷ lệ hoặc thay đổi kích thước mẫu và sau đó thực hiện thao tác tương quan. Điều này thêm một lượng tính toán đáng kể vào quá trình kết hợp.

Bình thường hóa cho vòng quay thậm chí còn khó khăn hơn. Nếu một đầu mối liên quan đến luân chuyển có thể được trích xuất từ hình ảnh, bạn chỉ cần xoay mẫu và thực hiện tương quan. Tuy nhiên, nếu bản chất của xoay là không rõ, tìm kiếm sự phù hợp tốt nhất đòi hỏi phải xoay toàn bộ mẫu.

Ta cũng có thể thực hiện tương quan trong miền tần số bằng cách sử dụng Biến đổi Fourier nhanh (FFT). Nếu hình ảnh và mẫu giống nhau kích thước, cách tiếp cận này hiệu quả hơn so với thực hiện tương quan trong miền không gian. Trong miền tần số, tương quan có được bởi nhân FFT của hình ảnh với liên hợp phức tạp của FFT của bản mẫu. Tương quan chéo bình thường hóa là khó khăn hơn đáng kể để thực hiện trong miền tần số.

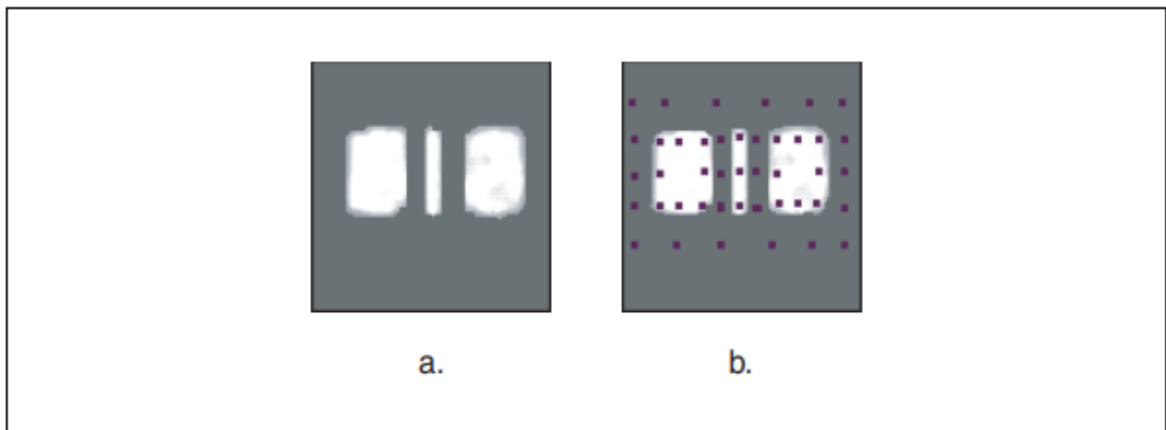
### **3.2.3. Kết hợp kim tự tháp (kết hợp hình chóp)**

Ta có thể cải thiện thời gian tính toán khớp mẫu bằng cách giảm kích thước của hình ảnh và mẫu. Trong kết hợp hình chóp, cả hình ảnh và mẫu được lấy mẫu với độ phân giải không gian nhỏ hơn. Chẳng hạn, bởi lấy mẫu mọi pixel khác, hình ảnh và mẫu có thể được giảm đến một phần tư kích thước ban đầu của chúng. Kết hợp được thực hiện đầu tiên trên giảm hình ảnh. Bởi vì hình ảnh nhỏ hơn, kết hợp nhanh hơn. Khi kết hợp hoàn tất, chỉ những khu vực có điểm phù hợp cao mới cần được coi là khu vực phù hợp trong hình ảnh gốc.



### 3.2.4. Hiểu hình ảnh

Một tính năng khớp mẫu là một mẫu pixel nổi bật mô tả một bản mẫu. Bởi vì hầu hết các hình ảnh đều chứa thông tin dư thừa, sử dụng tất cả thông tin trong hình ảnh phù hợp với các mẫu không nhạy cảm với thời gian và không chính xác sử dụng kỹ thuật lấy mẫu không đồng nhất kết hợp hình ảnh hiểu để mô tả kỹ lưỡng và hiệu quả các mẫu. Điều này kỹ thuật lấy mẫu thông minh đặc biệt bao gồm sự kết hợp của cạnh pixel và pixel vùng như trong hình 3.1. Sử dụng tương tự kỹ thuật khi người dùng chỉ ra rằng mô hình có thể được xoay trong hình ảnh. Kỹ thuật này sử dụng các pixel mẫu được chọn đặc biệt có các giá trị khác nhau hoặc các thay đổi tương đối trong các giá trị. Lấy mẫu thông minh của cả hai mẫu đều làm giảm sự dư thừa thông tin và nhấn mạnh tính năng cho phép hiệu quả, nhưng mạnh mẽ, thực hiện tương quan chéo. Kết hợp mẫu có thể xác định chính xác các đối tượng có kích thước khác nhau ( $\pm 5\%$ ) và hướng (giữa  $0^\circ$  và  $360^\circ$ ) và có vẻ ngoài xuống cấp.



Hình 3.1: Kỹ thuật lấy mẫu phù hợp

### 3.3. Thuật toán tìm kiếm

Chuẩn hóa chéo tương quan

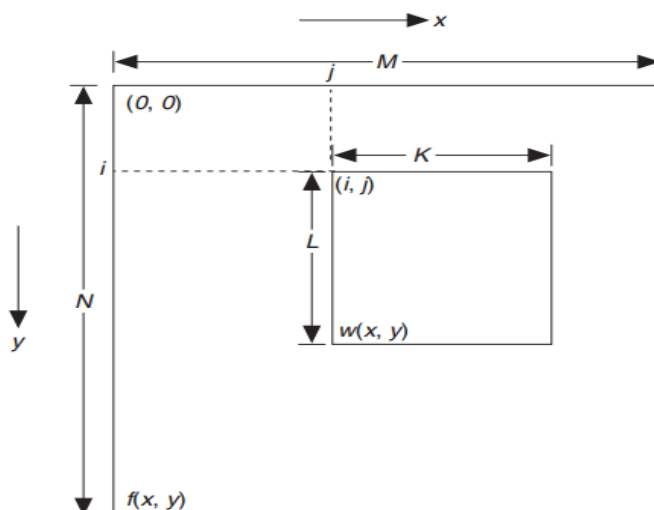
Sau đây là khái niệm cơ bản về tương quan: Xem xét một tiểu phần  $w(x, y)$  có kích thước  $K \times L$  trong một hình ảnh  $f(x, y)$  có kích thước  $M \times N$ , trong đó  $K < M$  và  $L < N$ . Mỗi tương quan giữa  $w(x, y)$  và  $f(x, y)$  tại một điểm  $(i, j)$  là được cho bởi

$$C(i, j) = \sum_{x=0}^{L-1} \sum_{y=0}^{K-1} w(x, y) f(x + i, y + j)$$

trong đó  $i = 0, 1, \dots, M - 1$

$j = 0, 1, \dots, N - 1$  và tổng kết được thực hiện trên vùng trong hình ảnh trong đó  $w$  và  $f$  trùng nhau.

Hình 4.2 minh họa chuẩn hóa tương quan. Giả sử rằng tọa độ gốc của ảnh  $f$  nằm ở góc trên cùng bên trái. Tương quan là quá trình di chuyển mẫu hoặc subimage  $w$  xung quanh khu vực hình ảnh và tính toán giá trị  $C$  trong khu vực đó. Điều này liên quan đến việc nhân từng pixel trong mẫu bằng pixel hình ảnh mà nó chồng lấp và sau đó tổng hợp kết quả trên tất cả các pixel của mẫu. Giá trị tối đa của  $C$  chỉ vị trí trong đó phù hợp nhất với  $f$ . Các giá trị tương quan không chính xác tại các biên của hình ảnh.



*Hình 3.2: Chuẩn hóa tương quan*

Tương quan cơ bản là rất nhạy cảm với sự thay đổi biên độ trong hình ảnh, như vậy như cường độ, và trong mẫu. Ví dụ: nếu cường độ của hình ảnh  $f$  được nhân đôi, các giá trị của  $c$ . Bạn có thể vượt qua sự nhạy cảm bằng cách tính toán hệ số tương quan chuẩn hóa, được định nghĩa là:

$$R(i,j) = \frac{\sum_{x=0}^{L-1} \sum_{y=0}^{K-1} (w(x,y) - \bar{w})(f(x+i,y+j) - \bar{f}(i,j))}{\left[ \sum_{x=0}^{L-1} \sum_{y=0}^{K-1} (w(x,y) - \bar{w})^2 \right]^{\frac{1}{2}} \left[ \sum_{x=0}^{L-1} \sum_{y=0}^{K-1} (f(x+i,y+j) - \bar{f}(i,j))^2 \right]^{\frac{1}{2}}}$$

Trong đó  $w$  (chỉ được tính một lần) là giá trị cường độ trung bình của các pixel trong mẫu  $w$ . Biến  $f$  là giá trị trung bình của  $f$  trong vùng trùng với vị trí hiện tại của  $w$ . Giá trị của  $R$  nằm trong phạm vi. Tích đến 1 và không phụ thuộc vào thay đổi tỷ lệ trong các giá trị cường độ của  $f$  và  $w$ .

### 3.4. Xây dựng, kết nối khối chức năng trên phần mềm LABVIEW

#### 3.4.1. Giới thiệu phần mềm LABVIEW

(Laboratory Virtual Instrumentation Engineering Workbench) là một phần mềm máy tính được phát triển bởi National Instruments. LabVIEW dùng trong hầu hết các phòng thí nghiệm, lĩnh vực khoa học kỹ thuật như tự động hóa, điều khiển, điện tử, cơ điện tử, hàng không, hóa sinh, điện tử y sinh ở các nước đặc biệt là Mỹ, Hàn Quốc, Nhật Bản.

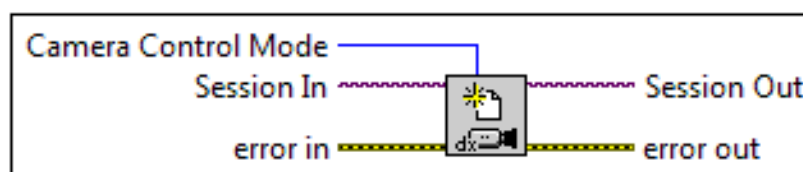
**LabVIEW là gì?** LabVIEW là một môi trường lập trình đồ họa mà bạn có thể sử dụng để tạo các ứng dụng với giao diện người dùng chuyên nghiệp một cách nhanh chóng và hiệu quả. Ngôn ngữ lưu đồ đồ họa của LabVIEW hấp dẫn các kỹ sư và nhà khoa học trên toàn thế giới như một phương pháp trực giác hơn trong việc tự động hóa các hệ thống đo lường và điều khiển. Ngôn ngữ lưu đồ kết hợp với VO gắn liền và điều khiển giao diện người sử dụng tương tác cùng đèn chỉ báo làm cho LabVIEW trở thành một sự lựa chọn lý tưởng cho kỹ sư và nhà khoa học. Trong thực tế, nền tảng LabVIEW có khả năng tích hợp với hàng nghìn thiết bị phần cứng và cung cấp hàng trăm thư viện được xây dựng sẵn để phân tích nâng cao và hiển thị dữ liệu giúp bạn tạo ra các thiết bị ảo có thể tùy chỉnh theo nhu cầu của mình.

