****

**BỘ THÔNG TIN VÀ TRUYỀN THÔNG**

**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

**KHOA: KỸ THUẬT ĐIỆN TỬ I**

-----🙞🙜🕮🙞🙜-----



**ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC**

**Đề tài: “Robot bám đối tượng”**

|  |  |
| --- | --- |
| Giảng viên hướng dẫn: | Nguyễn Quốc Dinh |
| Sinh viên thực hiện: | Phạm Hoàng Dự |
| Lớp : | D13XLTH |
| Hệ: | Chính Quy |

**Hà nội, năm 2017**

**NHẬN XÉT, ĐÁNH GIÁ, CHO ĐIỂM**

**(Của giảng viên hướng dẫn)**

**Điểm:** *(Bằng chữ:……………………………..)*

**Đồng ý/Không đồng ý** cho sinh viên bảo vệ trước hội đồng chấm đồ án tốt nghiệp…?

|  |  |
| --- | --- |
|  | *Hà Nội, Ngày…...Tháng…...Năm 2017*  CÁN BỘ - GIẢNG VIÊN HƯỚNG DẪN  *(ký, họ tên)* |

# LỜI CẢM ƠN

Trong thời gian làm đồ án tốt nghiệp, em đã nhận được sự giúp đỡ, đóng góp ý kiến và chỉ bảo nhiệt tình từ quý thầy cô, gia đình và bạn bè.

Trước hết em xin gửi tới thầy cô đang công tác tại khoa Kỹ Thuật Điện Tử 1 – Học viện Công nghệ Bưu chính Viễn thông lời chào trân trọng, lời chúc sức khỏe và lời cảm ơn sâu sắc. Với sự quan tâm, dạy dỗ,chỉ bảo chu đáo tận tình của thầy cô đến nay em đã có thể hoàn thành đồ án.

Đặc biệt em xin gửi lời cảm ơn chân thành nhất tới thầy giáo - THS. Nguyễn Quốc Dinh đã quan tâm giúp đỡ, hướng dẫn em hoàn thành tốt đồ án trong thời gian qua.

Với điều kiện thời gian cũng như kinh nghiệm còn hạn chế của một học viên, đồ án này không thể tránh được những thiếu sót. Em rất mong nhận được sự đóng góp từ quý thầy cô để có thêm kinh nghiệm trong việc nghiên cứu và phát triển sản phẩm trong tương lai.

Em xin chân thành cảm ơn.

*Hà Nội, ngày 10 tháng 12 năm 2017*

**Sinh viên thực hiện**

**Phạm Hoàng Dự**

MỤC LỤC

[LỜI CẢM ƠN i](#_Toc500981690)

[DANH MỤC HÌNH ẢNH iv](#_Toc500981691)

[DANH MỤC BẢNG BIỂU v](#_Toc500981692)

[LỜI MỞ ĐẦU 1](#_Toc500981693)

[CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI 2](#_Toc500981694)

[1.1. Mục tiêu 2](#_Toc500981695)

[1.2. Giới thiệu chung về robot 2](#_Toc500981696)

[*1.2.1* *Lịch sử phát triển robot* 2](#_Toc500981697)

[*1.2.2* *Các thế hệ robot* 5](#_Toc500981698)

[*1.2.3* *Giới thiệu robot bám đối tượng* 5](#_Toc500981699)

[1.3 Giới thiệu sơ lược về xử lý ảnh 6](#_Toc500981700)

[1.4 Khái niệm cơ bản về xử lý ảnh 10](#_Toc500981701)

[*1.4.1* *Phần tử ảnh* 10](#_Toc500981702)

[*1.4.2* *Ảnh* 11](#_Toc500981703)

[*1.4.3* *Mức xám* 11](#_Toc500981704)

[*1.4.4* *Ảnh đen trắng* 12](#_Toc500981705)

[*1.4.5* *Ảnh màu* 12](#_Toc500981706)

[1.5 Các vấn đề cơ bản trong xử lý ảnh 12](#_Toc500981707)

[1.6 Không gian màu 14](#_Toc500981708)

[*1.6.1* *Không gian màu RGB* 14](#_Toc500981709)

[*1.6.2.* *Không gian màu HSV* 15](#_Toc500981710)

[*1.6.3* *Không gian màu CMYK* 16](#_Toc500981711)

[*1.6.4* *Chuyển đổi giữa các không gian màu* 17](#_Toc500981712)

[1.7 Một số kỹ thuật tiền xử lý ảnh 18](#_Toc500981713)

[*1.7.1* *Chuyển ảnh xám* 18](#_Toc500981714)

[*1.7.2* *Lọc nhiễu* 18](#_Toc500981715)

[*1.7.3* *Nhị phân ảnh* 20](#_Toc500981716)

[*1.7.4* *Lược đồ mức xám (Histogram)* 21](#_Toc500981717)

[*1.7.5* *Phóng to, thu nhỏ ảnh* 22](#_Toc500981718)

[*1.7.6. Kỹ thuật phân vùng đối tượng* 22](#_Toc500981719)

[CHƯƠNG 2: CÔNG CỤ THIẾT KẾ SẢN PHẨM 30](#_Toc500981720)

[2.1. Giới thiệu module ESP8266 NodeMCU 30](#_Toc500981721)

[2.2. Module điều khiển động cơ L298 33](#_Toc500981722)

[2.3. Giới thiệu về OpenCV 38](#_Toc500981723)

[*2.3.1. Giới thiệu về thư viện OpenCV* 38](#_Toc500981724)

[*2.3.2. Một số hàm xử lý ảnh nhận dạng cơ bản của OpenCV* 41](#_Toc500981725)

[*2.3.3. Chuyển đổi các không gian màu trong OpenCV* 44](#_Toc500981726)

[CHƯƠNG 3: XỬ LÍ HÌNH ẢNH VÀ PHÂN LOẠI MÀU 45](#_Toc500981727)

[3.1. Phát hiện đối tượng 45](#_Toc500981728)

[*3.1.1. Phương pháp phân loại vật dựa trên màu sắc* 45](#_Toc500981729)

[3.2. Phát hiện tọa độ của đối tượng có màu cần tìm 46](#_Toc500981730)

[*3.2.1 Quá trình phát hiện tọa độ của đối tượng 46*](#_Toc500981731)

[*3.2.2 Loại bỏ nhiễu cho ảnh 48*](#_Toc500981732)

[CHƯƠNG 4: THIẾT KẾ VÀ THI CÔNG HỆ THỐNG 50](#_Toc500981733)

[4.1. Thiết kế mô hình 50](#_Toc500981734)

[4.2. Hệ thống điều khiển 51](#_Toc500981735)

[KẾT LUẬN. 53](#_Toc500981736)

[TÀI LIỆU THAM KHẢO 54](#_Toc500981737)

[Danh mục tài liệu tham khảo 54](#_Toc500981738)

[Danh mục website tham khảo 54](#_Toc500981739)

# DANH MỤC HÌNH ẢNH

[Hình 1. 1 : Robot 4 chân của hãng R.S Mosher và hãng General Electric 3](#_Toc500981740)

[Hình 1. 2 : Xe tự hành thám hiểm mặt trăng (Lunokohod 1) 3](#_Toc500981741)

[Hình 1. 3 : Robot Shakey-robot đầu tiên nhận dạng đối tượng bằng camera 4](#_Toc500981742)

[Hình 1. 4 : Tay robot trên tàu thám hiểm Viking 1 4](#_Toc500981743)

[Hình 1. 5 : Các bước cơ bản trong xử lý ảnh 7](#_Toc500981744)

[Hình 1. 6 : Sơ đồ phân tích và xử lý ảnh và lưu đồ thông tin giữa các khối 10](#_Toc500981745)

[Hình 1. 7 : Thuật toán tìm đường biên 13](#_Toc500981746)

[Hình 1. 8 : Nhận dạng màu sắc 14](#_Toc500981747)

[Hình 1. 9 : Không gian màu RGB 15](#_Toc500981748)

[Hình 1. 10 : Không gian màu HSV 15](#_Toc500981749)

[Hình 1. 11 : Hình tròn biểu diễn màu sắc (Hue) 16](#_Toc500981750)

[Hình 1. 12 : Không gian màu CMYK 17](#_Toc500981751)

[Hình 1. 13 : Mặt nạ thông thấp 19](#_Toc500981752)

[Hình 1. 14 : Lược đồ mức xám của các loại ảnh 22](#_Toc500981753)

[Hình 1. 15 : Lược đồ xám của ảnh 23](#_Toc500981754)

[Hình 1. 16 : Thuật toán tam giác 24](#_Toc500981755)

[Hình 1. 17 : Các điểm lân cận gradient 28](#_Toc500981756)

[Hình 1. 18 : Làm mảnh chữ 28](#_Toc500981757)

[Hình 1. 19 : Căn chỉnh đường biên 29](#_Toc500981758)

[Hình 2. 1 : Hình ảnh module ESP8266 thực tế 30](#_Toc500981759)

[Hình 2. 2 : Hình ảnh thực tế ESP NodeMCU 31](#_Toc500981760)

[Hình 2. 3 : Sơ đồ chân Pin trên ESP8266 NodeMCU 32](#_Toc500981761)

[Hình 2. 4 : Mạch cầu H 34](#_Toc500981762)

[Hình 2. 5 : Nguyên lý hoạt động mạch cầu H 35](#_Toc500981763)

[Hình 2. 6 : Sơ đồ nguyên lí module L298 36](#_Toc500981764)

[Hình 2. 7 : Hình ảnh thực tế module L298 37](#_Toc500981765)

[Hình 2. 8 : Cấu trúc cơ bản của OpenCV 39](#_Toc500981766)

[Hình 2. 9 : Một số ứng dụng của OpenCV 40](#_Toc500981767)

[Hình 2. 10 : Xác định và nhận diện khuôn mặt trong ảnh 40](#_Toc500981768)

[Hình 3. 1 : Lưu đồ giải thuật nhận dạng 45](#_Toc500981769)

[Hình 3. 2 : Hình gốc dùng để nhận dạng màu 47](#_Toc500981770)

[Hình 3. 3 : Hình sau khi được chuyển sang mã màu HSV 47](#_Toc500981771)

[Hình 3. 4 : Hình thể hiện vùng có màu đỏ(chấm đen là tâm của ảnh mỗi vật , toạ độ của chấm den được coi như là toạ độ của vật đó trong hình ) 48](#_Toc500981772)

[Hình 3. 5 : Hình thể hiện vùng có màu xanh da trời (chấm đen là tâm của ảnh mỗi vật , toạ độ của chấm den được coi như là toạ độ của vật đó trong hình ) 48](#_Toc500981773)

[Hình 3. 6 : Hình thể hiện vùng có màu xanh lá cây (chấm đen là tâm của ảnh mỗi vật , toạ độ của chấm den được coi như là toạ độ của vật đó trong hình ) 48](#_Toc500981774)

[Hình 3. 7 : Ăn mòn và giãn nở ảnh 49](#_Toc500981775)

[Hình 4. 1 : Mô hình thực tế 50](#_Toc500981780)

[Hình 4. 2 : Tổng quan hoạt động 51](file:///D:\Download\DATN_RobotBamDoiTuong_PhamHoangDu.docx#_Toc500981781)

[Hình 4. 3 : Giao diện phần mềm 51](#_Toc500981782)

[Hình 4. 4 : Giao diện xử lý ảnh 52](#_Toc500981783)

# DANH MỤC BẢNG BIỂU

[Bảng 2.1: Bảng trạng thái hoạt động của module WiFi ESP8266 8](#_Toc469347242)

[Bảng 3.1: So sánh mô hình TCP/IP với OSI 21](#_Toc469347243)

[Bảng 3.2: Kiến trúc giao thức MQTT 27](#_Toc469347244)

[Bảng 3.3: Connect Packet Fixed Header 28](#_Toc469347245)

[Bảng 3.4: Protocol Name 28](#_Toc469347246)

[Bảng 3.5: Cờ trạng thái 29](#_Toc469347247)

[Bảng 3.6: Header của gói tin Publish 30](#_Toc469347248)

[Bảng 3.7: Phản hồi trạng thái của gói tin Publish tương ứng với từng trạng thái QoS 30](#_Toc469347249)

[Bảng 3.8: Header của gói tin Subscribe 31](#_Toc469347250)

[Bảng 3.9: Định dạng gói tin Subscribe 31](#_Toc469347251)

[Bảng 3.10: Các thuộc tính của Websocket 33](#_Toc469347252)

[Bảng 3.11: Các sự kiện của Websocket 33](#_Toc469347253)

[Bảng 3.12: Phương thức của Websocket 34](#_Toc469347254)

# LỜI MỞ ĐẦU

Trong thời đại công nghiệp ngày nay, robot ngày càng được sử dụng phổ biến trong sản xuất cũng như trong cuộc sống hằng ngày. Robot đã có một vị trí quan trọng khó có thể thay thế được, nó giúp con người trong các công việc nguy hiểm khó khăn.  Ngoài ra còn được sử dụng trong các lĩnh vực thám hiểm không gian, quân sự, giải trí… Lĩnh vực robot ngày càng chiếm được sự quan tâm của các nhà nghiên cứu và xã hội . Trong đó, hệ thống cảm biến, nhận dạng là không thể thiếu. Ngày nay hệ thống nhận dạng càng trở nên phong phú với nhiều sự nhận dạng khác nhau cũng như sự tích hợp càng nhiều những công nghệ kỹ thuật cao. Xử lý ảnh là một trong những hệ thống nhận dạng giúp robot nhận thức thế giới chung quanh một cách đầy đủ nhất.

Xử lý ảnh bao gồm lý thuyết và các kỹ thuật liên quan nhằm mục đích tạo ra một hệ thống nhân tạo có thể tiếp nhận thông tin từ các hình ảnh thu được hoặc các tập dữ liệu đa chiều. Đối với mỗi người chúng ta, quá trình nhận thức thế giới bên ngoài là một điều dễ dàng. Quá trình nhận thức đó được “học” thông qua quá trình sống của mỗi người. Tuy nhiên với các vật vô tri vô giác như như các máy tính, robotv..v. thì điều đó quả thực là một bước tiến rất gian nan. Các thiết bị ngày nay không chỉ nhận thông tin ở dạng tín hiệu đơn lẻ mà còn có thể có cái “nhìn” thật với thế giới bên ngoài. Cái “nhìn” này qua quá trình phân tích, kết hợp với các mô hình như máy học, mạng nơron v..v, sẽ giúp cho thiết bị tiến dần tới một hệ thống nhân tạo có khả năng ra quyết định linh hoạt và đúng đắn hơn rất nhiều. Với mong muốn được tham gia, tìm hiểu về lĩnh vực này em đã thực hiện đề tài “Robot bám đối tượng di động”.

Nội dung của đồ án bao gồm 4 chương:

* ***Chương 1: Tổng quan về đề tài.***
* ***Chương 2: Công cụ thiết kế sản phẩm.***
* ***Chương 3: Xử lý hình ảnh và phân loại màu.***
* ***Chương 4: Thiết kế và thi công hệ thông.***

# CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI

## Mục tiêu

Tìm hiểu các phương pháp về xử lý hình ảnh.

Sử dụng camera để nhận dạng đối tượng theo màu sắc rồi gửi về máy tính vị trí của đối tượng.Máy tính nhận được dữ liệu vị trí và màu sắc của đối tượng sẽ bám theo và giữ khoảng cách nhất định với đối tượng.

## Giới thiệu chung về robot

Thuật ngữ robot được sinh ra từ trên sân khấu, không phải trong phân  xưởng sản xuất.  Những robot xuất hiện lần đầu tiên trên ở trên NewYork vào ngày 09/10/1922 trong vở “Rossum’s Universal Robot” của nhà soạn kịch người Tiệp Karen Kapek viết năm 1921, còn từ robot là cách gọi tắt của từ robota - theo tiếng Tiệp có nghĩa là công việc lao dịch.

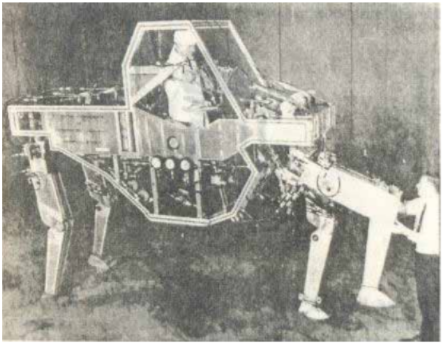
Robot hay người máy là một loại máy có thể thực hiện những công việc một cách tự động bằng sự điều khiển của máy tính. Robot là một tác nhân cơ khí, nhân tạo, thường là một hệ thống cơ khí-điện tử.

Với sự xuất hiện và chuyển động của mình, robot gây cho người ta cảm giác rằng nó giác quan giống như con người. Từ "robot" (người máy) thường được hiểu với hai nghĩa: rô bốt cơ khí và phần mềm tự hoạt động. Về lĩnh vực người máy, Nhật Bản là nước đi đầu thế giới về lĩnh vực này.

Ngày nay robot đang là tâm điểm của một cuộc cách mạng lớn sau Internet. Robot ngày càng được sử dụng rộng rãi trong lĩnh vực công nghiệp, y tế, giáo dục đào tạo, giải trí, an ninh quốc phòng, thám hiểm không gian. Robot là sản phẩm công nghệ có độ phức tạp cao, chứa hàm lượng tri thức vô cùng phong phú về tất cả các lĩnh vực của khoa học và công nghệ.

### *Lịch sử phát triển robot*

Vào năm 1948, nhà nghiên cứu Goertz đã nghiên cứu chế tạo loại tay máy đôi điều khiển từ xa đầu tiên, và cùng năm đó hãng General Mills chế tạo tay máy gần tương tự sử dụng cơ cấu tác động là những động cơ điện kết hợp với các cử hành trình. Đến năm 1954, Goertz tiếp tục chế tạo một dạng tay máy đôi sử dụng động cơ servo và có thể nhận biết lực tác động lên khâu cuối. Sử  dụng những thành quả đó, vào năm 1956 hãng General Mills cho ra đời tay máy hoạt động trong công việc khảo sát đáy biển. Năm 1968 R.S. Mosher, thuộc hãng General Electric, đã chế tạo một thiết bị biết đi có bốn chân, có chiều dài hơn 3m, nặng 1.400kg, sử dụng động cơ đốt trong  có công suất gắn 100 mã lực.



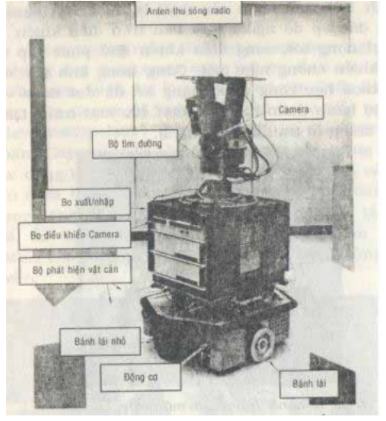
*Hình 1. 1 : Robot 4 chân của hãng R.S Mosher và hãng General Electric*

Cũng trong lĩnh vực này, một thành tựu khoa học công nghệ đáng kể đã đạt được vào năm 1970 là xe tự hành thám hiểm bề mặt của mặt trăng Lunokohod 1 được điều khiển từ trái đất.



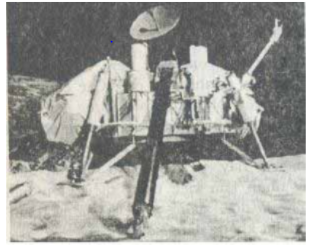
*Hình 1. 2 : Xe tự hành thám hiểm mặt trăng (Lunokohod 1)*

Viện nghiên cứu thuộc Trường Đại học Stanford vào năm 1969 đã thiết kế robot Shakey di động tinh vi hơn để thực hiện những thí nghiệm về điều khiển sử dụng hệ thống thu nhận hình ảnh để nhận dạng đối tượng. Robot này được lập trình trước để nhận dạng đối tượng bằng camera, xác định đường đi đến đối tượng và thực hiện một số tác động trên đối tượng.



*Hình 1. 3 : Robot Shakey-robot đầu tiên nhận dạng đối tượng bằng camera*

Năm 1952 máy điều khiển chương trình số đầu tiên ra đời tại học viện công nghệ Massachusetts (Hoa Kỳ). Trên cơ sở đó năm 1954, George Devol đã thiết kế robot lập trình với điều khiển chương trình số đầu tiên nhờ một thiết bị do ông phát minh được gọi là thiết bị chuyển khớp được lập trình. Joseph Engelberger, người mà ngày nay thường được gọi là cha đẻ của robot công nghiệp, đã thành lập hãng Unimation sau khi mua bản quyền thiết bị của Devol và sau đó đã phát triển những thế hệ robot điều khiển theo chương trình. Năm 1962, robot Unmation đầu tiên được đưa vào sử dụng tại hãng General Motors; và năm 1976 cánh tay robot  đầu tiên trong không gian đã được sử dụng trên tàu thám hiểm Viking của cơ quan  Không Gian NASA của Hoa Kỳ để lấy mẫu đất trên sao Hoả.



*Hình 1. 4 : Tay robot trên tàu thám hiểm Viking 1*

Trong hoạt động sản xuất, đa số những robot công nghiệp có hình dạng của “cánh tay cơ khí”,cũng chính vì vậy mà đôi  khi ta gặp thuật ngữ người máy - tay máy trong tài liệu tham khảo và giáo trình về robot.

### *1.2.2 Các thế hệ robot*

Kể từ khi khái niệm robot ra đời, việc thiết kế và chế tạo robot đã trải qua nhiều giai đoạn với nhiều thế hệ khác nhau. Có 5 thế hệ robot ra đời kể từ năm 1960 .

* Thế hệ thứ nhất: Bao gồm các loại robot hoạt động lặp lại theo một chu trình không thay đổi. Chương trình điều khiển có hai dạng:
  + Chương trình “cứng”: không thay đổi hoặc không sửa được trừ khi thay đổi phần cứng.
  + Chương trình có thể thay đổi được thông qua các panel điều khiển hoặc thông qua máy tính. Các robot thế hệ này sử dụng cơ cấu điều khiển servo vòng hở (open-loop nonservo controlled system). Đây là hệ thống không sử dụng thông tin phản hồi từ môi trường về để điều khiển robot.
* Thế hệ thứ hai: Robot được trang bị các sensor cho phép robot giao tiếp với môi trường bên ngoài. Các thiết bị này thực chất là các bộ biến đổi năng lượng. Nó chuyển các đại lượng không điện thành đại lượng điện mà qua đó bộ điều khiển robot có thể biết được trạng thái của môi trường xung quanh nó. Nhờ các sensor này robot có thể chọn các phương án khác nhau một cách linh hoạt nhằm thích nghi với môi trường bên ngoài. Dạng robot với trình độ điều khiển này còn được gọi là robot điều khiển thích nghi cấp thấp. Đây gọi là cơ cấu điều khiển servo vòng kín (closed-loop servo controller system).
* Thế hệ thứ ba: các bộ điều khiển logic khả trình PLC (Programmable Logic Controller) được sử dụng trong robot với nhiều chức năng chuyên biệt .
* Thế hệ thứ bốn***:*** Khác với PLC bị giới hạn trong chương trình của chúng, thế hệ robot này sử dụng các máy tính được trang bị các ngôn ngữ lập trình đặc biệt hoặc ngôn ngữ chuẩn như Basic, C, C++....,để  tạo ra nhiều ứng dụng CAD/CAM và CIM hoặc chương trình không trực tuyến.
* Thế hệ thứ năm: Các bộ điều khiển của robot sử dụng trí tuệ nhân tạo (artificial intelligence).  Robot được trang bị các kỹ thuật như nhận dạng tiếng nói, hình ảnh, xác định khoảng cách, cảm nhận đối tượng tiếp xúc (da nhân tạo) để  xử lý, ra những quyết định hợp lý.

### *1.2.3 Giới thiệu robot bám đối tượng*

Hệ thống điều khiển robot có thị giác được sử dụng để theo dõi mục tiêu di động trong các ứng dụng công nghiệp, quốc phòng và xã hội. Hệ thống thị giác có thể được phân thành 2 lớp, theo như cấu trúc của hệ, đó là hệ có camera gắn cố định và hệ có camera gắn trên tay máy (eye-in-hand). Trong hệ camera cố định, camera được gắn cố định so với hệ toạ độ thực, thu thập ảnh của cả mục  tiêu và cả môi trường. Mục tiêu của hệ này là cung cấp tín hiệu điều khiển sao cho robot bám theo đối tượng. Trên cơ sở ảnh thu được từ camera, được số hoá và tích hợp trong vòng điều khiển phản hồi.

Các nghiên cứu của môi trường động bị bỏ khá xa so với của môi trường tĩnh do bị ảnh hưởng khá  lớn của tốc độ tính toán cũng như độ chính xác của việc phân tích ảnh. Do vậy có nhiều thuật toán để cải thiện tốc độ xử lý và xem xét đến các quà trình nhiễu tác động lên quá trình thu thập hình ảnh.

Một thách thức của thị giác máy là việc phân loại đối  tượng. Một robot có thể phải đối mặt với nhiều đối tượng khác nhau, trong khi đó chỉ có một mục tiêu quan tâm, còn các đối tượng khác thì không.  Để nhận biết được thông tin về vị trí của mục tiêu trong môi trường động, cácđặc trưng của mục tiêu rất quan trọng. Các điểm lỗ, các góc cạnh, các đặc điểm hình học của mục tiêu được phân tích qua quá trình nhận dạng. Đặc tính về trọng tâm của mục tiêu có thể tính dễ dàng qua moment bâc nhất trong môi trường tĩnh, nhưng trong môi trường động việc này rất khó vì chi phí thời gian tính toán lớn.

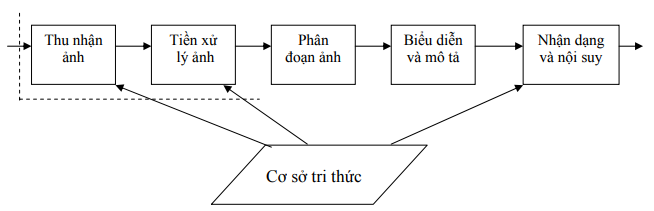
## Giới thiệu sơ lược về xử lý ảnh

Xử lý ảnh là một lĩnh vực mang tính khoa học và công nghệ. Nó là một ngành khoa học mới mẻ so với nhiều ngành khoa học khác nhưng tốc độ phát triển của nó rất nhanh, kích thích các trung tâm nghiên cứu, ứng dụng, đặc biệt là máy tính chuyên dụng riêng cho nó.

Xử lý ảnh được đưa vào giảng dạy ở bậc đại học ở nước ta khoảng chục năm nay. Nó là môn học liên quan đến nhiều lĩnh vực và cần nhiều kiến thức cơ sở khác. Đầu tiên phải kể đến Xử lý tín hiệu số là một môn học hết sức cơ bản cho xử lý tín hiệu chung, các khái niệm về tích chập, các biến đổi Fourier, biến đổi Laplace, các bộ lọc hữu hạn… Thứ hai, các công cụ toán như Đại số tuyến tính, Sác xuất, thống kê. Một số kiến thứ cần thiết như Trí tuệ nhân tao, Mạng nơ ron nhân tạo cũng được đề cập trong quá trình phân tích và nhận dạng ảnh.

Các phương pháp xử lý ảnh bắt đầu từ các ứng dụng chính: nâng cao chất lượng ảnh và phân tích ảnh. Ứng dụng đầu tiên được biết đến là nâng cao chất lượng ảnh báo được truyền qua cáp từ Luân đôn đến New York từ những năm 1920. Vấn đề nâng cao chất lượng ảnh có liên quan tới phân bố mức sáng và độ phân giải của ảnh. Việc nâng cao chất lượng ảnh được phát triển vào khoảng những năm 1955. Điều này có thể giải thích được vì sau thế chiến thứ hai, máy tính phát triển nhanh tạo điều kiện cho quá trình xử lý ảnh sô thuận lợi. Năm 1964, máy tính đã có khả năng xử lý và nâng cao chất lượng ảnh từ mặt trăng và vệ tinh Ranger 7 của Mỹ bao gồm: làm nổi đường biên, lưu ảnh. Từ năm 1964 đến nay, các phương tiện xử lý, nâng cao chất lượng, nhận dạng ảnh phát triển không ngừng. Các phương pháp tri thức nhân tạo như mạng nơ ron nhân tạo, các thuật toán xử lý hiện đại và cải tiến, các công cụ nén ảnh ngày càng được áp dụng rộng rãi và thu nhiều kết quả khả quan.

Để dễ tưởng tượng, xét các bước cần thiết trong xử lý ảnh. Đầu tiên, ảnh tự nhiên từ thế giới ngoài được thu nhận qua các thiết bị thu (như Camera, máy chụp ảnh). Trước đây, ảnh thu qua Camera là các ảnh tương tự (loại Camera ống kiểu CCIR). Gần đây, với sự phát triển của công nghệ, ảnh màu hoặc đen trắng được lấy ra từ Camera, sau đó nó được chuyển trực tiếp thành ảnh số tạo thuận lợi cho xử lý tiếp theo. (Máy ảnh số hiện nay là một thí dụ gần gũi). Mặt khác, ảnh cũng có thể tiếp nhận từ vệ tinh; có thể quét từ ảnh chụp bằng máy quét ảnh. Hình 1.3.1 dưới đây mô tả các bước cơ bản trong xử lý ảnh.