Bởi vì chương trình LabVIEW mô phỏng giao diện và hoạt động của các thiết bị thực, chẳng hạn như dao động ký và thiết bị đo năng lượng, chương trình LabVIEW được

gọi là thiết bị ảo (Virtual Instrument), thường gọi tắt là VI. VI có Front Panel và Block Diagram. Front Panel là giao diện người dùng. Block Diagram là chương trình phía sau giao diện người dùng. Sau khi bạn xây dựng Front Panel, bạn thêm mã (code) để điều khiển các đối tượng trên Front Panel bằng cách sử dụng các hình đồ họa đại diện cho các hàm. Mã trên Block Diagram là mã dạng đồ họa, thường được biết đến là G code (mã G) hoặc Block Diagram code.

### 3.4.2. Các khối chức năng

#### a, IMAQdx Open Camera VI



Mở camera, truy vấn camera để biết khả năng của nó, tải tập tin cấu hình camera và tạo một tham chiếu duy nhất cho camera. Sử dụng IMAQdx Mở Camera VI khi ta kết thúc với tham chiếu.

**Camera Control Mode** là chế độ điều khiển của camera được sử dụng trong quá trình phát hình ảnh. Mở một camera trong chế độ điều khiển để chủ động cấu hình và thu thập dữ liệu hình ảnh. Mở camera ở chế độ nghe để thu thập động dữ liệu hình ảnh từ phiên được mở ở chế độ điều khiển trên máy chủ hoặc máy tính đích khác. Giá trị mặc định là Bộ điều khiển.

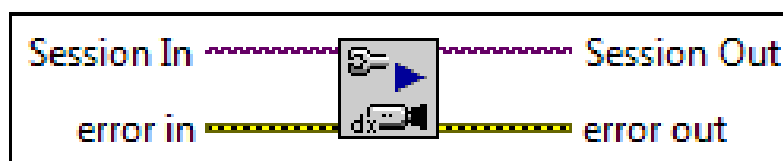
**Session In** chỉ định tên của máy ảnh bạn muốn mở. Giá trị mặc định là cam 0.

**Error in** mô tả trạng thái lỗi trước khi VI hoặc chức năng này chạy. Mặc định là không có lỗi. Nếu xảy ra lỗi trước khi VI hoặc hàm này chạy, VI hoặc hàm sẽ chuyển lỗi về giá trị sang lỗi. VI hoặc chức năng này chỉ chạy bình thường nếu không có lỗi xảy ra trước khi VI hoặc chức năng này chạy. Nếu xảy ra lỗi trong khi VI hoặc chức năng này chạy, nó sẽ chạy bình thường và đặt lỗi trạng thái lỗi của chính nó. Sử dụng Trình xử lý lỗi đơn giản hoặc Trình xử lý lỗi chung VI để hiển thị mô tả mã lỗi. Sử dụng lỗi trong và lỗi ra để kiểm tra lỗi và chỉ định thứ tự thực hiện bằng cách nối dây lỗi từ một nút sang lỗi trong nút tiếp theo. Giải thích Lỗi (hoặc Giải thích Cảnh báo) cung cấp thêm thông tin về lỗi được hiển thị.

**Session Out** là một tài liệu tham khảo duy nhất cho máy ảnh. **Session Out** cũng giống như **Session In**.

**Error out** chứa thông tin lỗi. Nếu lỗi trong chỉ ra rằng đã xảy ra lỗi trước khi VI hoặc hàm này chạy, lỗi xuất hiện chứa thông tin lỗi tương tự. Mặt khác, lỗi ra mô tả trạng thái lỗi mà VI này tạo ra. Tùy chọn bật lên Giải thích Lỗi (hoặc Giải thích Cảnh báo) cung cấp thêm thông tin về lỗi được hiển thị.

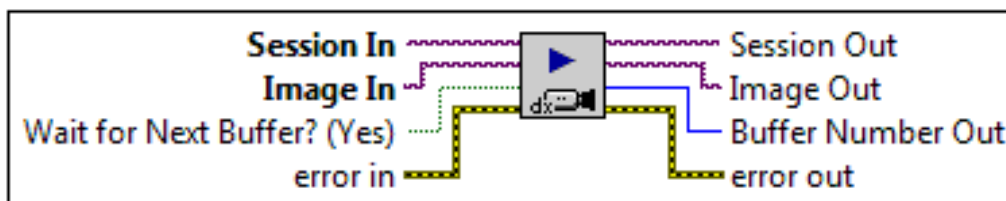
#### b, IMAQdx Configure Grab VI



Định cấu hình và bắt đầu thu lại. Một cái kẹp thực hiện một giao dịch lặp lại liên tục trên một vòng đệm. Sử dụng lấy VI để thu nhận hình ảnh tốc độ cao. Sử dụng IMAQdx Grab VI để sao chép hình ảnh ra khỏi bộ đệm. Nếu bạn gọi VI này trước khi gọi IMAQdx Open Camera VI, IMAQdx Configure Grab VI sẽ sử dụng cam 0 theo mặc định. Sử dụng IMAQdx Uncool Acquisition VI để hủy cấu hình mua lại.

**Session In** là một tham chiếu duy nhất cho máy ảnh, mà bạn có thể có được với Máy ảnh mở IMAQdx VI.

#### c, IMAQdx Grab VI



Có được khung hình mới nhất vào Image Out. Chỉ gọi VI này sau khi gọi IMAQdx Định cấu hình Grab VI. Nếu loại hình ảnh không khớp với định dạng video của máy ảnh, VI này sẽ thay đổi loại hình ảnh thành định dạng phù hợp.

**Session In** là một tham chiếu duy nhất cho máy ảnh, mà bạn có thể có được với Máy ảnh mở IMAQdx VI.

**Image In** là tham chiếu đến hình ảnh nhận dữ liệu pixel đã chụp.

**Wait for Next Buffer? (Yes)** nếu giá trị là Có, trình điều khiển sẽ đợi bộ đệm có sẵn tiếp theo. Nếu chờ bộ đệm tiếp theo? Giá trị là Không, trình điều khiển sẽ không đợi bộ đệm có sẵn tiếp theo và thay vào đó sẽ trả về bộ đệm thu được cuối cùng.

**Image Out** là tham chiếu đến hình ảnh được chụp.

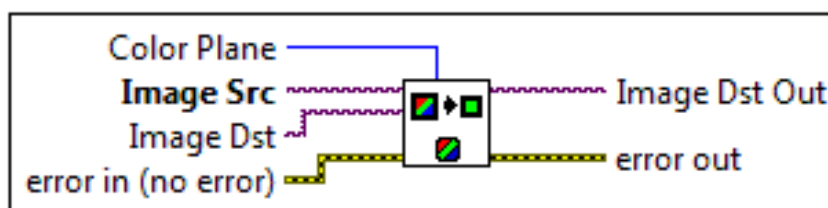
**Buffer Number Out** là số bộ đệm thu được thực tế được trả về.

#### d, IMAQdx Close Camera VI



**Session In** là một tham chiếu duy nhất cho máy ảnh, mà bạn có thể có được với Máy ảnh mở IMAQdx VI.

#### e, IMAQ ExtractSingleColorPlane VI



Sử dụng các tiện ích màu VI để thao tác màu sắc và mặt phẳng màu của hình ảnh. Sử dụng các VI này để trích xuất các mặt phẳng màu khác nhau từ một hình ảnh, thay thế các mặt phẳng của hình ảnh màu bằng dữ liệu mới, chuyển đổi hình ảnh màu thành mảng 2D và quay lại, đọc và đặt giá trị pixel trong hình ảnh màu và chuyển đổi giá trị pixel từ một không gian màu khác.

**Color Plane** xác định mặt phẳng màu cần trích xuất. Chọn từ các giá trị sau:

Lưu ý Hình ảnh RGB 64 bit không được chỉ định chỉ hỗ trợ các giá trị mặt phẳng màu là Đỏ, Xanh lục hoặc Xanh lam.

Red (0) (Mặc định) Trích xuất mặt phẳng màu đỏ

Màu xanh lá cây (1) Trích xuất mặt phẳng màu xanh lục

Blue (2) Trích xuất mặt phẳng màu xanh

Hue (3) Trích xuất mặt phẳng màu

Độ bão hòa (4) Trích xuất mặt phẳng màu bão hòa

Độ chói (5) Chiết xuất mặt phẳng màu chói

Giá trị (6) Trích xuất mặt phẳng màu giá trị

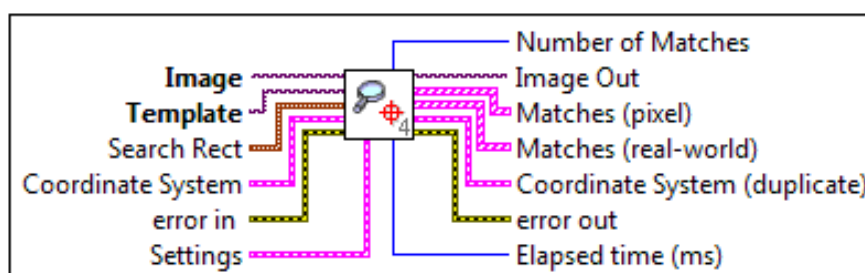
Cường độ (7) Trích xuất mặt phẳng màu cường độ

**Image Src** là tham chiếu đến một hình ảnh màu có một trong các mặt phẳng màu của nó được trích xuất. Nếu Image Dst không được kết nối, hình ảnh nguồn được chuyển đổi thành hình ảnh chứa mặt phẳng được trích xuất.

**Image Dst** là một tham chiếu đến hình ảnh đích.

**Image Dst Out** là một tham chiếu đến hình ảnh đích. Nếu Image Dst được kết nối, Image Dst Out giống như Image Dst. Mặt khác, Image Dst Out đề cập đến hình ảnh được tham chiếu bởi Image Src.

#### f, IMAQ Find Pattern 4 VI



Tìm kiếm một hình ảnh mẫu trong một khu vực tìm kiếm hình chữ nhật của hình ảnh.