*Hình 1. 5 : Các bước cơ bản trong xử lý ảnh*

Sơ đồ này bao gồm các thành phần sau:

1. *Phần thu nhận ảnh (Image Acquisition)*

Ảnh có thể nhận qua camera màu hoặc đen trắng. Thường ảnh nhận qua camera là ảnh tương tự (loại camera ống chuẩn CCIR với tần số 1/25, mỗi ảnh 25 dòng), cũng có loại camera đã số hoá (như loại CCD – Change Coupled Device) là loại photodiot tạo cường độ sáng tại mỗi điểm ảnh.

Camera thường dùng là loại quét dòng ; ảnh tạo ra có dạng hai chiều. Chất lượng một ảnh thu nhận được phụ thuộc vào thiết bị thu, vào môi trường (ánh sáng, phong cảnh).

1. *Tiền xử lý (Image Processing)*

Sau bộ thu nhận, ảnh có thể nhiễu độ tương phản thấp nên cần đưa vào bộ tiền xử lý để nâng cao chất lượng. Chức năng chính của bộ tiền xử lý là lọc nhiễu, nâng độ tương phản để làm ảnh rõ hơn, nét hơn.

1. *Phân đoạn (Segmentation) hay phân vùng ảnh*

Phân vùng ảnh là tách một ảnh đầu vào thành các vùng thành phần để biểu diễn phân tích, nhận dạng ảnh. Ví dụ: để nhận dạng chữ (hoặc mã vạch) trên phong bì thư cho mục đích phân loại bưu phẩm, cần chia các câu, chữ về địa chỉ hoặc tên người thành các từ, các chữ, các số (hoặc các vạch) riêng biệt để nhận dạng. Đây là phần phức tạp khó khăn nhất trong xử lý ảnh và cũng dễ gây lỗi, làm mất độ chính xác của ảnh. Kết quả nhận dạng ảnh phụ thuộc rất nhiều vào công đoạn này.

1. *Biểu diễn ảnh (Image Representation)*

Đầu ra ảnh sau phân đoạn chứa các điểm ảnh của vùng ảnh (ảnh đã phân đoạn) cộng với mã liên kết với các vùng lận cận. Việc biến đổi các số liệu này thành dạng thích hợp là cần thiết cho xử lý tiếp theo bằng máy tính. Việc chọn các tính chất để thể hiện ảnh gọi là trích chọn đặc trưng (Feature Selection) gắn với việc tách các đặc tính của ảnh dưới dạng các thông tin định lượng hoặc làm cơ sở để phân biệt lớp đối tượng này với đối tượng khác trong phạm vi ảnh nhận được. Ví dụ: trong nhận dạng ký tự trên phong bì thư, chúng ta miêu tả các đặc trưng của từng ký tự giúp phân biệt ký tự này với ký tự khác.

1. *Nhận dạng và nội suy ảnh (Image Recognition and Interpretation)*

Nhận dạng ảnh là quá trình xác định ảnh. Quá trình này thường thu được bằng cách so sánh với mẫu chuẩn đã được học (hoặc lưu) từ trước. Nội suy là phán đoán theo ý nghĩa trên cơ sở nhận dạng. Ví dụ: một loạt chữ số và nét gạch ngang trên phong bì thư có thể được nội suy thành mã điện thoại. Có nhiều cách phân loai ảnh khác nhau về ảnh. Theo lý thuyết về nhận dạng, các mô hình toán học về ảnh được phân theo hai loại nhận dạng ảnh cơ bản: - Nhận dạng theo tham số. - Nhận dạng theo cấu trúc. Một số đối tượng nhận dạng khá phổ biến hiện nay đang được áp dụng trong khoa học và công nghệ là: nhận dạng ký tự (chữ in, chữ viết tay, chữ ký điện tử), nhận dạng văn bản (Text), nhận dạng vân tay, nhận dạng mã vạch, nhận dạng mặt người…

1. *Cơ sở tri thức (Knowledge Base)*

Như đã nói ở trên, ảnh là một đối tượng khá phức tạp về đường nét, độ sáng tối, dung lượng điểm ảnh, môi trường để thu ảnh phong phú kéo theo nhiễu. Trong nhiều khâu xử lý và phân tích ảnh ngoài việc đơn giản hóa các phương pháp toán học đảm bảo tiện lợi cho xử lý, người ta mong muốn bắt chước quy trình tiếp nhận và xử lý ảnh theo cách của con người. Trong các bước xử lý đó, nhiều khâu hiện nay đã xử lý theo các phương pháp trí tuệ con người. Vì vậy, ở đây các cơ sở tri thức được phát huy. Trong tài liệu, chương 6 về nhận dạng ảnh có nêu một vài ví dụ về cách sử dụng các cơ sở tri thức đó.

1. *Mô tả (biểu diễn ảnh)*

Từ hình 1.3.1, ảnh sau khi số hoá sẽ được lưu vào bộ nhớ, hoặc chuyển sang các khâu tiếp theo để phân tích. Nếu lưu trữ ảnh trực tiếp từ các ảnh thô, đòi hỏi dung lượng bộ nhớ cực lớn và không hiệu quả theo quan điểm ứng dụng và công nghệ. Thông thường, các ảnh thô đó được đặc tả (biểu diễn) lại (hay đơn giản là mã hoá) theo các đặc điểm của ảnh được gọi là các đặc trưng ảnh (Image Features) như: biên ảnh (Boundary), vùng ảnh (Region).

Một số phương pháp biểu diễn thường dùng:

• Biểu diễn bằng mã chạy (Run-Length Code)

• Biểu diễn bằng mã xích (Chaine -Code)

• Biểu diễn bằng mã tứ phân (Quad-Tree Code)

* Biểu diễn bằng mã chạy

Phương pháp này thường biểu diễn cho vùng ảnh và áp dụng cho ảnh nhị phân. Một vùng ảnh R có thể mã hoá đơn giản nhờ một ma trận nhị phân:

U(m, n) = 1 nếu (m, n) thuộc R

U( m, n) = 0 nếu (m, n) không thuộc R

Trong đó: U(m, n) là hàm mô tả mức xám ảnh tại tọa độ (m, n). Với cách biểu diễn trên, một vùng ảnh được mô tả bằng một tập các chuỗi số 0 hoặc 1. Giả sử chúng ta mô tả ảnh nhị phân của một vùng ảnh được thể hiện theo toạ độ (x, y) theo các chiều và đặc tả chỉ đối với giá trị “1” khi đó dạng mô tả có thể là: (x, y)r; trong đó (x, y) là toạ độ, r là số lượng các bit có giá trị “1” liên tục theo chiều ngang hoặc dọc.

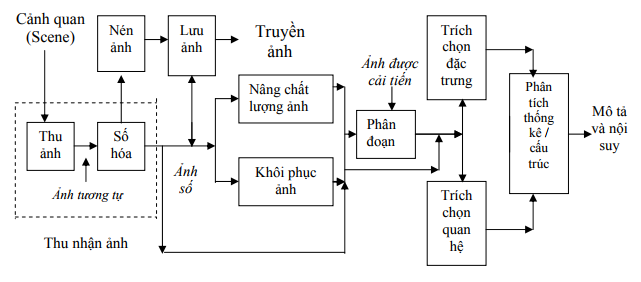
* Biểu diễn bằng mã xích

Phương pháp này thường dùng để biểu diễn đường biên ảnh. Một đường bất kỳ được chia thành các đoạn nhỏ. Nối các điểm chia, ta có các đoạn thẳng kế tiếp được gán hướng cho đoạn thẳng đó tạo thành một dây xích gồm các đoạn. Các hướng có thể chọn 4, 8, 12, 24,… mỗi hướng được mã hoá theo số thập phân hoặc số nhị phân thành mã của hướng.

* Biểu diễn bằng mã tứ phân

Phương pháp mã tứ phân được dùng để mã hoá cho vùng ảnh. Vùng ảnh đầu tiên được chia làm bốn phần thường là bằng nhau. Nếu mỗi vùng đã đồng nhất (chứa toàn điểm đen (1) hay trắng (0)), thì gán cho vùng đó một mã và không chia tiếp. Các vùng không đồng nhất được chia tiếp làm bốn phần theo thủ tục trên cho đến khi tất cả các vùng đều đồng nhất. Các mã phân chia thành các vùng con tạo thành một cây phân chia các vùng đồng nhất.

Trên đây là các thành phần cơ bản trong các khâu xử lý ảnh. Trong thực tế, các quá trình sử dụng ảnh số không nhất thiết phải qua hết các khâu đó tùy theo đặc điểm ứng dụng. Ảnh sau khi được số hóa được nén, luuw lai để truyền cho các hệ thống khác sử dụng hoặc để xử lý tiếp theo. Mặt khác, ảnh sau khi số hóa có thể bỏ qua công đoạn nâng cao chất lượng (khi ảnh đủ chất lượng theo một yêu cầu nào đó) để chuyển tới khâu phân đoạn hoặc bỏ tiếp khâu phân đoạn chuyển trực tiếp tới khâu trích chọn đặc trưng. Hình 1.3.3 cũng chia các nhánh song song như: nâng cao chất lượng ảnh có hai nhánh phân biệt: nâng cao chất lượng ảnh (tăng độ sáng, độ tương phản, lọc nhiễu) hoặc khôi phục ảnh (hồi phục lại ảnh thật khi ảnh nhận được bị méo) v.v…



*Hình 1. 6 : Sơ đồ phân tích và xử lý ảnh và lưu đồ thông tin giữa các khối*

## 1.4 Khái niệm cơ bản về xử lý ảnh

### *1.4.1 Phần tử ảnh*

Pixel (Picture Element – phần tử ảnh): ảnh trong thực tế là một ảnh liên tục về không gian và về giá trị độ sáng. Để có thể xử lý ảnh bằng máy tính cần thiết phải tiến hành số hóa ảnh. Trong quá trình số hóa, người ta biến đối tín hiệu liên tục sang tín hiệu rời rạc thông qua quá trình lấy mẫu ( rời rạc hóa về không gian) và lượng hóa thành phần giá trị mà về nguyên tắc bằng mắt thường không phân biệt được hai điểm kề nhau. Trong quá trình này, người sử dụng khái niệm pixel – phần tử ảnh ở đây cũng cần phải phân biệt rõ khái niệm pixel hay đề cập đến trong các hệ thống đồ họa máy tính. Để tránh nhầm lẫn ta tạm gọi khái niệm này là pixel thiết bị có thể xem xét như sau: khi ta quan sát màn hình (trong chế độ đồ họa), màn hình không liên tục mà gồm nhiều điểm nhỏ, gọi là pixel. Mỗi pixel gồm một cặp tọa độ x, y và màu.

Cặp tọa độ x, y tạo nên độ phân giải (resolution) như màn hình máy tính có nhiều loại với độ phân giải khác nhau: màn hình CGA có độ phân giải là 320x 200, màn hình VGA là 640x350 …

Như vậy, một ảnh là một tập hợp các điểm ảnh. Khi được số hóa, nó thường được biểu diễn bởi bảng hai chiều I (n, p): n dòng và p cột. Ta nói ảnh gồm n x p pixels. Người ta thường ký hiệu I(x, y) để chỉ một pixel. Thường giá trị của n chọn bằng p và bằng 256. Một pixel có thể lưu trữ trên 1, 4, 8 hay 24 bít.

### *1.4.2 Ảnh*

Tín hiệu ảnh thuộc loại tín hiệu đa chiều: tọa độ(x, y, z), độ sáng (λ), thời gian(t). Ảnh tĩnh trong không gian 2 chiều được định nghĩa là một hàm 2 biến S(x, y), với S là giá trị biên độ (được biểu diễn bằng màu sắc) tại vị trí không gian (x, y).

Phân loại ảnh:

* Ảnh tương tự S(x, y): (x, y) liên tục, S liên tục.
* Ảnh số S(m,n): (m,n) rời rạc, S rời rạc.

Một ảnh (gồm một tập các điểm ảnh) có thể xem như bao gồm tập các ảnh con (các vùng ảnh). Thuật ngữ gọi là ROIs – vùng quan tâm.

Ảnh số trong không gian rời rạc thu được từ ảnh tương tự trong không gian liên tục thông qua quá trình số hóa. Quá trình số hóa có thể hiểu đơn giản như sau:

* Ảnh tương tự được chia thành M hàng, N cột.
* Giao của hàng và cột được gọi là: pixel
* Giá trị biên độ của pixel tại tọa độ nguyên (m, n) là s (m, n): là trung bình độ sáng trong pixel đó. S (m, n) ≤ L (L số mức xám dùng biểu diễn ảnh). M, N thường được chọn là M=N=2K (K=8, 9, 10). L =2B, B là số bít mã hóa cho độ sáng (biên độ) mỗi pixel.
* Ảnh số được biểu diễn bởi ma trận 2 chiều. Các phần tử của nó là biểu diễn cho các pixel số hóa.
* Ta ký hiệu 1 ảnh số là S (M, N). Ta nói ảnh có độ phân giải MxN. Ký hiệu S (m, n) để chỉ ra một phần tử ảnh.

### *1.4.3 Mức xám*

Mức xám là kết quả của sự mã hóa tương ứng với một cường độ sáng của mỗi điểm ảnh với một giá trị số, kết quả của quá trình lượng tử hóa. Cách mã hóa kinh điển là dùng 16, 32 hay 64 mức. Mã hóa 256 mức được dùng phổ biến. Do 28 = 256 nên mỗi pixel được mã hóa bởi 8 bit.

### *1.4.4 Ảnh đen trắng*

Thực tế ảnh đen trắng gồm có ảnh nhị phân và ảnh đa cấp xám. Chúng ta phân biệt sự biến đổi thành L mức. Nếu L bằng 2, nghĩa là chỉ có hai mức 0 và 1 còn gọi là ảnh nhị phân. Nếu L lớn hơn hai mức ta có ảnh đa cấp xám. Việc xác định số mức phụ thuộc vào tiêu chí lượng tử hóa. L thường được chọn bằng 32, 64, 128 và 256. Trong đó ảnh đa cấp xám 256 mức là ảnh có chất lượng cao và thường được sử dụng.