**Image** là một tham chiếu đến hình ảnh nguồn.

**Template** là một tham chiếu đến hình ảnh để xác định vị trí trong quá trình khớp. Hình ảnh mẫu là hình ảnh thu được từ đầu ra của Mẫu IMAQ Learn 5 VI. Nếu mẫu chưa được học, VI này trước tiên trải qua quá trình học.

**Search Rect** chỉ định vùng tìm kiếm hình chữ nhật trong ảnh. Các tọa độ Trái, Trên, Phải và Dưới mô tả một hình chữ nhật không xoay. Rotation chỉ định góc quay của hình chữ nhật xung quanh tâm của nó.

**Coordinate System** chỉ định hệ tọa độ mà hình chữ nhật được liên kết. Nếu đầu vào hệ thống tọa độ được sử dụng (được kết nối), vị trí đo cường độ sẽ bị dịch chuyển và xoay bởi sự khác biệt giữa vị trí tham chiếu của hệ tọa độ và vị trí mới.

**Settings** là một cụm xác định các tham số của thuật toán vị trí mẫu và thông tin được phủ lên hình ảnh kết quả.

**Elapsed time (ms)** trả về thời lượng tìm kiếm, tính bằng ms.

**Matches** là một loạt các cụm khớp.

**Image Out** là một tham chiếu đến hình ảnh đích. Vùng tìm kiếm and/or kết quả của phép đo có thể được phủ lên hình ảnh theo các cài đặt.

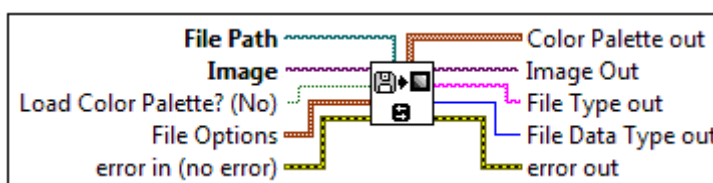
**Number of Matches** là số lượng các cụm khớp mẫu được tìm thấy trong hình ảnh kiểm tra dựa trên các cài đặt đầu vào.

**Coordinate System (duplicate)** là một tham chiếu đến hệ tọa độ.

**Error in** mô tả trạng thái lỗi trước khi VI hoặc hàm này chạy. Mặc định là không có lỗi. Nếu xảy ra lỗi trước khi VI hoặc hàm này chạy, VI hoặc hàm sẽ chuyển lỗi về giá trị sang lỗi. VI hoặc chức năng này chỉ chạy bình thường nếu không có lỗi xảy ra trước khi VI hoặc chức năng này chạy. Nếu xảy ra lỗi trong khi VI hoặc chức năng này chạy, nó sẽ chạy bình thường và đặt lỗi trạng thái lỗi của chính nó. Sử dụng Trình xử lý lỗi đơn giản hoặc trình xử lý lỗi chung VI để hiển thị mô tả mã lỗi. Sử dụng lỗi trong và lỗi ra để kiểm tra lỗi và chỉ định thứ tự thực hiện bằng cách nối dây lỗi từ một nút sang lỗi trong nút tiếp theo.

**Error out** chứa thông tin lỗi. Nếu lỗi trong chỉ ra rằng đã xảy ra lỗi trước khi VI hoặc hàm này chạy, lỗi xuất hiện chứa thông tin lỗi tương tự. Mặt khác, nó mô tả trạng thái lỗi mà VI hoặc hàm này tạo ra. Nhấp chuột phải vào chỉ báo lỗi trên bảng mặt trước và chọn giải thích lỗi từ menu phím tắt để biết thêm thông tin về lỗi.

### **g, IMAQ ReadFile VI**



Đọc một tập tin hình ảnh. Định dạng tệp có thể là định dạng chuẩn (BMP, TIFF, JPEG, JPEG2000, PNG và AIPD) hoặc định dạng không chuẩn được người dùng biết đến. Trong mọi trường hợp, các pixel đọc được tự động chuyển thành loại hình ảnh được truyền bởi Image.



**File Path** là tên đường dẫn đầy đủ, bao gồm ổ đĩa, thư mục và tên tệp của tệp cần đọc.

**Image** là một tham chiếu đến hình ảnh mà dữ liệu từ tệp hình ảnh được áp dụng.

**Load Color Palette? (No)** xác định có tải bảng màu có trong tệp hay không, nếu bảng màu tồn tại. Nếu được tải, bảng này được đọc và trả về bởi Bảng màu đầu ra. Mặc định này sai.

**File Options** là một cụm các giá trị tùy chọn người dùng mà bạn có thể sử dụng để đọc các định dạng tệp không chuẩn. Cấu trúc phải được biết đến với người dùng.

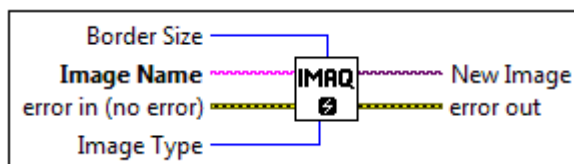
**Color Palette out** có chứa bảng màu RGB (nếu tệp có một) đọc từ tệp khi người dùng chuyển giá trị TRUE cho Bảng màu tải đầu vào? (Không).

**Image Out** là một tham chiếu đến hình ảnh đích. Nếu Image Dst được kết nối, Image Dst Out giống như Image Dst. Mặt khác, Image Dst Out đề cập đến hình ảnh được tham chiếu bởi Image Src.

**File Type out** chỉ ra loại tệp được đọc. Chuỗi này trả về một định danh của định dạng tệp, có thể là BMP, TIFF, JPEG, PNG hoặc AIPD (định dạng tệp nội bộ). Loại tệp trả về xxx nếu định dạng tệp không xác định.

**File Data Type out** cho biết kích thước pixel được xác định trong tiêu đề cho các loại tệp hình ảnh tiêu chuẩn. Tùy chọn tệp không cần thiết để đọc tệp hình ảnh tiêu chuẩn. Đối với các loại tệp hình ảnh khác, các giá trị được trả về được truyền từ tùy chọn tệp/loại dữ liệu tệp.

#### h, IMAQ Create VI

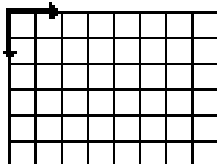


Tạo một vị trí bộ nhớ tạm thời cho một hình ảnh. Sử dụng IMAQ tạo kết hợp với IMAQ loại bỏ VI để tạo hoặc loại bỏ hình ảnh NI Vision trong LabVIEW.

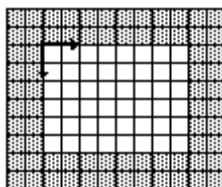
**Border Size** xác định chiều rộng, tính bằng pixel, của đường viền để tạo xung quanh hình ảnh. Những pixel này chỉ được sử dụng cho các VI cụ thể. Tạo đường viền ở đầu ứng dụng của bạn nếu hình ảnh sẽ được xử lý sau bằng cách sử dụng các

chức năng yêu cầu đường viền (ví dụ: ghi nhãn và hình thái). Giá trị đường viền mặc định là 3. Với đường viền ba pixel, bạn có thể sử dụng hạt nhân lên đến  $7 \times 7$  mà không thay đổi. Nếu bạn dự định sử dụng hạt nhân lớn hơn  $7 \times 7$  trong quy trình của mình, hãy chỉ định đường viền lớn hơn khi tạo hình ảnh của bạn.

Đồ họa sau đây minh họa một hình ảnh  $8 \times 6$  với đường viền bằng 0.



Trong hình ảnh  $8 \times 6$  sau đây, đường viền bằng 2, cho phép sử dụng hạt nhân lên đến  $5 \times 5$ .



Lưu ý đường viền của hình ảnh chỉ được tính đến khi hình ảnh được xử lý. Nó không bao giờ được hiển thị hoặc lưu trữ trong một tập tin.

**Image Name** là tên liên quan đến hình ảnh được tạo. Mỗi hình ảnh được tạo ra phải có một tên duy nhất.

**Image Type** chỉ định loại hình ảnh. Chọn từ các giá trị sau:

Grayscale (U8) (0) 8 bit cho mỗi pixel (không dấu, đơn sắc tiêu chuẩn)

Thang độ xám (16) (1) 16 bit cho mỗi pixel (đã ký)

Thang độ xám (SGL) (2) 32 bit cho mỗi pixel (điểm nổi)

Phức tạp (CSG) (3)  $2 \times 32$  bit cho mỗi pixel (điểm nổi)

RGB (U32) (4) 32 bit mỗi pixel (đỏ, xanh lá cây, xanh dương, alpha)

HSL (U32) (5) 32 bit mỗi pixel (màu sắc, độ bão hòa, độ chói, alpha)

RGB (U64) (6) 64 bit mỗi pixel (đỏ, xanh lá cây, xanh lam, alpha)

Grayscale (U16) (7) 16 bit cho mỗi pixel (không dấu, đơn sắc tiêu chuẩn)

**New Image** là tham chiếu Hình ảnh được cung cấp làm đầu vào cho tất cả các chức năng tiếp theo (xuôi dòng) được sử dụng bởi NI Vision. Nhiều hình ảnh có thể được tạo trong một ứng dụng LabVIEW.

### i, Array To Cluster Function

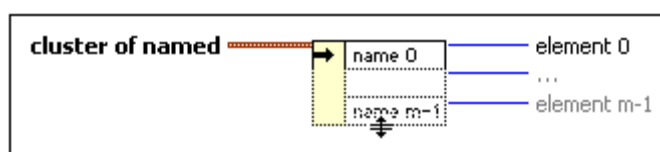


Chuyển đổi một mảng 1D thành một cụm các phần tử cùng loại với các phần tử mảng. Nhấp chuột phải vào chức năng và chọn Kích thước cụm từ menu phím tắt để đặt số lượng phần tử trong cụm.

Mặc định là 9. Kích thước cụm tối đa cho chức năng này là 256.

Sử dụng chức năng này khi bạn muốn hiển thị các phần tử cùng loại trong chỉ báo cụm bảng mặt trước nhưng muốn thao tác các phần tử trên sơ đồ khối bằng các giá trị chỉ mục của chúng.

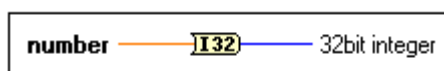
### j, Unbundle By Name Function



Trả về các thành phần cụm có tên bạn chỉ định.

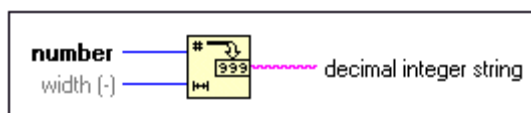
Ta không phải theo dõi thứ tự của các yếu tố trong cụm. Hàm này không yêu cầu số lượng phần tử khớp với số trong cụm. Sau khi bạn kết nối một cụm với chức năng này, bạn có thể chọn một thành phần riêng lẻ từ chức năng.

### k, To Long Integer Function



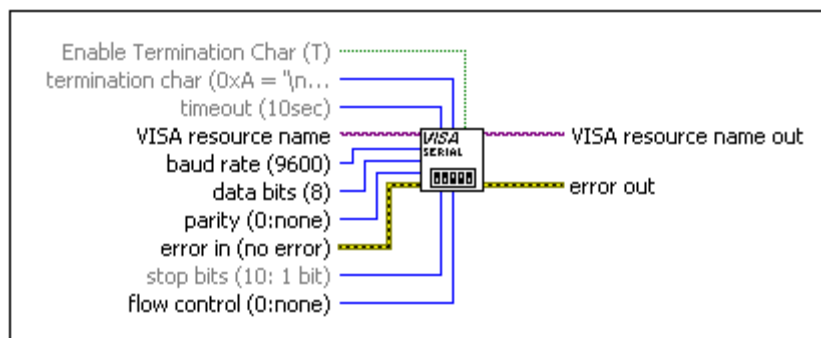
Chuyển đổi một số thành số nguyên 32 bit trong phạm vi  $-(2^{31})$  thành  $(2^{31}) - 1$ . Hàm này làm tròn tất cả các giá trị số của dấu phẩy động và điểm cố định thành số nguyên gần nhất.

### l, Number To Decimal String Function



Chuyển đổi số thành một chuỗi các chữ số thập phân ít nhất là chiều rộng ký tự rộng hoặc rộng hơn nếu cần. Nếu số là dấu phẩy động hoặc điểm cố định, nó được làm tròn thành số nguyên 64 bit trước khi chuyển đổi.

### **m, VISA Configure Serial Port VI**



Khởi tạo cổng nối tiếp được chỉ định bởi tên tài nguyên VISA cho các cài đặt đã chỉ định. Nối dữ liệu vào đầu vào tên tài nguyên VISA để xác định thể hiện đa hình để sử dụng hoặc chọn thủ công.

**VISA resource name** chỉ định cổng COM sẽ được mở. Kiểm soát tên COM

**Baud rate** là tốc độ truyền. Mặc định là 9600.

**Data bits** liệu là số bit trong dữ liệu đến. Giá trị của các bit dữ liệu nằm trong khoảng từ năm đến tám. Giá trị mặc định là 8.

**Parity** chỉ định tính chẵn lẻ được sử dụng cho mọi khung hình được truyền hoặc nhận. Đầu vào này chấp nhận các giá trị sau.

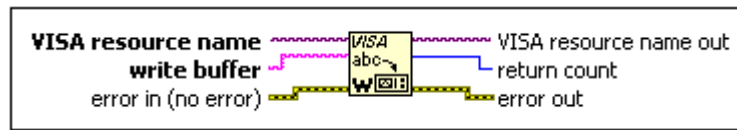
**Error in** các điều kiện lỗi xảy ra trước khi nút này chạy. Đầu vào này cung cấp lỗi tiêu chuẩn trong chức năng.

**Stop bits** chỉ định số lượng bit dừng được sử dụng để chỉ ra kết thúc của khung.

**Flow control** đặt loại điều khiển được sử dụng bởi cơ chế truyền. Đầu vào này chấp nhận các giá trị sau.

**VISA resource name out** là một bản sao của tên COM chức năng VISA trả về.

### **n, VISA Write Function**



**VISA resource name** chỉ định tài nguyên sẽ được mở. Kiểm soát tên tài nguyên VISA cũng chỉ định phiên và lớp.

**Write buffer** chứa dữ liệu được ghi vào thiết bị.

**VISA resource name out** là một bản sao của tên tài nguyên VISA mà chức năng VISA trả về.

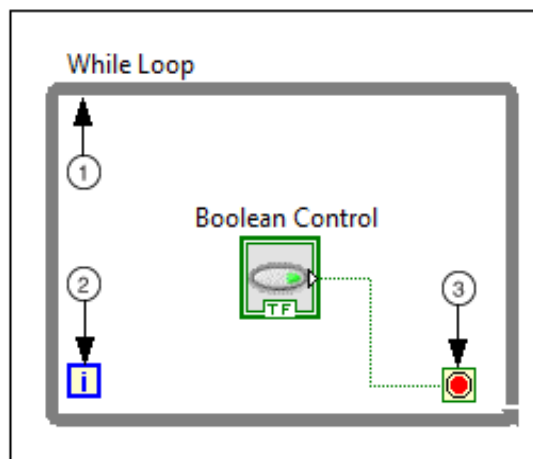
**Return count** chứa số byte thực tế được ghi.

#### p, VISA Close Function



**VISA resource name** chỉ định tài nguyên sẽ được mở. Đóng VISA

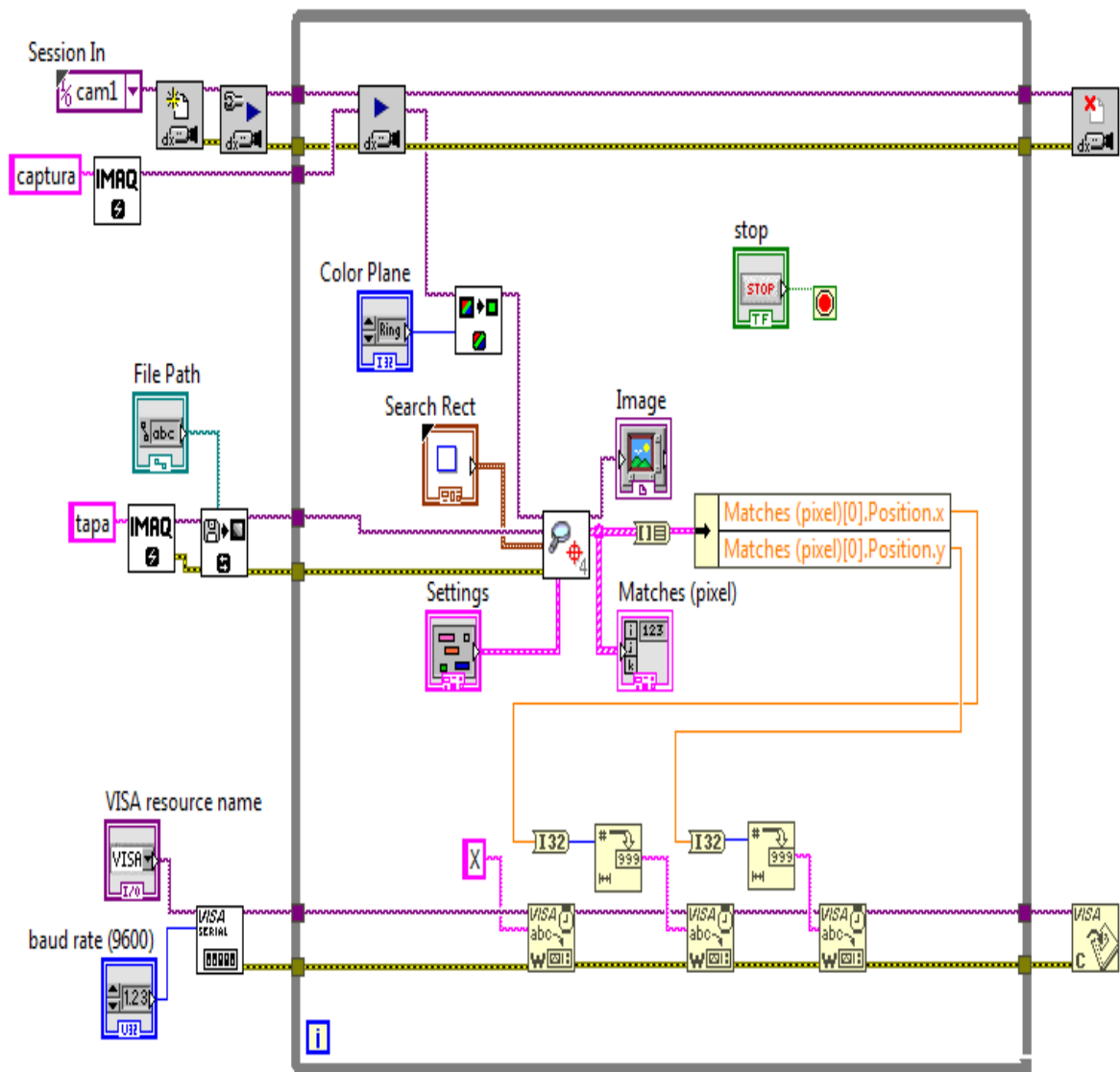
#### q, While Loop



Lặp lại mã trong chương trình con của nó cho đến khi một điều kiện cụ thể xảy ra. Vòng lặp While luôn luôn thực thi ít nhất một lần.