Với ảnh nhị phân, mỗi pixel được mã hóa trên một bit. Còn với ảnh xám 256 mức được mã hóa trên 1 byte = 8 bit. Ảnh nhị phân khá đơn giản, các phần tử ảnh có thể coi như phần tử logic. Ảnh nhị phân được dùng theo tính logic, để phân biệt ảnh với nền hay đơn giản để phân biệt biên với điểm khác.

### *1.4.5 Ảnh màu*

Màu sắc của một tia sáng là cảm giác mà tia sáng đó gây nên ở mắt người. Màu sắc của vật thể là màu của ánh sáng phát từ chúng. Ánh sáng này nằm trong dải phổ nhìn thấy (visible).

Tùy theo ngữ cảnh nhất định mà ảnh màu được diễn tả theo những mô hình màu khác nhau. Không có mô hình màu nào là đầy đủ cho mọi khía cạnh của màu. Do đó người ta sử dụng những mô hình màu khác nhau để mô tả các tính chất được nhận biết

## 1.5 Các vấn đề cơ bản trong xử lý ảnh

* Biểu diễn ảnh:

Ảnh có thể xem là một hàm 2 biến chứa các thông tin như biểu diễn của một ảnh. Các mô hình biểu diễn ảnh cho ta một mô tả logic hay định lượng của hàm này. Dựa vào phần tử đặc trung của ảnh đó là pixel. Giá trị pixel có thể là một giá trị vô hướng, hoặc là 1 vector (3 thành phần trong trường hợp ảnh màu).

Ta có thể biểu diễn ảnh bằng hàm toán học, hoặc các ma trận điểm.

* Mô hình hóa ảnh**:**

Mô hình cảm nhận ảnh: Là mô hình biểu diễn thông qua các thuộc tính cảm nhận ảnh (màu sắc, cường độ sáng), các thuộc tính về thời gian, các cảm nhận về phối cảnh, bố cục.

Mô hình cục bộ: Là mô hình biểu diễn thể hiện mối tương quan cục bộ của các phần tử ảnh (ứng dụng cho các bài toán xử lý và nâng cao chất lượng ảnh).

Mô hình tổng thể: Là mô hình biểu diễn ành xem ảnh như là một tập họp các đối tượng, và các đối tượng này có mối quan hệ không gian vói nhau (ứng dụng cho các bài toán phân nhóm và nhận dạng ảnh).

* Khôi phục ảnh

Mục đích: Khôi phục lại ảnh ban đầu, loại bỏ các biến dạng ra khỏi ảnh tùy theo nguyên nhân gây ra biến dạng.

* Biến đổi ảnh

Mục đích: Biến đổi the hiện cùa ảnh dưới các góc nhìn khác nhau tiến cho việc xử lý, phân tích ảnh.

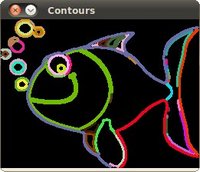
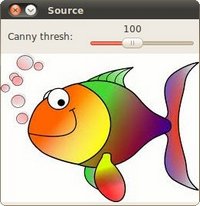
Các phương pháp: Biển đổi Fourier, Sin, Cosin, KL, ...

* Phân tích ảnh

Mục đích: Tìm ra các đặc trưng của ảnh, xây dụng quan hệ giữa chúng dựa vào các đặc trưng cục bộ.

Các thao tác: Tìm biên, tách biên, làm mảnh đường biên, phân vùng ảnh, phân loại đối tượng.

Các phương pháp: Phương pháp phát hiện biên cục bộ, dò biên theo qui hoạch động, phân vùng theo miền đồng nhất, phân vùng dựa theo đường biên...



*Hình 1. 7 : Thuật toán tìm đường biên*

* Nén ảnh

Mục đích: giảm không gian lưu trữ, thuận tiện truyền thông trên mạng.

Phương pháp: nén không mất thông tin, nén mất thông tin

Nén không mất thông tin (nén chính xác): khai thác các thông tin dư thừa.

Nén mất thông tin: khai thác các thông tin dư thừa và các thông tin không liên quan.

Hiện nay có một số chuẩn nén hay dùng; JPEG, MPEG (JPEG-2000, MPEG-4)

* Nhận dạng ảnh

Nhận dạng ảnh là quá trình liên quan đến mô tả đối tượng mà người ta muổn đặc tả nó. Thường đi sau quá trình trích chọn các đặc tính của đối tượng.

Có 2 kiểu mô tả đối tượng:

+ Mô tả theo tham số (nhận dạng theo tham số).

+ Mô tả theo cấu trúc (nhận dạng theo cấu trúc).

Ứng dụng: nhận dạng đối tượng, mặt, vân tay, văn bản...

Nhận dạng chữ in hoặc đánh máy phục vụ cho việc tự động hóa việc đọc tài liệu, tăng nhanh tốc độ và chất lượng thu nhận thông tin từ máy tính.

Nhận dạng chữ viết tay (với một so ràng buộc)

Mạng nơron là một kỹ thuật mới đang được áp dụng vào nhận dạng và cho kết quả khả quan.



*Hình 1. 8 : Nhận dạng màu sắc*

## 1.6 Không gian màu

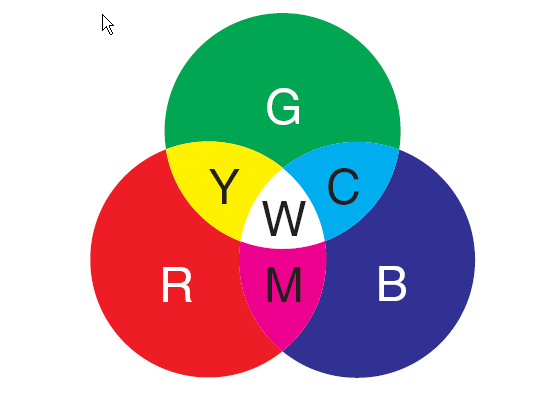
Không gian màu là một mô hình toán học dùng để mô tả các màu sắc trong thực tế được biểu diễn dưới dạng số học. Trên thực tế có rất nhiều không gian màu khác nhau được mô hình để sử dụng vào những mục đích khác nhau.

Ba không gian màu cơ bản hay được nhắc tới và ứng dụng nhiều, đó là hệ không gian màu RGB, HSV và CMYK*.*

### *1.6.1 Không gian màu RGB*

RGB là không gian màu rất phổ biến được dùng trong đồ họa máy tính và nhiều thiết bị kĩ thuật số khác. Ý tưởng chính của không gian màu này là sự kết hợp của 3 màu sắc cơ bản : màu đỏ (R, Red), xanh lục (G, Green) và xanh lơ (B, Blue) để mô tả tất cả các màu sắc khác.

Nếu như một ảnh số được mã hóa bằng 24bit, nghĩa là 8bit cho kênh R, 8bit cho kênh G, 8bit cho kênh B, thì mỗ kênh này màu này sẽ nhận giá trị từ 0-255. Với mỗi giá trị khác nhau của các kênh màu kết hợp với nhau ta sẽ được một màu khác nhau, như vậy ta sẽ có tổng cộng 255x255x255 = 1.66 triệu màu sắc. Ví dụ: màu đen là sự kết hợp của các kênh màu (R, G, B) với giá trị tương ứng (0, 0, 0) màu trắng có giá trị (255, 255, 255), màu vàng có giá trị (255, 255, 0), màu tím đậm có giá trị (64, 0, 128) ...Nếu ta dùng 16bit để mã hóa một kênh màu (48bit cho toàn bộ 3 kênh màu) thì dãi màu sẽ trãi rộng lên tới 3\*2^16 = ... Một con số rất lớn.



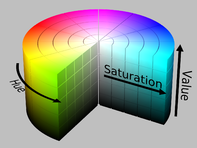
*Hình 1. 9 : Không gian màu RGB*

### *1.6.2. Không gian màu HSV*

HSV và cũng gần tương tự như HSL là không gian màu được dùng nhiều trong việc chỉnh sửa ảnh, phân tích ảnh và một phần của lĩnh vực thị giác máy tính. Hệ không gian này dựa vào 3 thông số sau để mô tả màu sắc

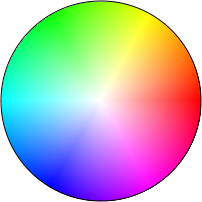
H = Hue: màu sắc, S = Saturation: độ đậm đặc, sự bảo hòa, V = value: giá trị cường độ sáng.

Không gian màu này thường được biểu diễn dưới dạng hình trụ hoặc hình nón.



*Hình 1. 10 : Không gian màu HSV*

Theo đó, đi theo vòng tròn từ 0 -360 độ là trường biểu diễn màu sắc(Hue). Trường này bắt đầu từ màu đỏ đầu tiên (red primary) tới màu xanh lục đầu tiên (green primary) nằm trong khoảng 0-120 độ, từ 120 - 240 độ là màu xanh lục tới xanh lơ (green primary - blue primary). Từ 240 - 360 là từ màu đen tới lại màu đỏ.



*Hình 1. 11 : Hình tròn biểu diễn màu sắc (Hue)*

Theo như cách biểu diễn không gian màu theo hình trụ như trên, đi từ giá trị độ sáng (V) được biểu diễn bằng cách đi từ dưới đáy hình trụ lên và nằm trong khoảng từ 0 -1. Ở đáy hình trụ V có giá trị là 0, là tối nhất và trên đỉnh hình trụ là độ sáng lớn nhất (V = 1). Đi từ tâm hình trụ ra mặt trụ là giá trị bão hòa của màu sắc (S). S có giá trị từ 0 - 1. 0 ứng với tâm hình trụ là chỗ mà màu sắc là nhạt nhất. S = 1 ở ngoài mặt trụ, là nơi mà giá trị màu sắc là đậm đặc nhất.

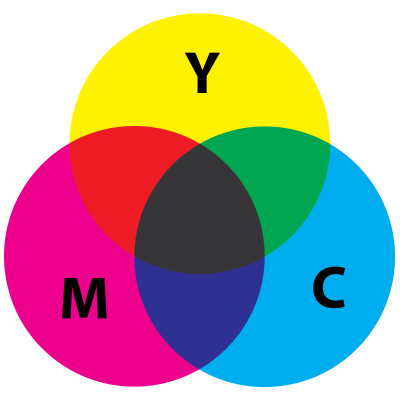
Như vậy với mỗi giá trị (H, S, V) sẽ cho ta một màu sắc mà ở đó mô tả đầy đủ thông tin về màu sắc, độ đậm đặc và độ sáng của màu đó.

Chính vì vậy trong quá trình nhận dạng màu sắc , ta nên chọn không gian màu HSV.

### *1.6.3 Không gian màu CMYK*

CMYK là không gian màu được sử dụng phổ biến trong ngành công nghiệp in ấn.Ý tưởng cơ bản của hệ không gian này là dùng 4 màu sắc cơ bản để phục vụ cho việc pha trộn mực in. Trên thực tế, người ta dùng 3 màu là C=Cyan: xanh lơ, M=Magenta: hồng xẫm, và Y=Yellow: vàng để biểu diễn các màu sắc khác nhau. Nếu lấy màu hồng xẫm cộng với vàng sẽ ra màu đỏ, màu xẫm kết hợp với xanh lơ sẽ cho xanh lam ... Sự kết hợp của 3 màu trên sẽ cho ra màu đen, tuy nhiên màu đen ở đây khôn phải là đen tuyệt đối và thường có độ tương phản lớn, nên trong ngành in, để tiết kiệm mực in người ta thêm vào màu đen để in những chi tiết có màu đen thay vì phải kết hợp 3 màu sắc trên. Và như vậy ta có hệ màu CMYK. chữ K ở đây là để kí hiệu màu đen (Black), có nhẽ chữ B đã được dùng để biểu diễn màu Blue nên người ta lấy chữ cái cuối K để biểu diễn màu đen?

Nguyên lý làm việc của hệ màu này như sau : Trên một nền giấy trắng, khi mỗi màu này được in lên sẽ loại bỏ dần đi thành phần màu trắng. 3 màu C, M, Y khác nhau in theo những tỉ lệ khác nhau sẽ loại bỏ đi thành phần đó một cách khác nhau và cuối cùng cho ta màu sắc cần in. Khi cần in màu đen, thay vì phải in cả 3 màu người ta dùng màu đen để in lên. Nguyên lý này khác với nguyên lý làm việc của hệ RGB ở chỗ hệ RGB là sự kết hợp của các thành phần màu, còn hệ CMYK là sự loại bỏ lẫn nhau của các thành phần màu.

**

*Hình 1. 12 : Không gian màu CMYK*

### *1.6.4 Chuyển đổi giữa các không gian màu*

Chuyển đổi RGB sang CMYK và ngược lại.

Như đã nói ở trên, thành phần K là thành phần phụ dùng để in cho những điểm màu có màu đen trong hệ CYMK, do vậy để chuyển không gian màu từ RGB sang CMYK trước hết ta chuyển RGB sang CMY sau đó tìm thành phần K còn lại. Công thức chuyển từ RGB sang CMY:

(C', M', Y') = ((255 - R), (255 - G), (255 - B)).

Việc tính giá trị của K lại là một vấn đề khác vì nó liên quan tới nhà sản xuất công nghệ in, tuy nhiên về mặt lý thuyết có thể chấp nhận rằng K = min {C'/2,55, M'/2,55, Y'/2,55} , như vậy 0<= K <=100.

Nếu K = 100, thì C = M = Y =0 (trương hợp in màu đen)

Nếu 0< K < 100: C = (C'/2.55 - K) \* 100 /(100 - K), M = (M'/2.55 - K) \* 100 /(100 - K), Y = (Y'/2.55 - K) \*100 /(100 - K) và K = K. Trong đó, C, M, Y, K được làm tròn tới để lấy chỉ số nguyên.

Chuyển đổi RGB sang HSV và ngược lại

Giả sử ta có một điểm màu có giá trị trong hệ RGB là (R, G, B). ta chuyển sang không gian HSV như sau:

Đặt M = Max(R, G, B), m = Min(R, G, B) và C = M - m.

Nếu M = R, H' = (G - B)/C mod 6. Nếu M = G, H' = (B - R)/C + 2. Nếu M = B, H' = (R - G)/C + 4. Và H = H'x60. Trong trường hợp C = 0, H = 0 độ.

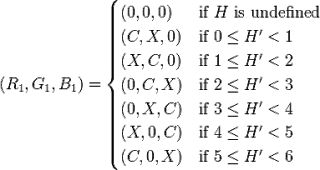
V = M.

S = C/V. Trong trường hợp V hoặc C bằng 0, S = 0.