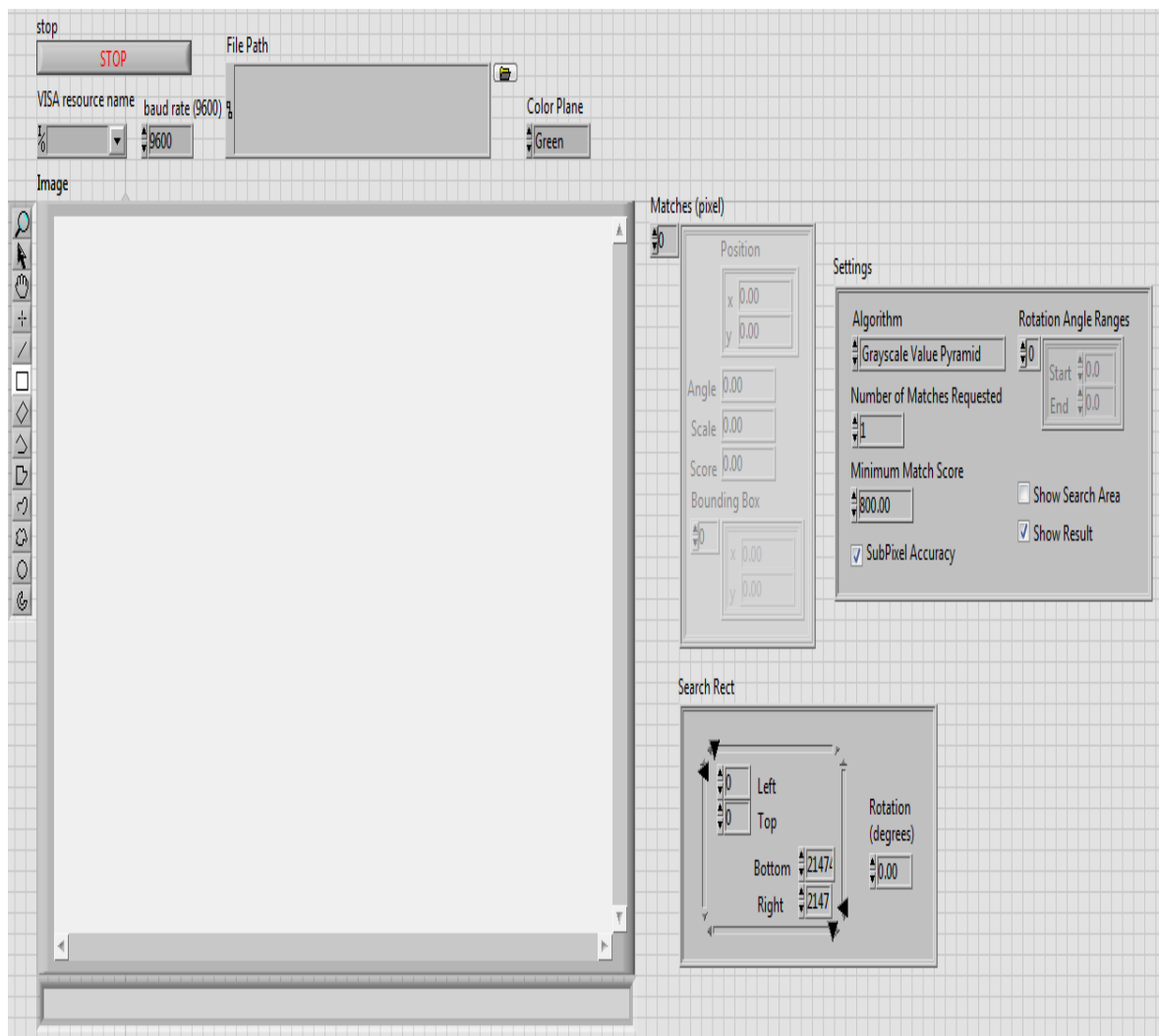
### 3.4.3. Kết nối các khối chức năng

#### a, Sơ đồ khối



Hình 3.3: Sơ đồ kết nối các khối chức năng trên Labview

## b, Giao diện



*Hình 3.4: Giao diện người dùng*

## **CHƯƠNG 4: XÂY DỰNG MÔ HÌNH THỰC TẾ**

### **4.1. Mục đích xây dựng mô hình**

- Mô hình được xây dựng với mục đích bám theo mục tiêu di chuyển trên một tấm chắn đối diện Camera.
- Dữ liệu tọa độ mục tiêu được truyền xuống vi điều khiển để điều khiển cơ cấu chấp hành.
- Cơ cấu chấp hành bao gồm 2 động cơ RC Servo được gắn kết để hoạt động được theo 2 trục độc lập X và Y.
- Trên đầu trục động cơ trục Y có gắn một đèn Laze với mục đích chiếu vào mục tiêu. Nghĩa là mục tiêu di chuyển đi đâu, đèn Laze sẽ chiếu theo đó.

### **4.2. Các thành phần trên mô hình**

#### **4.2.1. Khung mô hình**

Với vật liệu bằng ống kẽm vuông và có kích thước:

$$\text{Dài} \times \text{Rộng} \times \text{Cao} = 60 \times 40 \times 60\text{cm.}$$

#### **4.2.2. Camera**

Loại C720 HD WEBCAM



*Hình 4.1: Camera C720 HD WEBCAM*



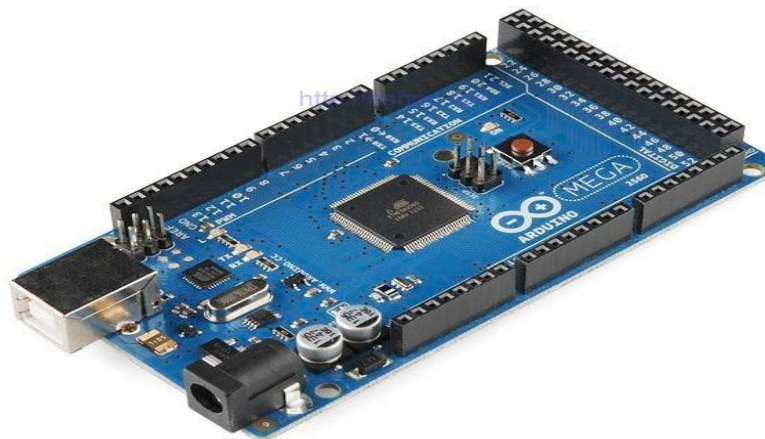
### Các thông số cơ bản

Độ phân giải	720p/30fps	720p/30fps	720p/30fps
Chất lượng video	Tốt hơn	Tốt	Tốt
Tiêu cự	Tự động	Cố định	Cố định
Thấu kính	Tiêu chuẩn	Tiêu chuẩn	Tiêu chuẩn
Trường ngắm	69°	60°	60°

*Hình 4.2: Bảng các thông số Camera C720 HD WEBCAM*

### 4.2.3. Vi điều khiển

Loại Arduino Mega 2560 R3



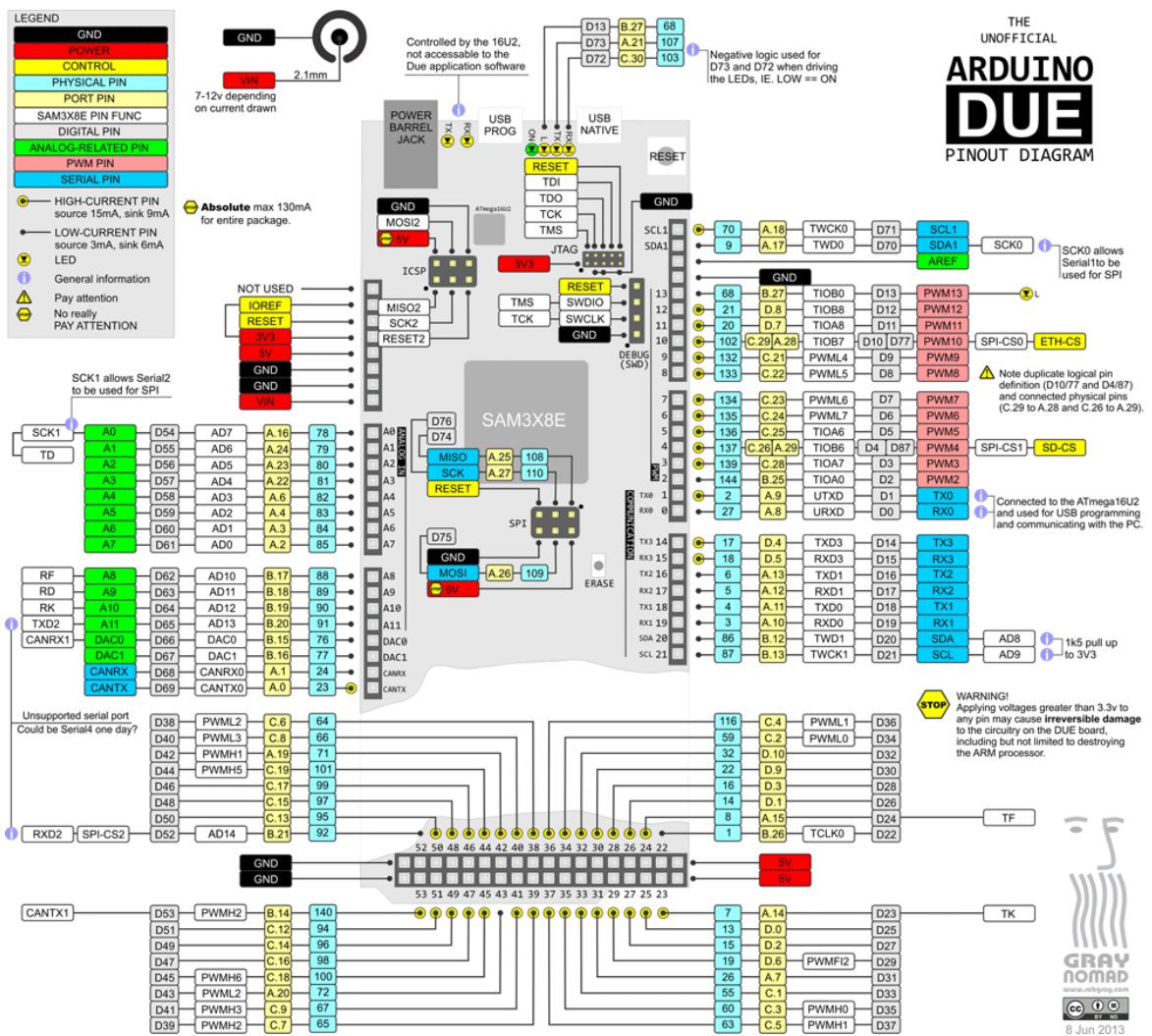
*Hình 4.3: Vi điều khiển Arduino Mega 2560 R3*

Arduino Mega2560 là một trong những bo mạch Arduino lớn hơn. Nó dựa trên kiến trúc AVR của Atmel và thẻ thao MCU ATmega2560. Nó giống như nhiều Arduinos mở rộng bằng cách sử dụng để.

Phản ứng:

MCU	ATmega2560
Family	AVR/ATmega
Vendor	Atmel
RAM	8Kb
Flash	256Kb
Frequency	16MHz
Timers	6 (2x 8bit, 4x 16bit)
ADCs	14 analog input pins (10bit resolution)
UARTs	4
SPIs	1
I2Cs	1 (called TWI)
Vcc	5.0V
Datasheet / Reference Manual	Datasheet and Reference Manual
Board Manual	Board Manual

Hình 4.4: Các thông số của Arduino Mega 2560 R3



Hình 4.5: Sơ đồ chân chức năng Arduino Mega 2560 R3

Trong mô hình ta sử dụng các chân:

- 19(Rx1), 20(Tx1) cho truyền thông RS232 với dữ liệu được gửi xuống từ máy tính thông qua bộ chuyển đổi USB to COM.
- 9, 10 tạo xung PWM điều khiển cho 2 động cơ Servo ứng với hai trục X và Y.

#### **4.2.4. Động cơ RC Servo**

RC servo là một loại động cơ điện đặc biệt có khả năng quay cơ cấu chấp hành tới một vị trí chính xác và giữ cứng tại vị trí đó ngay cả khi cơ cấu chấp hành bị đẩy trở lại. Dải góc quay chuẩn của đầu trục ra thường là 90 và 180 độ. Trên thị trường thế giới có rất nhiều loại servo khác nhau do nhiều nước sản xuất.