Để chuyển từ HSV sang RGB ta làm như sau:

Giả sử ta có không gian màu HSV với H = [0, 360], S = [0, 1], V = [0, 1]. Khi đó, ta tính C = VxS. H' = H/60.

X = C(1 - |H' mod2 -1|). Ta biểu diễn hệ (R1, G1, B1) như sau:



Đặt m = V - C và ta có kết quả cuối cùng: (R, G, B) = (R1 + m, G1 + m, G1 +m).

## 1.7 Một số kỹ thuật tiền xử lý ảnh

### *1.7.1 Chuyển ảnh xám*

Ảnh là tập hợp từ rất nhiều điểm ảnh. Với ảnh 24 bít của hệ màu RGB do CIE đề xuất thì mỗi thành phần màu cấu tạo nên điểm ảnh có giá trị thể hiện bởi 8 bít (tức là có 256 giá trị). Mô hình màu RGB sử dụng mô hình bổ sung trong đó ánh sáng đỏ, xanh lá cây và xanh lam được tổ hợp với nhau theo nhiều phương thức khác nhau để tạo thành các màu khác nhau. Đối với ảnh xám, thông thường mỗi pixel mang thông tin của 256 mức xám (tương ứng với tám bít) như vậy ảnh xám hoàn toàn có thể tái hiện đầy đủ cấu trúc của một ảnh màu tương ứng thông qua tám mặt phẳng bít theo độ xám.

Bước chuyển từ ảnh màu thành ảnh xám là một công đoạn phổ biến trong các quá trình xử lý ảnh vì nó làm tăng tốc độ xử lý, giảm mức độ phức tạp của các thuật toán trên ảnh. Để chuyển về ảnh đa cấp xám, ta cần làm cho 3 thành phần R,G,B bằng nhau theo cách sau:

G=αR+βG+γB với điều kiện α+β+γ=1

Trong đó các giá trị R,G,B lần lượt là các mức độ màu Đỏ, Xanh lá và Xanh biển của pixel màu.

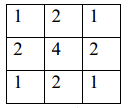
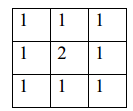
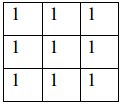
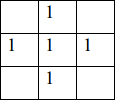
### *1.7.2 Lọc nhiễu*

Trong nhiều lĩnh vực kỹ thuật, nhiễu đóng vai trò chủ yếu gây nên những khó khăn khi ta cần phân tích một tín hiệu nào đó. Giữa một ảnh thực và ảnh số hoá thu nhận được khác nhau khá nhiều vì có nhiều quá trình can thiệp vào. Nguyên nhân là do nhiễu điện tử của máy thu hay chất lượng kém của bộ số hoá. Giả sử ảnh là một miền có mức xám đồng nhất, như vậy các phần tử của ma trận biểu diễn ảnh sau quá trình số hoá phải có cùng giá trị. Nhưng thực tế quan sát ta thấy: gần giá trị trung bình của mức xám có những phần tử trội lên khá nhiều. Đó chính là hiện tượng nhiễu. Như vậy nhiễu trong ảnh số được xem như sự dịch chuyển nhanh của tín hiệu thu nhận trên một khoảng cách ngắn. Xem xét một cách tương đương trong không gian tần số, nhiễu ứng với các thành phần tần số cao trong ảnh. Do vậy, người ta nghĩ đến việc biến đổi có tính đến ảnh hưởng của các phần tử lân cận bằng cách lấy “tổ hợp” các điểm lân cận này (trong không gian thực) hay lọc các thành phần tần số cao (trong không gian tần số). Đây chính là kỹ thuật lọc.

Trong kỹ thuật này, ta sử dụng một mặt nạ di chuyển khắp ảnh gốc. Tuỳ theo cách tổ hợp điểm đang xét với các điểm lân cận mà có kỹ thuật lọc tuyến tính hay phi tuyến. Điểm ảnh chịu tác động là điểm ở tâm mặt nạ.

* Bộ lọc tuyến tính

Trong kỹ thuật lọc tuyến tính, ảnh thu được sẽ là tổng trọng số hay là trung bình trọng số các điểm lân cận với nhân cuộn hay mặt nạ.Vì có nhiều loại nhiễu can thiệp vào quá trình xử lý ảnh nên cần có nhiều bộ lọc thích hợp. Để làm trơn nhiễu ta sử dụng các mặt nạ thông thấp.



*Hình 1. 13 : Mặt nạ thông thấp*

* Lọc trung bình

Là kĩ thuật lọc tuyến tính, hoạt động như một bộ lọc thông thấp. Ý tưởng chính của thuật toán là: ta sử dụng một cửa sổ lọc (ma trận 3x3) quét qua lần lượt từng điểm ảnh của ảnh đầu vào input. Tại vị trí mỗi điểm ảnh lấy giá trị của các điểm ảnh tương ứng trong vùng 3x3 của ảnh gốc “lấp” vào ma trận lọc. Giá trị điểm ảnh của ảnh đầu ra là giá trị trung bình của tất cả các điểm ảnh trong cửa sổ lọc. Việc tính toán này khá đơn giản với hai bước gồm tính tổng các thành phần trong cửa sổ lọc và sau đó chia tổng này cho số các phần tử của cửa sổ lọc.

Sơ lược một cách ngắn gọn các bước của giải thuật:

* Quét cửa sổ lọc lần lượt lên các thành phần của ảnh đầu vào; điền các giá trị được quét vào cửa sổ lọc.
* Xử lý bằng cách thao tác trên các thành phần của cửa sổ lọc.
* Tính giá trị trung bình các thành phần trong cửa sổ lọc.
* Gán giá trị trung bình này cho ảnh đầu ra.

Nhận xét: Đây là mạch lọc tuyến tính đơn giản trong tính toán. Ảnh đạt độ trơn mịn. Tuy vậy, nó không hoàn toàn loại bỏ tác dụng của nhiễu, các pixel đơn lẻ (nhiễu) sẽ ảnh hưởng đến giá trị trung bình của các pixel lân cận. Nó cũng làm cho độ sắc nét của ảnh kém cũng như độ tương phản thấp.

Lọc tuyến tính ngoài làm trơn nhiễu còn có thể trích chọn biên (nổi biên). Dễ dàng nhận thấy rằng biên là điểm có độ biến thiên nhanh về giá trị mức xám. Theo quan điểm về tần số tín hiệu, các điểm biên ứng với các thành phần tần số cao. Do vậy, ta có thể dùng bộ lọc thông cao để cải thiện: lọc các thành phần tần số thấp và chỉ giữ lại thành phần tần số cao. Vì thế, lọc thông cao thường được dùng làm trơn biên trước khi tiến hành các thao tác với biên ảnh. Dưới đây là một số mặt nạ dùng trong lọc thông cao:



### *1.7.3 Nhị phân ảnh*

Phân ngưỡng hay còn gọi là nhị phân hóa. Mục đích của phân ngưỡng là chuyển từ ảnh mầu, ảnh đa cấp xám sang ảnh nhị phân (ảnh 2 cấp xám, ảnh đen trắng). Nếu phân ngưỡng hợp lý thì có thể phân biệt được vật thể với nền. Từ đó ảnh hưởng lớn đến các thao tác lựa chọn vật thể đúng hay lẫn cả vào với nền.

Giả sử đã chọn được một ngưỡng L thích hợp. Ta thực hiện việc sau:

* Nếu điểm ảnh có mức xám cao hơn ngưỡng L thì đặt cho nó giá trị 255
* Nếu ngược lại thì đặt giá trị 0.
* Từ đó ta thấy rõ ánh xạ của ảnh mới chính là 1 ánh xạ sang ảnh nhị phân có 2 giá trị 0 và 1 (1 tương ứng với 255).

Có nhiều phương pháp để xác định giá trị ngưỡng. Một phương pháp là thiết lập ngưỡng sao cho số lượng các điểm đen đạt một ngưỡng chấp nhận được theo phân phối xác suất mức xám. Ví dụ, chúng ta có thể biết rằng các ký tự chiếm 25% diện tích của một trang văn bản thông thường. Vì thế chúng ta có thể thiết lập ngưỡng sao cho số lượng điểm đen còn lại chiếm ¼ trang văn bản. Một cách tiếp cận khác là chọn ngưỡng nằm ở vị trị thấp nhất trên biểu đồ histogram giữa hai đỉnh của nó. Tuy nhiên việc xác định vị trí này thường rất khó khăn do hình dạng của histogram thường lởm chởm. Một giải pháp để giải quyết vấn đề này là xấp xỉ giá trị của histogram giữa hai đỉnh với một hàm giải tích và sử dụng vi phân để xác định điểm thấp nhất. Ví dụ, coi x và y lần lượt là hoành độ và tung độ trên histogram. Chúng ta có thể sử dụng hàm: y=ax2 + bx + c. Với a, b, c là hằng số làm hàm xấp xỉ đơn giản cho histogram ở vị trí giữa hai đỉnh của nó. Vị trí thấp nhất sẽ có tọa độ x=-b/2a.

Phương pháp xấp xỉ các giá trị của histogram và tìm vị trí thấp nhất cho giá trị ngưỡng tốt hơn nhưng lại yêu cầu tài nguyên về tính toán để thực hiện độ phức tạp trong việc cài đặt. Phương pháp tìm ngưỡng theo phân phối xác suất đơn giản hơn và kết quả của nó tương đối đáp ứng được các yêu cầu cho việc nhận dạng.

### *1.7.4 Lược đồ mức xám (Histogram)*

Lược đồ mức xám là một hàm cung cấp tần suất xuất hiện của mỗi mức xám (gray-level) trong ảnh. Biễu diễn toán học của histogram của một ảnh số có L=256 mức xám là một hàm rời rạc :



trong đó :

 là giá trị xám thứ k (k = 0, 1,..., L-1)

 là số pixel có mức xám đó và

 là tổng số pixel của ảnh.

Miền giá trị của.

Một cách biễu diễn toán học khác của histogram của một ảnh là số lần xuất hiện của mỗi mức xám:



Khi lược đồ xám được biễu diễn trong một hệ tọa độ vuông góc x, y (trục hoành x biễu diễn số mức xám từ 0 đến L -1, trục tung y biễu diễn số điểm ảnh có cùng mức xám hay tỷ lệ số điểm ảnh có cùng mức xám trên tổng số điểm ảnh), thì hình dạng của histogram của ảnh sẽ mang đến cho chúng ta thông tin về tính động của ảnh (ảnh rất sáng hay ảnh rất đậm) dùng làm cơ sở cho việc tăng cường độ tương phản.

. 

Ảnh có độ tương phản thấp

Ảnh có độ tương phản cao

Ảnh tối

Ảnh sáng

*Hình 1. 14 : Lược đồ mức xám của các loại ảnh*

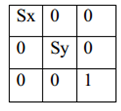
Nhìn và biểu đồ trên ta nhận thấy rằng: nếu ảnh tối thì mức xám sẽ tập trung ở gần gốc tọa độ.

### *1.7.5 Phóng to, thu nhỏ ảnh*

Còn được gọi là sự chia lại thang tọa độ. Khi chia thang tọa độ gốc và hướng được giữ nguyên, chia thang tiến hành với hệ số chia thang Sx, Sy và các tọa độ mới được tính toán theo công thức:

x’ = x\*Sx y’ = y\*Sy

Phép biến đổi có thể được mô tả dưới định thức sau:



T=

### *1.7.6. Kỹ thuật phân vùng đối tượng*

Phân vùng ảnh là bước quan trọng của xử lý ảnh, giai đoạn này nhằm phân tích ảnh thành những thành phần có cùng tính chất nào đó dựa theo biên hay các vùng liên thông.

Vùng ảnh là một chi tiết, thực thể trong toàn cảnh, vùng ảnh nói lên tính chất bề mặt của ảnh, vùn ảnh được bao quanh bởi một biên, các điểm trong vùng ảnh có độ biến thiên giá trị mức xám tương đối đồng đều. Dựa vào đặc tính vật lý của vùng ảnh mà ta xác định nhiều kỹ thuật phân vùng. Một số phương pháp phân vùng chính như:

* Dựa trên sự phân lớp hay lấy ngưỡng.
* Dựa trên cấu trúc.
* Dựa trên đường biên.
* Phân vùng ảnh dựa trên phương pháp phân lớp

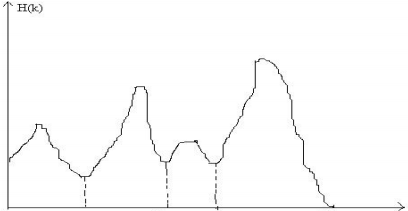
Kỹ thuật này thích hợp với ảnh nhị phân như văn bản, ảnh X-quang.

Giả sử ta có T0=Tmin, T1, T2…Tn ngưỡng giá trị xám thì ảnh sẽ được chia làm (N-1) vùng như sau: điểm ảnh I(m,n) thuộc vùng Cinếu Ti-1<= I(m,n) < Ti.Do đó việc chọn ngưỡng ở đây là bước rất quan trọng. Kỹ thuật lấy ngưỡng không nhất thiết phải được áp dụng cho toàn bộ ảnh, mà có thể áp dụng cho từng vùng ảnh một. Các giá trị ngưỡng được tính riêng biệt cho từng vùng một và sau đó được kết hợp lại thông qua phép nội suy để hình thành nên một mặt ngưỡng cho toàn bộ ảnh. Trong thuật toán mới này, kích thước của các vùng cần được chọn một cách thích hợp sao cho có một lượng đáng kể các điểm ảnh ở trong một vùng, nhằm phục vụ cho việc tính lược đồ và xác định

\*Xác định ngưỡng dựa trên biểu đồ Histogram

Các bước chọn ngưỡng như sau:

* Xem xét lược đồ xám của ảnh để xác định các ngưỡng và các khe, nếu ảnh có dạng hình rắn lượn (nhiều đỉnh và khe) thì các khe cỏ thể chọn làm ngưỡng.
* Chọn ngưỡng t sao cho có một phần xác định trước của mẫu là nhỏ hơn t
* Điều chỉnh ngưỡng dựa trên lược đồ xám của các điểm lân cận



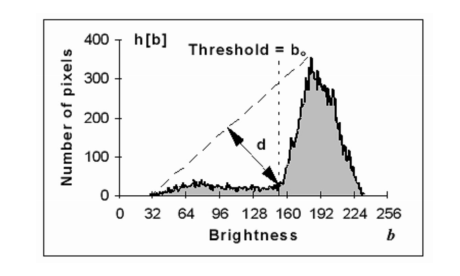
T0 T1 T2 T3 T4 L( mức xám)

*Hình 1. 15 : Lược đồ xám của ảnh*

Giả sử ảnh có lược đồ xám như hình trên ta chọn các ngưỡng T0 = Lmin, T4 = Lmax. Ta có 5 ngưỡng và tiến hành chia làm phân vùng. Sau khi phân xong, nếu ảnh rõ nét thì việc phân vùng coi như kết thúc. Trường hợp ngược lại cần điều chỉnh ngưỡng.

* Xác định ngưỡng dựa trên thuật toán tam giác

Thuật toán này do Zack đưa ra và được minh họa trong hình dưới.



*Hình 1. 16 : Thuật toán tam giác*

Trong hình này, chúng ta có thể quan sát thấy một đường thẳng đã được xây dựng bằng cách nối từ giá trị lớn nhất của lược đồ tại độ sáng đến giá trị nhỏ nhất của lược đồ sáng bmin. Với mỗi độ sáng b trong khoảng [bmax, bmin], chúng ta đi tính khoảng cách d từ giá trị lược đồ tại b là h[b] đến đường thẳng đã có. Giá trị bo ứng với khoảng cách lớn nhất sẽ được chọn làm giá trị ngưỡng ϴ.[3].

* Xác định ngưỡng dựa trên thuật toán ISODATA

Kỹ thuật chọn ngưỡng theo kiểu lặp này do Ridler và Calvard đưa ra.

Thuật toán được mô tả như sau:

- Bước 1: chia lược đồ thành 2 đoạn bằng một giá trị ngưỡng khởi động

ϴ= (Lmin + Lmax)/2

Lmin, Lmax: Giá trị mức xám nhỏ nhất, lớn nhất của vùng ảnh.

- Bước 2: tính giá trị xám trung bình của hai vùng được phân chia theo

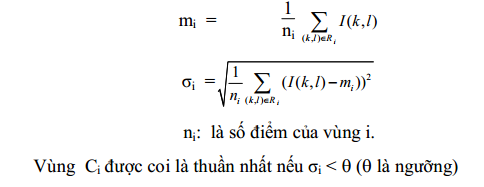
ngưỡng ϴ ta được L1 và L2. Xét giá trị ngưỡng mới:

ϴ1= (L1 + L2)/2

Quá trình này cứ thế sẽ được tiếp tục với các ngưỡng mới cho đến khi nào giá trị ngưỡng không thay đổi nữa thì dừng lại.

* Phân vùng ảnh dựa trên cấu trúc

Trên thực tế người ta hay sử dụng trung bình số học mi và độ lệch chuẩn I cho vùng R có ni điểm:



* Tách vùng theo cây tứ phân phương pháp này ta sử dụng một cây tứ phân để đánh dấu các vùng, mỗi nút trong cây được gán nhãn duy nhất, quá trình tách vùng diễn ra như sau:

+ Gốc của cây tương ứng với ảnh ban đầu.

+ Nếu một nút của cây chưa là vùng đồng nhất thì chia tiếp làm 4 vùng con, quá trình chia chỉ dừng lại khi tất cả các nút lá đều là vùng đồng nhất hoặc không thể chia được nữa.

+ Giá trị xám tại mỗi nút là giá trị xám trung bình của vùng.

* Hợp vùng

Ý tưởng của phương pháp là xét ảnh từ các miền nhỏ nhất rồi hợp chúng lại nếu thỏa mãn tiêu chuẩn để được một miền đồng nhất lớn hơn. Tiếp tục với các miền thu được cho đến khi không thể hợp thêm được nữa. Số miền còn lại cho ta kết quả phân đoạn. Như vậy, miền nhỏ nhất của bước xuất phát là điểm ảnh. Phương pháp này hoàn toàn ngược với phương pháp tách. Song điều quan trọng ở đây là nguyên lý nối 2 vùng. Việc nối 2 vùng được thực hiện theo nguyên tắc sau :

- Hai vùng phải đáp ứng tiêu chuẩn, ví dụ như cùng mức xám.

- Hai vùng phải kề cận nhau

Dựa theo nguyên lý của phương pháp hợp, ta có 2 thuật toán :

- Thuật toán tô màu (Blob Coloring) : sử dụng khái niệm 4 liên thông,

dùng một cửa sổ di chuyển trên ảnh để so sánh với tiêu chuẩn nối.

- Thuật toán đệ quy cục bộ: sử dụng phương pháp tìm kiếm trong một cây để làmtăng kích thướcvùng.

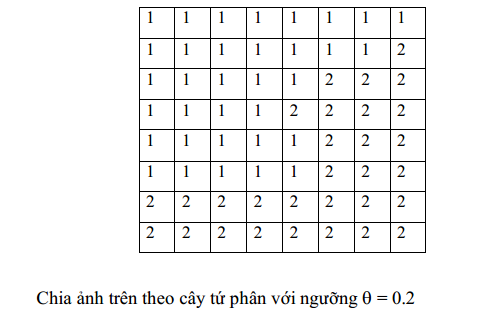
* Phương pháp tổng hợp

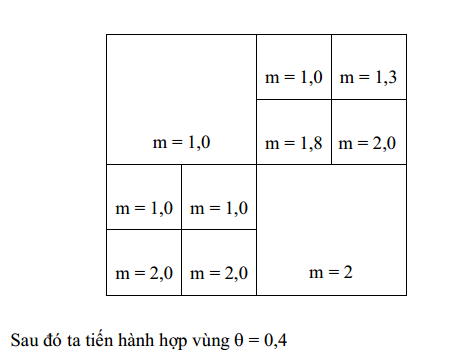
Hai phương pháp trên tuy thực hiện nhanh nhưng có những nhược điểm, ở phương pháp chia sẽ phân mảnh ảnh quá chi tiết, còn phương pháp hợp cho phép giảm số vùng xuống tối thiểu nhưng cấu trúc vùng dàn trải không cho ta thấy rõ được liên hệ của các miền. Do đó ta đưa ra thuật toán kết hợp giữa 2 phương pháp trên.

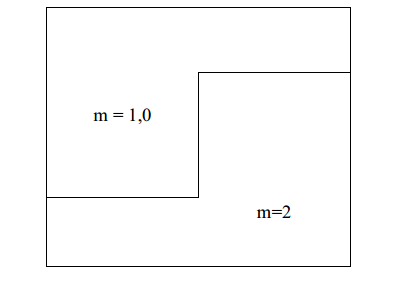
Bước 1: Tiến hành chia ảnh theo cây tứ phân.

Bước 2: Ta xét các vùng liền kề đã được chia trong bước 1, nếu thỏa mãn điều kiện hợp vùng thì tiến hành hợp vùng và tính toán lại giá trị xám cho vùng mới.

Ta xét ví dụ sau







* Phân vùng dựa vào đường biên

Việc phân đoạn ảnh dựa vào biên được tiến hành qua một số bước:

- Phát hiện và làm nổi biên.

- Làm mảnh biên.

- Nhị phân hóa đường biên.

- Mô tả biên.

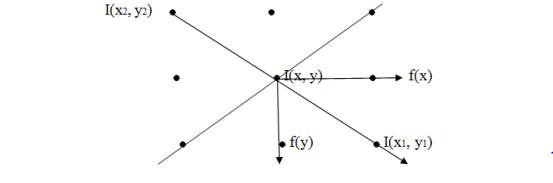
\*Phát hiện và nổi biên

Nhị phân hóa đường biên

Nhị phân hóa đường biên là giai đoạn then chốt trong quá trình chích chọn vì nó xác định đường bao nào thực sự cần và đường bao nào có thể loại bỏ. Ta thường nhị phân hóa đường biên theo cách thức làm giảm nhiễu hoặc tránh hiện tượng kéo sợi trên ảnh. Điều này cũng giải thích tại sao phân vùng hợp ngược lại, có thể sẽ bị mất một phần đường bao hay đường bao có chân, không khép kín... Một phương pháp hay được dùng là chọn ngưỡng thích nghi. Với cách này, ngưỡng sẽ phụ thuộc vào hướng của gradient nhằm làm giảm sự xoắn của biên độ. Lúc đầu ta định ra một ngưỡng nào đó và sau đó sử dụng hệ số sinh thích nghi thông qua lời giải toán tử đạo hàm theo hướng tìm được để tinh chỉnh.

Làm mảnh biên

Làm giảm hay mảnh biên thực chất là làm nổi biên với độ rộng chỉ 1pixel. Theo kỹ thuật gradient, những điểm cực trị cục bộ có thể coi như biên. Do vậy cần tách biệt những điểm cực trị đó để xách định chính xác biên ảnh và để giảm độ rộng biên ảnh.Một phương pháp hay dùng trong làm mảnh biên chữ là phương pháp “Loại bỏ các điểm không cực đại”. Với mỗi điểm ảnh I(x, y), ta xác định các điểm lân cận của nó theo hướng gradient I(x1, y1) và I(x2, y2)



*Hình 1. 17 : Các điểm lân cận gradient*

Nếu I(x, y) lớn hơn cả I(x1, y1) và I(x2, y2) giá trị của I(x, y) sẽ đ ược bảo toàn. Nếu không nó sẽ bị loại bỏ.

Ngoài thuật toán trên còn nhiều kỹ thuật làm mảnh biên như kỹ thuật mảnh biên chữ do Sherman đề xuất sau đó được Fraser cải tiến cho ảnh nhị phân. Kỹ thuật này có thể mô tả tóm tắt như sau:

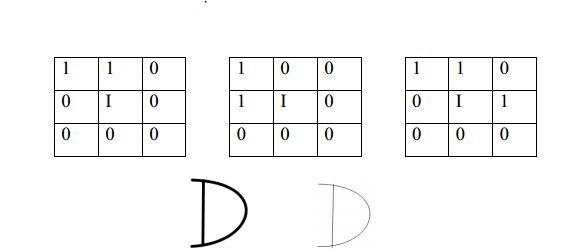
- Gọi I(x,y) là một pixel. Ik(k=1..8) là các pixel lân cận theo 8 láng giềng của I.

- Ta xóa được I nếu thỏa mãn các điều kiện sau đây:

+ I là điểm ảnh.

+ 2≤ số lân cận là điểm ảnh của I < 6; I2\*I4\*I8=0

- Tuy nhiên có một số ngoại lệ do nếu xóa I sẽ gây phân đoạn.

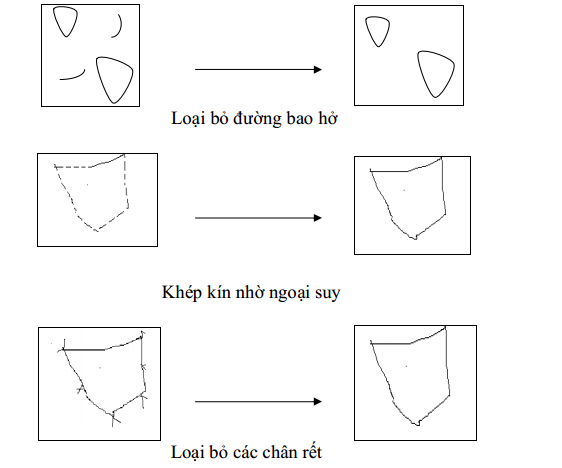


Ảnh gốc Ảnh làm mảnh

*Hình 1. 18 : Làm mảnh chữ*

Miêu tả đường biên

Khi đã có bản đồ biên của ảnh, ta cần phải biểu diễn nó dưới dạng thích hợp phục vụ cho việc phân tích và làm giảm lượng thông tin dùng để miêu tả đối tượng. Nguyên tắc chủ yếu là tách biệt mỗi biên và gán mã.Quá trình miêu tả biên đối tượng là khá rộng vì có nhiều phương án khác nhau và mỗi phương án có liên quan mật thiết với các đặc thù của từng ứng dụng. Việc tách biệt đường bao có thể được tăng cường thêm các điều kiện nhằm loại bỏ các đường bao không khép kín hoặc khép kín nhờ phương pháp ngoại suy, hay bỏ đi các chân rết bám theo các đường bao kín.



*Hình 1. 19 : Căn chỉnh đường biên*

Việc mã hóa đường bao có thể thực hiện theo nhiều cách khác nhau. Có thể dùng biểu diễn chính xác đường bao hay xấp xỉ nhờ nội suy. Thông thường các cấu trúc cơ sở mã hóa đường bao gồm 4 loại: điểm, đoạn thẳng, cung và đường cong.Tuy nhiên, cũng cần lưu ý giữa độ phức tạp tính toán và khả năng biểu diễn ảnh của cách mã hóa thông tin. Biểu diễn đường bao bởi các điểm nói chung không phức tạp song lại rất nghèo nàn về cấu trúc và không cô đọng vì ta phải mô tả tất cả các điểm trên biên.Trong khi đó, biểu diễn bằng đường cong đa thức bậc cao làm tăng đọ phức tạp tính toán, song bù lại cấu trúc dữ liệu lại rất cô đọng.

# CHƯƠNG 2: CÔNG CỤ THIẾT KẾ SẢN PHẨM

## Giới thiệu module ESP8266 NodeMCU

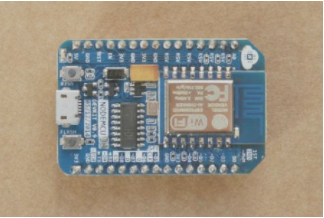
ESP8266 là dạng Vi điều khiển tích hợp Wifi (Wifi SoC) được phát triển bởi Espressif Systems - một nhà sản xuất Trung Quốc có trụ sở tại Thượng Hải. Với Vi điều khiển và Wifi tích hợp, ESP8266 cho phép lập trình viên có thể thực hiện vô số các tác vụ TCP/IP đơn giản để thực hiện vô số các ứng dụng khác nhau, đặc biệt là các ứng dụng IoT. Tuy nhiên, vào thời điểm ra mắt năm 2014, hầu như chỉ có tài liệu bằng tiếng Trung Quốc nên ESP8266 chưa được phổ biến như hiện nay.

Module ESP8266 có giá thành rẻ, chỉ khoảng 2USD cho phiên bản đầu tiên, điều này đã thu hút các IoT-er khám phá cũng như dịch các tài liệu của ESP8266 sang tiếng Anh và phát triển vô số các ứng dụng kèm theo. Sau nhiều năm phát triển, hiện nay đã có hơn 14 phiên bản ESP ra đời, trong đó phổ biến nhất là ESP-12.