Có nhiều cách phân loại servo:

+ Phân loại về nguồn cấp: có servo 1 chiều, servo xoay chiều 1 pha, servo xoay chiều 3 pha.

+ Phân loại về vật liệu làm hộp giảm tốc có: bằng composit, kim loại, hợp kim. Về phương pháp điều khiển, servo có hai loại cơ bản: analog và digital. Bề ngoài thì không có gì khác nhau và về cơ bản, các phần bên trong cũng không phân biệt nhiều ngoại trừ một vài phần điện tử, digital servo có một bộ vi xử lý.

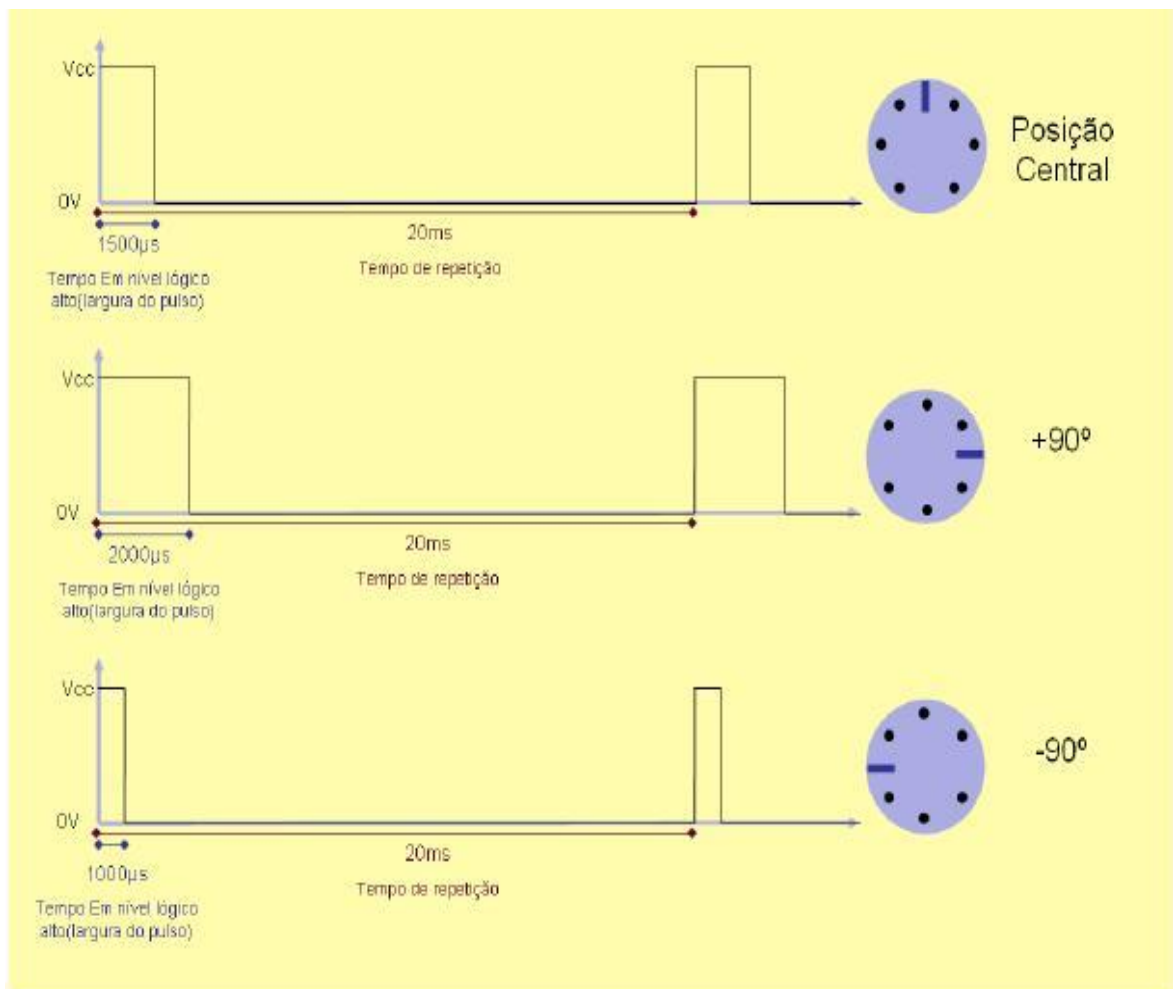
Để có thể hoạt động được thì servo cần 3 dây: một dây nguồn, một dây nối đất và một dây tín hiệu. Thứ tự của các dây là như nhau nhưng màu sắc có thể khác nhau. Dây đỏ thường là dây cấp nguồn (thường là 5V), dây nối đất là dây màu đen và dây tín hiệu vào thường là màu vàng (kiểu S) hoặc màu trắng (kiểu J). Với các loại servo khác, màu sắc có thể thay đổi.



*Hình 4.6: Động cơ RC Servo*

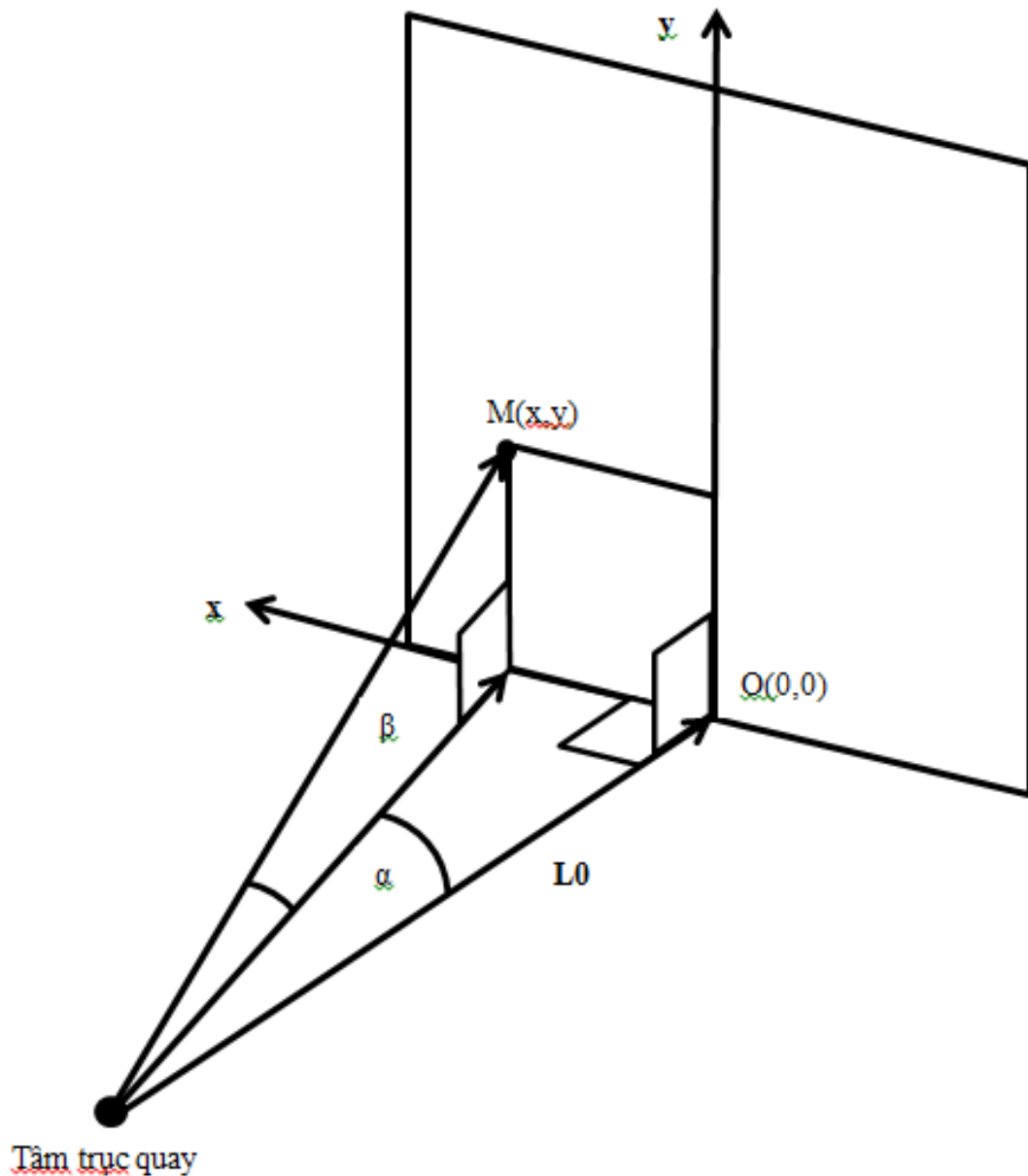
## Điều khiển Servo

Trong các Servo RC hiện đại, góc quay cơ học được xác định bởi chiều rộng của một xung điện phát lên dây điều khiển. Đây là một hình thức điều chế độ rộng xung. Servo RC điển hình dự kiến sẽ thấy một xung mỗi 20 ms, tuy nhiên điều này có thể thay đổi trong một phạm vi rộng khác nhau từ servo này sang servo khác. Chiều rộng của xung sẽ xác định được tốc độ của động cơ. Ví dụ, trong nhiều Servo RC một xung 1,5 ms sẽ làm cho động cơ quay về vị trí 90 độ (vị trí trung tính). Thời gian của giá trị thấp (và tổng thời gian) có thể thay đổi trong phạm vi rộng, và thay đổi từ xung này sang xung kế tiếp, mà không ảnh hưởng đến vị trí của động cơ servo.



Hình 4.7: Xung điều khiển Servo

#### 4.3. Tính toán góc quay của các servo



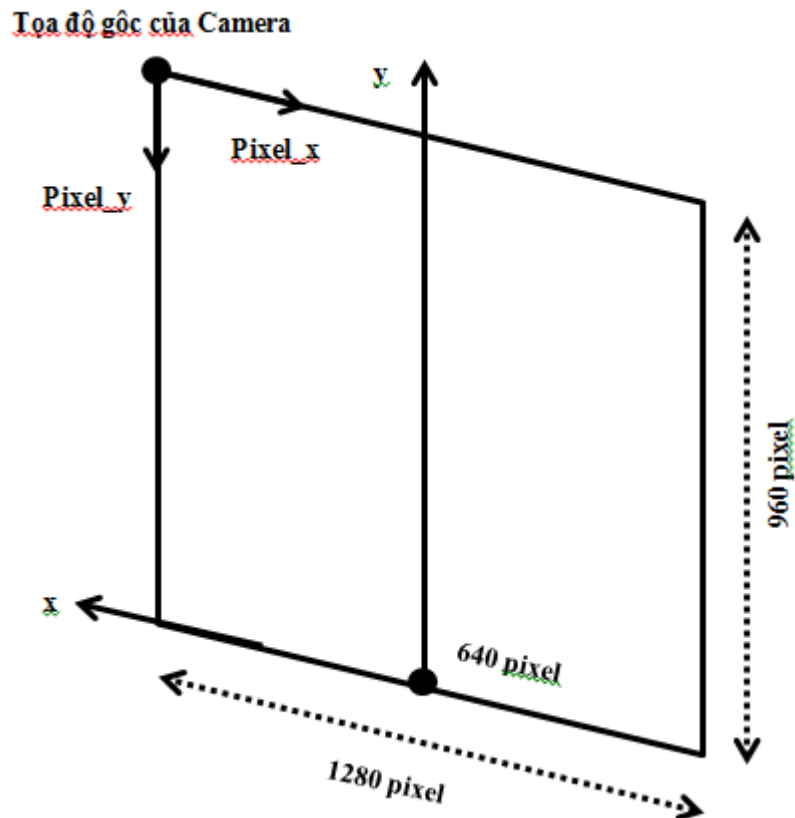
Hình 4.8: Mô tả vị trí của đối tượng thực

- Giả sử tọa độ điểm  $M(x,y)$  là tọa độ của tâm mục tiêu di chuyển lúc này:

$$\alpha = \arctan \frac{x}{L_0}; \beta = \arctan \frac{y}{L_0/\cos(\alpha)};$$

với  $L_0$  là khoảng cách giữa tâm trục quay đến mặt phẳng chứa mục tiêu.

- $\alpha, \beta$  chính là góc cần đưa vào để điều khiển lần lượt Servo cho hai trục  $x, y$ .
- Tham chiếu giữa trục tọa độ đang xét với trục tọa độ theo pixel của Camera



Hình 4.9: Quy chiếu từ tọa độ của Camera sang tọa độ làm việc

- + Hai thành phần  $y$  và **pixel\_y** có mối quan hệ ngược nhau, với độ phân giải **pixel\_y max = 960** thì ta có  $y = 960 - \text{pixel\_y}$ .
- + Với tọa độ góc ở điểm  $x = 0$ , ta lấy **pixel\_x = 640**,  $x = 640 - \text{pixel\_x}$ . Như vậy  $x$  sẽ có cả giá trị âm và dương.
- + Thực tế ta chọn tâm  $O(x,y)$  với các góc quay của Servo tương ứng là 78 và 99°, ta sẽ thay đổi góc từ tọa độ này.

#### 4.4. Chương trình điều khiển trên Arduino Mega 2560

```
#include <Servo.h>

#define L0 1300

Servo myservo_1;
Servo myservo_2;

int PIXEL_X, PIXEL_Y, POS_X, POS_Y, D_X, D_Y, A, B, C, U, V, W, i=0;
double DEC_X, DEC_Y, X, Y;
```

```
char TEXT[50];
void setup() {
  Serial.begin(9600);
  Serial1.begin(9600); // Khởi tạo các cổng nối tiếp với tốc độ Baurate bằng 9600
  delay(1000);
  myservo_1.attach(9);
  myservo_2.attach(10); // Cấu hình các chân điều khiển động cơ RC-Servo
}
void loop() {
  LABVIEW_TERMINAL();
  if(D_X!=POS_X){myservo_1.write(POS_X);D_X = POS_X;}
  if(D_Y!=POS_Y){myservo_2.write(POS_Y);D_Y = POS_Y;}
  // Chỉ cập nhập vị trí mới khi tọa độ thay đổi
}
void LABVIEW_TERMINAL(){
  while(Serial1.available() > 0) { // Kiểm tra xem nếu có dữ liệu truyền tới
    TEXT[i] = Serial1.read(); // Lưu giá trị
    if(TEXT[i]=='X')i=0; // Nếu xuất hiện ký tự "X" bắt đầu lưu tọa độ
    i++;
    if(i == 7){A = TEXT[1] - 48; B = TEXT[2] - 48;
              C = TEXT[3] - 48; U = TEXT[4] - 48;
              V = TEXT[5] - 48; W = TEXT[6] - 48;
    // Các ký tự muốn chuyển về dạng thập phân phải trừ đi 48 theo bảng mã ASCII
    PIXEL_X = A*100+B*10+C; PIXEL_Y = U*100+V*10+W;
    // Tính toán ra giá trị thập phân
    X = 640 - PIXEL_X; Y = 960 - PIXEL_Y;
    // Quy chiếu về gốc tọa độ của Laze
    DEC_X = atan(X/L0)*180/PI;
    DEC_Y = atan(Y/(L0/cos(atan(X/L0))))*180/PI;
```



```
// Tính các góc cần xoay cho RC-Servo
```

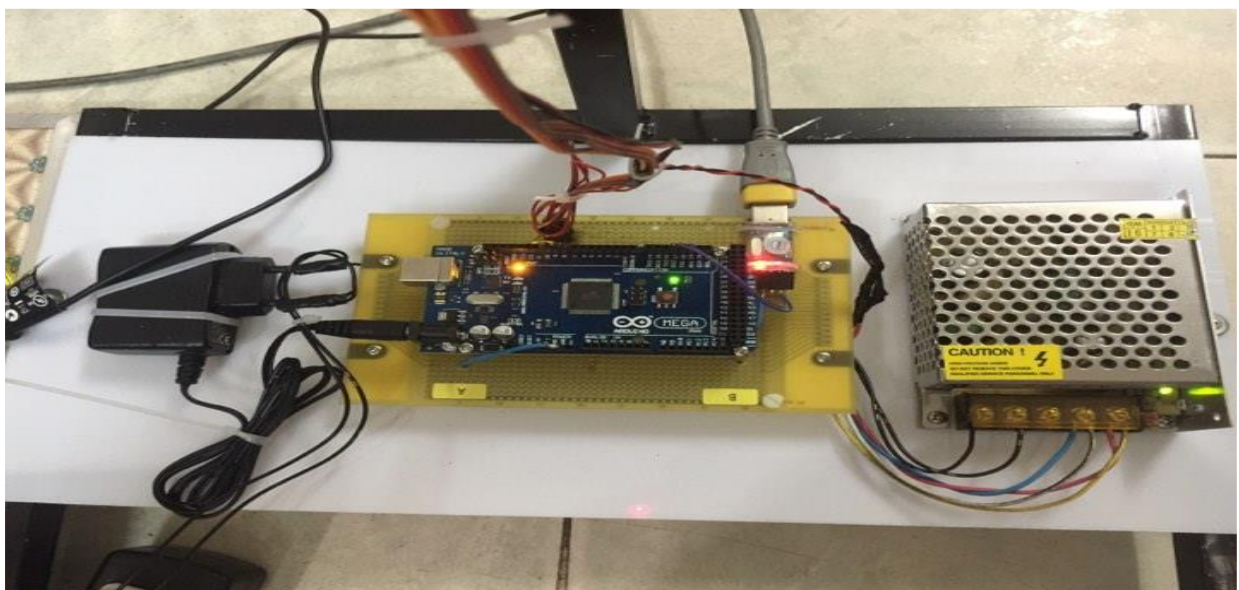
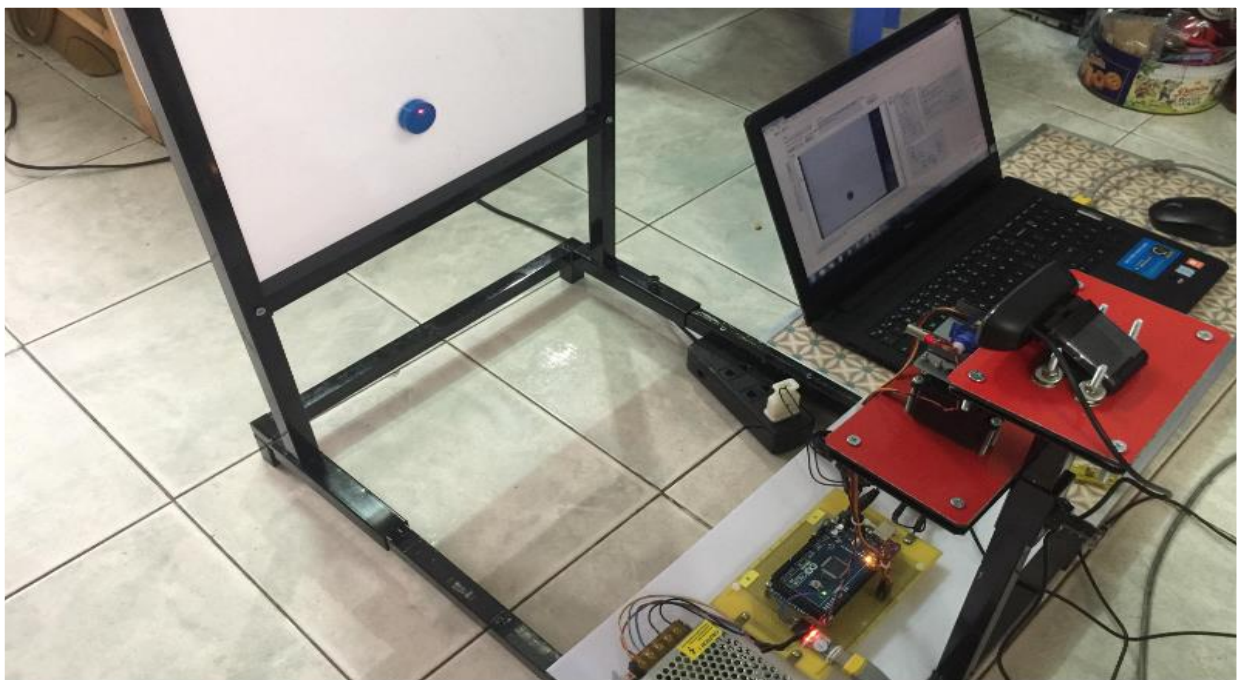
```
POS_X = 78 + DEC_X;
```

```
POS_Y = 99 - DEC_Y; // Cộng với giá trị tương ứng với góc O(0,0)
```