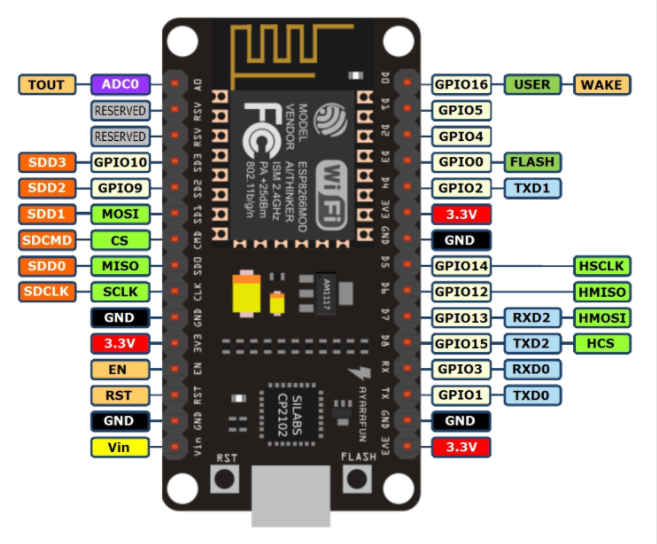
*Hình 2. 1 : Hình ảnh module ESP8266 thực tế*

Module ESP-12 kết hợp với firmware ESP8266 trên Arduino và thiết kế phần cứng giao tiếp tiêu chuẩn đã tạo nên NodeMCU, loại Kit phát triển ESP8266 phổ biến nhất trong thời điểm hiện tại. Với cách sử dụng, kết nối dễ dàng, có thể lập trình, nạp chương trình trực tiếp trên phần mềm Arduino, đồng thời tương tích với các bộ thư viện Arduino sẵn có, NodeMCU là sự lựa chọn hàng đầu.



*Hình 2. 2 : Hình ảnh thực tế ESP NodeMCU*

* Khả năng hoạt động như một modem wifi:
* Có thể quét và kết nối đến một mạng wifi bất kỳ (Wifi Client) để thực hiện các tác vụ như lưu trữ, truy cập dữ liệu từ server.
* Tạo điểm truy cập wifi (Wifi Access Point) cho phép các thiết bị khác kết nối, giao tiếp và điều khiển.
* Là một server để xử lý dữ liệu từ các thiết bị sử dụng internet khác.



*Hình 2. 3 : Sơ đồ chân Pin trên ESP8266 NodeMCU*

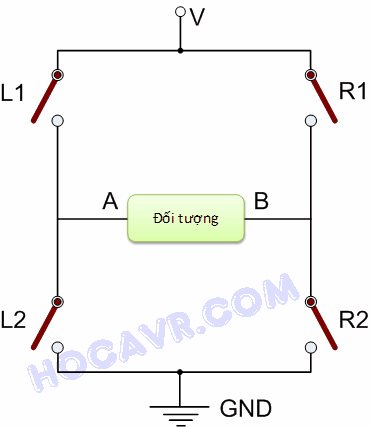
* Nguồn vào và nguồn ra
* ESP8266 NodeMCU nhận nguồn từ cổng micro USB tích hợp sẵn trên mạch, giúp việc nạp code trở nên dễ dàng hơn. Bên cạnh đó, việc cấp nguồn cho module cũng linh động hơn vì bạn có thể sử dụng sạc dự phòng thay cho nguồn từ USB trên máy tính (nguồn cấp tối đa là 5V).
* ESP8266 NodeMCU có thể cung cấp nguồn cho tối đa 4 thiết bị: 3 nguồn ra 3.3V và một nguồn từ chân Vin (điện thế bằng điện thế từ cổng micro USB). Khi sử dụng các chân cấp nguồn này, hãy luôn kiểm tra để chắc chắn không cắm nhầm chân dương (trên mạch in là 3v3 và Vin) và chân âm (GND). Tuy nhiên, 3 chân 3.3V đều được bảo vệ, khi cắm ngược cực, module sẽ chỉ nóng lên và dừng hoạt động. Chân Vin thì KHÔNG, cắm ngược cực ở chân này là module bị cháy.
* Truyền và nhận tín hiệu
* ESP8266 NodeMCU có tổng cộng 13 chân GPIO (General-purpose input/output) – chân có thể truyền/nhận tín hiệu (trên mạch in từ D0 đến D8 và RX, TX, SD2, SD3).

## Module điều khiển động cơ L298

Module L298 tích hợp 2 mạch cầu H có thể điều khiển động cơ 2 động cơ DC hoặc 1 động cơ, có 4 lỗ ở 4 tiện ích thuận tiện cho người dùng sử dụng môđun định vị.

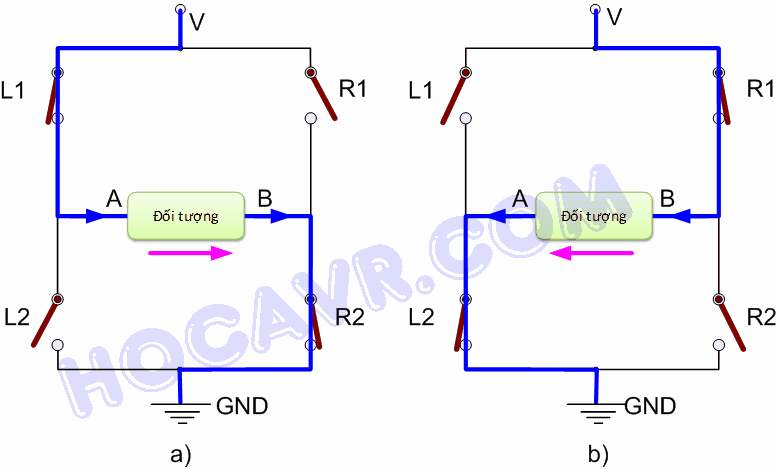
* Mạch cầu H (H-Bridge Circuit):

Giả sử bạn có một động cơ DC có 2 đầu A và B, nối 2 đầu dây này với một nguồn điện DC (ắc qui điện – battery). Ai cũng biết rằng nếu nối A với cực (+), B với cực (-) mà động cơ chạy theo chiều thuận (kim đồng hồ) thì khi đảo cực đấu dây (A với (-), B với (+)) thì động cơ sẽ đảo chiều quay. Tất nhiên khi bạn là một “control guy” thì bạn không hề muốn làm công việc “động tay động chân” này (đảo chiều đấu dây), bạn ắt sẽ nghĩ đến một mạch điện có khả năng tự động thực hiện việc đảo chiều này, mạch cầu H (H-Bridge Circuit) sẽ giúp bạn. Như thế, mạch cầu H chỉ là một mạch điện giúp đảo chiều dòng điện qua một đối tượng. Tuy nhiên, rồi bạn sẽ thấy, mạch cầu H không chỉ có một tác dụng “tầm thường” như thế. Nhưng tại sao lại gọi là mạch cầu H, đơn giản là vì mạch này có hình chữ cái H. Xem minh họa trong hình 1.



*Hình 2. 4 : Mạch cầu H*

Trong hình 1, hãy xem 2 đầu V và GND là 2 đầu (+) và (-) của ắc qui, “đối tượng” là động cơ DC mà chúng ta cần điều khiển, “đối tượng” này có 2 đầu A và B, mục đích điều khiển là cho phép dòng điện qua “đối tượng” theo chiều A đến B hoặc B đến A. Thành phần chính tạo nên mạch cầu H của chúng ta chính là 4 “khóa” L1, L2, R1 và R2 (L: Left, R:Right). Ở điều kiện bình thường 4 khóa này “mở”, mạch cầu H không hoạt động. Tiếp theo chúng ta sẽ khảo sát hoạt động của mạch cầu H thông qua các hình minh họa 2.5.a và 2.5.b.



*Hình 2. 5 : Nguyên lý hoạt động mạch cầu H*

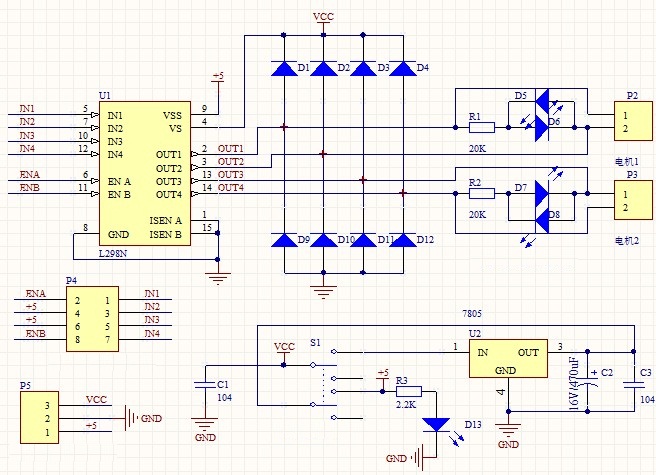
Giả sử bằng cách nào đó (cái cách nào đó chính là nhiệm vụ của người thiết kế mạch) mà 2 khóa L1 và R2 được “đóng lại” (L2 và R1 vẫn mở), bạn dễ dàng hình dung có một dòng điện chạy từ V qua khóa L1 đến đầu A và xuyên qua đối tượng đến đầu B của nó trước khi qua khóa R2 và về GND (như hình 2.2.2a). Như thế, với giả sử này sẽ có dòng điện chạy qua đối tượng theo chiều từ A đến B. Bây giờ hãy giả sử khác đi rằng R1 và L2 đóng trong khi L1 và R2 mở, dòng điện lại xuất hiện và lần này nó sẽ chạy qua đối tượng theo chiều từ B đến A như trong hình 2b (V->R1->B->A->L2->GND). Vậy là đã rõ, chúng ta có thể dùng mạch cầu H để đảo chiều dòng điện qua một “đối tượng” (hay cụ thể, đảo chiều quay động cơ) bằng “một cách nào đó”.

Chuyện gì sẽ xảy ra nếu ai đó đóng đồng thời 2 khóa ở cùng một bên (L1 và L2 hoặc R1 và R2) hoặc thậm chí đóng cả 4 khóa? Rất dễ tìm câu trả lời, đó là hiện tượng “ngắn mạch” (short circuit), V và GND gần như nối trực tiếp với nhau và hiển nhiên ắc qui sẽ bị hỏng hoặc nguy hiểm hơn là cháy nổ mạch xảy ra. Cách đóng các khóa như thế này là điều “đại kị” đối với mạch cầu H. Để tránh việc này xảy ra, người ta thường dùng thêm các mạch logic để kích cầu H, chúng ta sẽ biết rõ hơn về mạch logic này trong các phần sau.

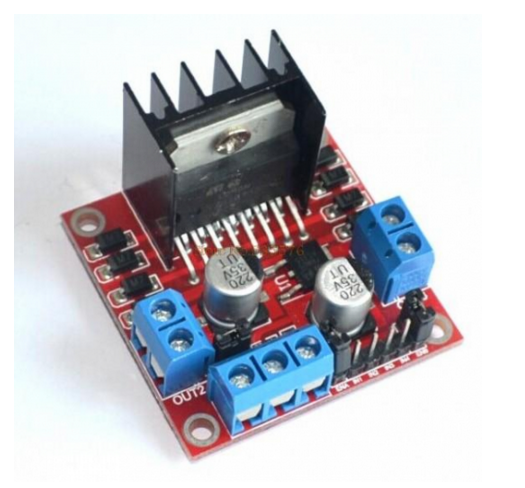
Giả thuyết cuối cùng là 2 trường hợp các khóa ở phần dưới hoặc phần trên cùng đóng (ví dụ L1 và R1 cùng đóng, L2 và R2 cùng mở). Với trường hợp này, cả 2 đầu A, B của “đối tượng” cùng nối với một mức điện áp và sẽ không có dòng điện nào chạy qua, mạch cầu H không hoạt động. Đây có thể coi là một cách “thắng” động cơ (nhưng không phải lúc nào cũng có tác dụng). Nói chung, chúng ta nên tránh trường hợp này xảy ra, nếu muốn mạch cầu không hoạt động thì nên mở tất cả các khóa thay vì dùng trường hợp này.

Sau khi đã cơ bản nắm được nguyên lý hoạt động của mạch cầu H, phần tiếp theo chúng ta sẽ khảo sát cách thiết kế mạch này bằng các loại linh kiện cụ thể. Như tôi đã trình bày trong phần trước, thành phần chính của mạch cầu H chính là các “khóa”, việc chọn linh kiện để làm các khóa này phụ thuộc vào mục đích sử dụng mạch cầu, loại đối tượng cần điều khiển, công suất tiêu thụ của đối tượng và cả hiểu biết, điều kiện của người thiết kế. Nhìn chung, các khóa của mạch cầu H thường được chế tạo bằng rờ le (relay), BJT (Bipolar Junction Transistor) hay MOSFET (Metal Oxide Semiconductor Field-Effect Transistor). Phần thiết kế mạch cầu H vì vậy sẽ tập trung vào 3 loại linh kiện này. Trong mỗi cách thiết kế, tôi sẽ giải thích ngắn gọn nguyên lý cấu tạo và hoạt động của từng loại linh kiện để bạn đọc dễ nắm bắt hơn.

* Thông số kĩ thuật



*Hình 2. 6 : Sơ đồ nguyên lí module L298*



*Hình 2. 7 : Hình ảnh thực tế module L298*

* Có gắn tản nhiệt chống nóng cho IC, giúp IC có thể điều khiển với dòng đỉnh đạt 2A.
* IC L298N được gắn với các đi ốt trên board giúp bảo vệ vi xử lý chống lại các dòng điện cảm ứng từ việc khởi động/ tắt động cơ.
* Điện áp điều khiển : +5V ~ +12 V
* Dòng tối đa cho mỗi cầu H là :2A
* Điện áp của tín hiệu điều khiển : +5 V ~ +7 V
* Dòng của tín hiệu điều khiển : 0 ~ 36Ma
* Công suất hao phí : 20W (khi nhiệt độ T = 75 °C)
* Nhiệt độ bảo quản : -25°C ~ +130
* Công dụng

IC L298 là một IC tích hợp nguyên khối gồm 2 mạch cầu H bên trong. Với điện áp làm tăng công suất nhỏ như động cơ DC loại vừa…

Mình tóm tắt qua chức năng các chân của L298

- 4 chân INPUT: IN1, IN2, IN3, IN4 được nối lần lượt với các chân 5, 7, 10, 12 của L298. Đây là các chân nhận tín hiệu điều khiển.

- 4 chân OUTUT: OUT1, OUT2, OUT3, OUT4 (tương ứng với các chân INPUT) được nối với các chân 2, 3,13,14 của L298. Các chân này sẽ được nối với động cơ.

- Hai chân ENA và ENB dùng để điều khiển mạch cầu H trong L298. Nếu ở mức logic “1” (nối với nguồn 5V) cho phép mạch cầu H hoạt động, nếu ở mức logic “0” thì mạch cầu H không hoạt động

Với bài toán của mình ở trên, các bạn chỉ cần lưu ý đến cách điều khiển chiều quay với L298:

- Khi ENA = 0: Động cơ không quay với mọi đầu vào.