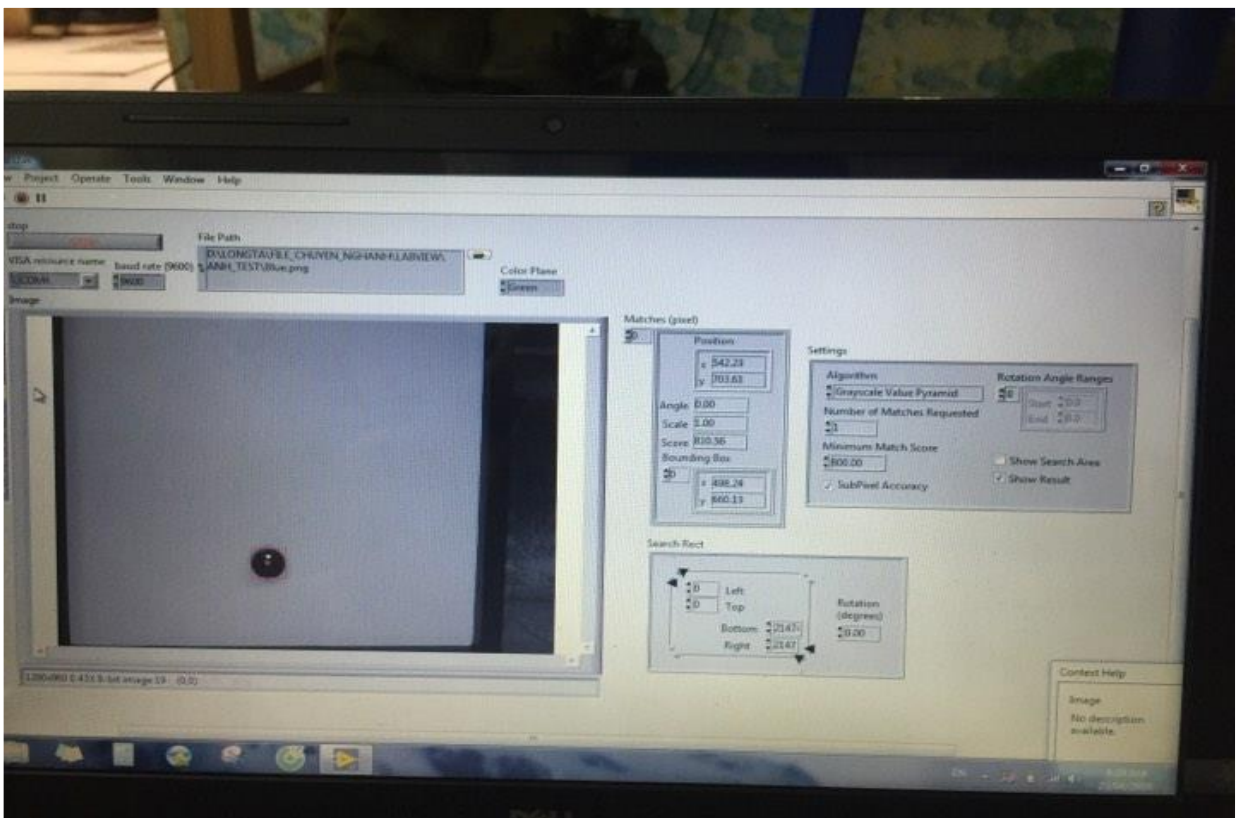
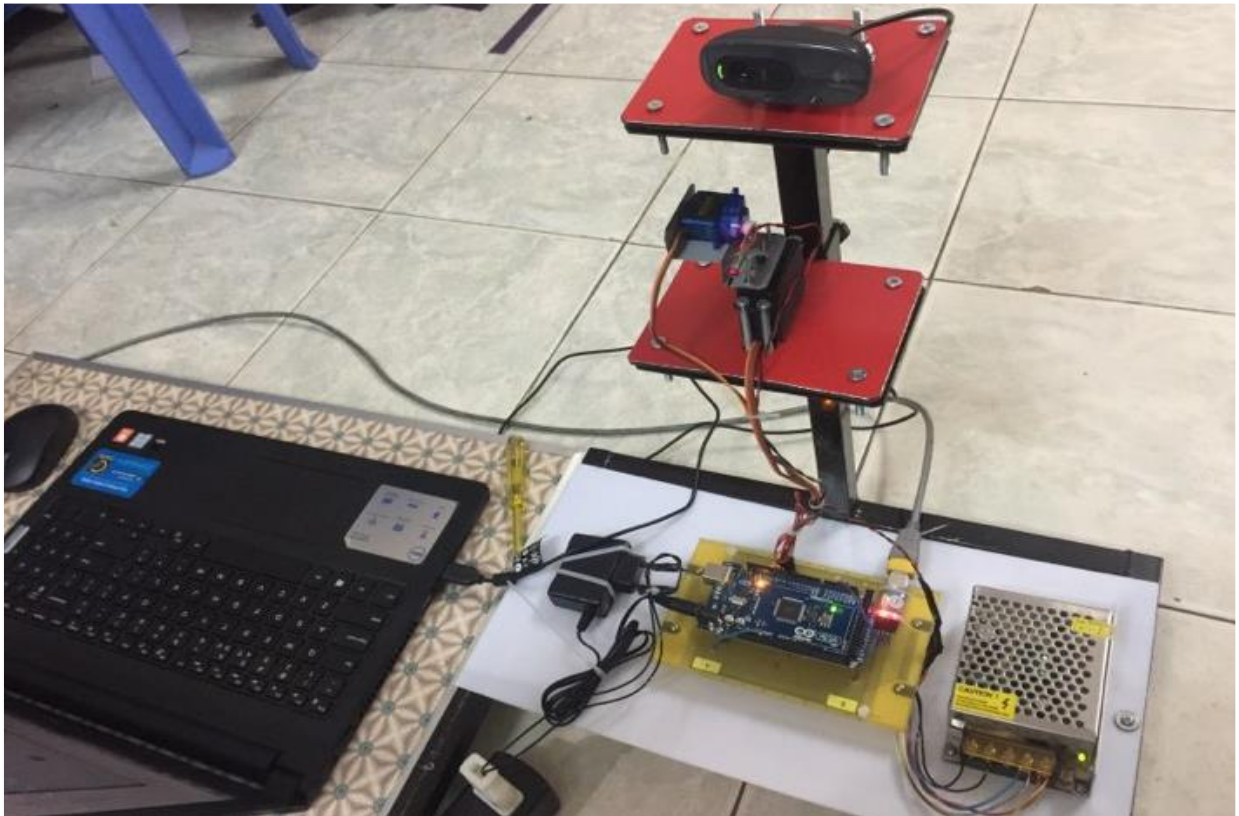
```
}
```

```
}}
```

#### 4.5. Một số hình ảnh mô hình thực tế







*Hình 4.10: Một số hình ảnh mô hình thực tế*

## **KẾT LUẬN VÀ KIẾN NGHỊ**

### **Kết quả chính đã đạt được:**

Luận văn này đã trình bày cơ sở lý thuyết việc nhận dạng, xử lý và điều khiển bằng hình ảnh. Thiết kế module xử lý hình ảnh để điều khiển bám mục tiêu đi sâu nghiên cứu các vấn đề lý thuyết bám bắt mục tiêu di động, cấu trúc điều khiển, các thuật toán xử lý ảnh bám mục tiêu;

Thiết kế, chế tạo được Robot bám mục tiêu di động thông qua xử lý ảnh nhận dạng mục tiêu.

### **Hạn chế của đề tài:**

Thiết kế đề xuất thể hiện nhiều ưu điểm như đã nêu, tuy nhiên bên cạnh đó còn thể hiện các nhược điểm chính sau:

- Nếu màu nền giống màu vật mẫu thì có thể hạn chế việc nhận dạng làm cho không nhận dạng dc.
- Chỉ dùng cho khoảng cách giữa camera và vật mẫu có khoảng cách cố định.

### **Đề xuất hướng nghiên cứu tiếp theo:**

Để giúp cho bài toán được hoàn thiện hơn, các nghiên cứu tới cần quan tâm các vấn đề sau:

- Xây dựng mô hình robot bám mục tiêu sử dụng bánh xe di chuyển với mẫu phức tạp và khoảng cách giữa camera với vật mẫu không cần cố định
- Tiến hành áp dụng vào mô hình trong thực tế sản xuất.

## **TÀI LIỆU THAM KHẢO**

- [1] Phạm Thượng Cát, Trần Việt Phong (2000), “*Nghiên cứu phát triển hệ robot-Camera tự tìm kiếm và bám đối tượng di động*” Tạp chí Cơ điện tử.
- [2] Phạm Thượng Cát, Bùi Trọng Tuyên (2004), “*Một sơ đồ điều khiển hệ hand-eye robot bám mục tiêu di động sử dụng mạng nơron*” Hội nghị toàn quốc lần thứ 2 về Cơ điện tử.
- [3] Ngô Mạnh Tiến (2014), “*Xây dựng robot tự hành dạng nonholonomic và tổng hợp bộ điều khiển bám quỹ đạo*” Luận án tiến sĩ kỹ thuật.
- [4] Nguyễn Văn Tính, Phạm Thượng Cát, Phạm Minh Tuấn, Bùi Thị Thanh Uyên (2011), “*Thiết kế quỹ đạo và điều khiển xe tự hành vận chuyển trong kho VCCA*”, Luận án tiến sĩ kỹ thuật.
- [5] Bùi Trọng Tuyên, Phạm Thượng Cát (2002), “*Về một phương pháp điều khiển hệ Camera-robot bám mục tiêu sử dụng mạng nơron*” Hội nghị lần thứ nhất về cơ điện tử.
- [6] Aubery Marchel Tientcheu (2016), “*Formation control for Multirobot system*”, by CRC Press Reference - 300 Pages - 80 B/W Illustrations ISBN 9781466501423.
- [7] Cai Ze-su, Zhao Jie, and Cao Jian (2012), “*Formation Control and Obstacle Avoidance for Multiple Robots Subject to Wheel-Slip*”, International Journal of Advanced Robotic Systems.
- [8] D. Lee, O. Martinez-Palafox, and M. W. Spong (2006), “*Passive bilateral teleoperation of a wheeled mobile robot over a delayed communication network*”, in Proceedings of the IEEE international conference on robotics and automation, pp. 3298-3303.

- [9] Dongjun Lee and Mark W. Spong (2005), “*Passive bilateral control of teleoperators under constant time-delay.*” Proc. of the 16th IFAC World Congress, Czech Republic.
- [10] J. Park, R. Cortesao, and O. Khatib (2010), “*Robust and Adaptive Teleoperation for Compliant Motion Tasks*”. Stanford University, Robotics Group, 94305-9010 CA, USA.
- [11] John T. Feddema, Chris Lewis and David A. Schoenwald (2002), “*Decentralized Control of Cooperative Robotic, Vehicles: Theory and Application*”, IEEE transactions on robotics and automation, Vol 18, No. 5, October.
- [12] L. E. Parker (2000), “*Current state of the art in distributed autonomous mobile robotics,*” in Distributed Autonomous Robotic Systems 4, L. E. Parker, G. Bekey, and J. Barhen, Eds. New York: Springer-Verlag.
- [13] Nicolas Jouandeau<sup>1</sup> and Arab Ali Cherif (2012), “*A Survey and Analysis of Multi-Robot Coordination*”, Paris 8 University, Saint-Denis, France.