- Khi ENA = 1:

INT1 = 1; INT2 = 0: Động cơ quay thuận.

INT1 = 0; INT2 = 1: Động cơ quay nghịch.

INT1 = INT2: Động cơ dùng ngay tức thì.

Với ENB cũng tương tự với INT3, INT4.

## 2.3. Giới thiệu về OpenCV

### *2.3.1. Giới thiệu về thư viện OpenCV*

OpenCV (Open Source Computer Vision) là một thư viện mã nguồn mở về thị giác máy với hơn 500 hàm và hơn 2500 các thuật toán đã được tối ưu về xử lý ảnh, và các vấn đề liên quan tới thị giác máy. OpenCV được thiết kế một cách tối ưu, sử dụng tối đa sức mạnh của các dòng chip đa lõi để thực hiện các phép tính toán trong thời gian thực, nghĩa là tốc độ đáp ứng của nó có thể đủ nhanh cho các ứng dụng thông thường. OpenCV là thư viện được thiết kế để chạy trên nhiều nền tảng khác nhau (cross-patform), nghĩa là nó có thể chạy trên hệ điều hành Window, Linux, Mac, iOS,… Việc sử dụng thư viện OpenCV tuân theo các quy định về sử dụng phần mềm mã nguồn mở BSD do đó bạn có thể sử dụng thư viện này một cách miễn phí cho cả mục đích phi thương mại lẫn thương mại.

Dự án về OpenCV được khởi động từ những năm 1999, đến năm 2000 nó được giới thiệu trong một hội nghị của IEEE về các vấn đề trong thị giác máy và nhận dạng, tuy nhiên bản OpenCV 1.0 mãi tới tận năm 2006 mới chính thức được công bố và năm 2008 bản 1.1(pre-release) mới được ra đời. Tháng 10 năm 2009, bản OpenCV thế hệ thứ hai ra đời (thường gọi là phiên bản 2.x), phiên bản này có giao diện của C++ (khác với phiên bản trước có giao diện của C) và có khá nhiều điểm khác biệt so với phiện bản thứ nhất. Phiên bản OpenCV hiện nay có giao diện chương trình với C, C++, Python và Android.

Thư viện OpenCV ban đầu được sự hỗ trợ từ Intel, sau đó được hỗ trợ bởi Willow Garage, một phòng thí nghiệm chuyên nghiên cứu về công nghệ robot. Cho đến nay, OpenCV vẫn là thư viện mở, được phát triển bởi nguồn quỹ không lợi nhuận (none -profit foundation) và được sự hưởng ứng rất lớn của cộng đồng.

Cấu trúc tổng quan của OpenCV bao gồm 5 phần chính. 4 trong 5 phần đó được chỉ ra trong hình vẽ dưới.

**CV**

Các hàm về xử lý ảnh và giải thuật về thị giác máy tính

**CXCORE**

Các cấu trúc dữ liệu cơ bản, cấu trúc XML, các hàm về đồ họa …

**MLL**

Các thuật toán học máy, bao gồm các bộ phân cụm, phân loại thống kê

**HighGUI**

Các hàm và thủ tục làm việc với file ảnh và file video

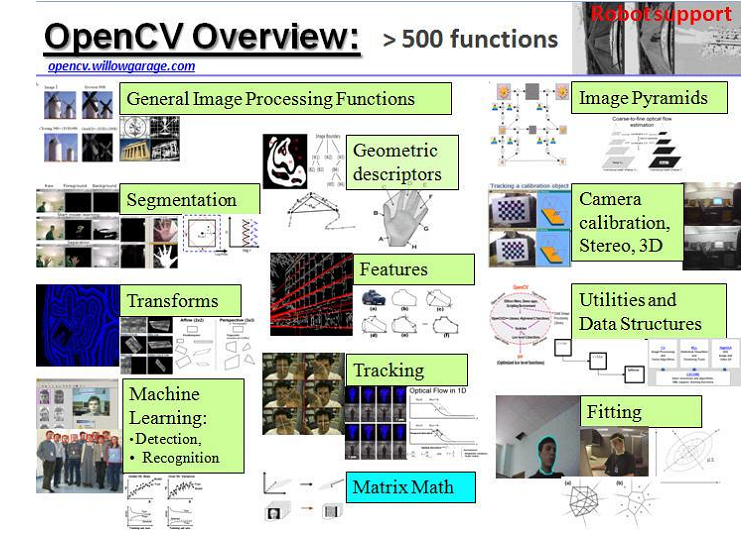
*Hình 2. 8 : Cấu trúc cơ bản của OpenCV*

Phần CV bao gồm các thư viện cơ bản về xử lý ảnh và các giải thuật về thị giác máy tính. ML là bộ thư viện về các thuật toán học máy, bao gồm rất nhiều bộ phân cụm và phân loại thống kê. HighGUI chứa đựng những thủ tục vào ra, các chức năng về lưu trữ cũng như đọc các file ảnh và video. Phần thứ 4, Cxcore chứa đựng các cấu trúc dữ liệu cơ bản (ví dụ như cấu trúc XML, các cây dữ liệu …). Phần cuối cùng là CvAux, phần này bao gồm các thư viện cho việc nhận diện, theo dõi và nhận dạng đối tượng (khuôn mặt, mắt …).

OpenCV cung cấp rất nhiều kiểu dữ liệu có cấu trúc nhằm hỗ trợ tối việc xử lí.

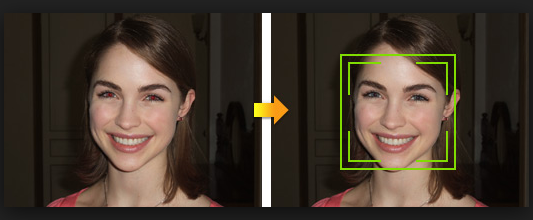
Ứng dụng công nghệ nhận dạng:

* Hiệu chỉnh và phục chế ảnh.
* Xử lí màu sắcvà độ phân giải ảnh.
* Xử lí hình trạng và phân tách chi tiết.
* Tái dựng ảnh ba chiều và diễn giải ảnh.
* Nhận dạng và giám sát thực thể.
* Y khoa (chẩn đoán, giải phẫu).
* An ninh (camera thông minh, nhận dạng)
* Giao thông (theo dõi lưu thông, điều khiển xe)
* Tự động hóa sản xuất (đo lường, kiểm định chất lượng) và v.v…



*Hình 2. 9 : Một số ứng dụng của OpenCV*

OpenCV có thể tải về và cài đặt trên Window, chạy trên phần mềm Microsoft Visual Studio. OpenCV thường được lập trình bằng C/C++, xem xét nhận diện bức ảnh bằng OpenCV được xử lý trên máy tính.

****

*Hình 2. 10 : Xác định và nhận diện khuôn mặt trong ảnh*

OpenCV nhận dạng mặt bằng lớp Haar Cascade. Hình trên cho ta thấy một ví dụ về nhận dạng mặt của OpenCV.

Với một ảnh có thể lấy từ một file hoặc từ file video trực tiếp thì nhận dạng ảnh nghiên cứu mỗi ảnh và phân loại chúng thành “Mặt” và “không mặt”. Sự phân loại này cho rằng có một tỷ lệ có sẵn cho mặt, chọn điểm ảnh 50x50. Một mặt trong ảnh có thể lớn hơn hoặc nhỏ hơn giá trị này. Sự phân loại hướng về ảnh trong một khoảng thời gian, và tìm kiếm mặt cắt vượt qua giới hạn chia.

### *2.3.2. Một số hàm xử lý ảnh nhận dạng cơ bản của OpenCV*

* Ngôn ngữ lập trình Python

Trong đồ án này sẽ sử dụng ngôn ngữ Python để lập trình.Hiện nay có rất nhiều ngôn ngữ lập trình mạnh mẽ và phổ biến như: C, C++, Java,… Có rất nhiều chương trình mà khi viết bằng C, C++, Java sẽ gặp rất nhiều khó khăn, phức tạp, tiêu tốn nhiều thời gian cho việc thực hiện nó thậm chí từ những bản nháp đầu tiên của chương trình. Đối với ngôn ngữ lập trình Python sử dụng đơn giản hơn, chạy được cả trên Windows, MacOS X, và các hệ điều hành Unix, nó cũng giúp bạn hoàn thành công việc nhanh hơn.

Python cho phép viết các chương trình nhỏ gọn và dễ hiểu. Các chương trình viết bằng Python thường ngắn hơn so với các chương trình viết bằng C, C++ hoặc Java, vì nhiều lý do:

* Các kiểu dữ liệu cao cấp cho phép bạn thực hiện nhanh các thao tác phức tạp chỉ với một lệnh đơn giản.
* Phát biểu lệnh được nhóm lại bằng khoảng cách thụt đầu dòng thay vì đóng mở với các dấu ngoặc.
* Không cần khai báo biến hoặc tham số trước khi sử dụng.
* Python hỗ trợ nhiều module xử lý hình ảnh và ma trận, do đó việc lập trình trở nên đơn giản hơn

Trong các ngôn ngữ khác, khối lệnh thường được đánh dấu bằng cặp kí hiệu hoặc từ khóa. Ví dụ, trong C/C++, cặp ngoặc nhọn { } được dùng để bao bọc một khối lệnh. Python, trái lại, có một cách rất đặc biệt để tạo khối lệnh, đó là thụt các câu lệnh trong khối vào sâu hơn (về bên phải) so với các câu lệnh của khối lệnh cha chứa nó.

Ví dụ, giả sử có đoạn mã sau trong C/C++:

1. *#include <math.h>*
2. //...
3. delta = b \* b – 4 \* a \* c;
4. if (delta > 0)
5. {
6. // Khối lệnh mới bắt đầu từ kí tự { đến }
7. x1 = (- b + sqrt(delta)) / (2 \* a);
8. x2 = (- b - sqrt(delta)) / (2 \* a);
9. printf("Phuong trinh co hai nghiem phan biet:\n");
10. printf("x1 = %f; x2 = %f", x1, x2);
11. }

Đối với Python:

1. import math
2. *#...*
3. delta = b \* b – 4 \* a \* c
4. if delta > 0:
5. *# Khối lệnh mới, thụt vào đầu dòng*
6. x1 = (- b + math.sqrt(delta)) / (2 \* a)
7. x2 = (- b – math.sqrt(delta)) / (2 \* a)
8. print "Phuong trinh co hai nghiem phan biet:"
9. print "x1 = ", x1, "; ", "x2 = ", x2

* Sử dụng các hàm trong ngôn ngữ Python.

Lọc hình ảnh: Hàm và các lớp học được mô tả trong phần này được sử dụng để thực hiện khác nhau, tuyến tính hoặc phi tuyến tính lọc các hoạt động trên hình ảnh 2D (đại diện là cv.’s), có nghĩa là, đối với mỗi vị trí pixel (x, y) trong nguồn hình ảnh một số khu vực (thường là hình chữ nhật) được xem xét và được sử dụng để tính toán các phản ứng. Trong trường hợp của một bộ lọc tuyến tính, nó là một tổng trọng số của các giá trị pixel, trong trường hợp của hình thái học hoạt động đó là ...tối thiểu hoặc tối đa phản ứng tính toán được lưu trữ đến đích hình ảnh ở cùng một vị trí (x, y).

Một số module được sử dụng trong bài như numpy, PIL, os là các module tích hợp nhiều hàm xử lý các ma trận, vector, hình ảnh, file, …

* Hàm: cv.CreateStructuringElementEx – Tạo một yếu tố cấu trúc.

CreateStructuringElementEx (cols, rows, anchorX, anchorY, shape, values = None) -> kernel

Cols: Số cột trong các yếu tốcấu trúc.

Rows: Số hàng trong các yếu tố cấu trúc.

Anchor: Tương đối ngang bù đắp của các điểm neo.

AnchorY: Thẳng đứng tương đối bù đắp của các điểm neo.

Shape: Hình dạng của các yếu tố cấu trúc, có thể có giá trị sau:

CV\_SHAPE\_RECT một yếu tố hình chữ nhật.

CV\_SHAPE\_CROSS một yếu tố hình chữ thập.

CV\_SHAPE\_ELLIPSE một yếu tố elip.

CV\_SHAPE\_CUSTOM một yếu tố người dùng định nghĩa. Trong trường hợp này các values tham số quy định cụ thể mặt nạ, có nghĩa là, những vùng lân cận của điểm ảnh phải được xem xét

* Hàm: cv.CvtColor – Chuyển đổi một ảnh từ không gian màu khác.

CvtColor (src, dst, code) -> None

Src: Với nguồn 8bit hoặc 16bit

Dst: Hình ảnh đích giống như kiểu dữ liệu của nguồn, số lượng kênh có thể khác nhau.

Code: Quá trình biến đổi mầu có thể sử dụng: CV\_src\_color\_ space 2 dst\_color\_space.

* Hàm: cv.CreateImage – tạo ra tiêu đề hình ảnh và được sắp xếp theo dữ liệu của ảnh

CreateImage (size, depth, channels) ->image

Size: Chiều rộng và chiều cao của ảnh.

Depth: Độ sâu bit của các yếu tố hình ảnh.

Channels: Số lượng kênh cho mỗi điểm ảnh

* Hàm: cv.Resize – Kích thước yêu cầu của một ảnh.

Resize (src, dst, interpolation=CV\_INTER\_LINEAR) -> None

Src: Ảnh nguồn

Dst: Ảnh đích

Interpolation: phương pháp nội suy

CV\_INTER\_NN: Nội suy lận cận gận nhất.

CV\_INTER\_LINEAR: Nội suy song tuyến (sử dụng mặc định).

CV\_INTER\_AREA: Lấy lại mẫu sự dụng quan hệ của các điểm ảnh.

CV\_INTER\_CUBIC: Nội suy song lập phương.

* Hàm cv.Load – Load một đối tượng từ file

Load (filename, storage=NULL, name=NULL) -> generic

Filename: tên của file.

Storage: Bộ nhớ lưu trữ cho các cấu trúc động. Nó không sử dụng cho ma trận và hình ảnh.

Name: Tùy chọn vào tên đối tượng. Nếu là NULL, sẽ nạp các đối tượng mức cao nhất đầu tiên.

Hàm này dùng để load một đối tượng từ một tập tin. Nó cung cấp giao diện đơn giản cho hàm cv.Read. Sau khi tập tin được tải lên, lưu trữ tập tin đóng lại và bộ nhớ đệm dữ liệu sẽ bị xóa. Vì vậy để tải một cấu trúc động, ví dụ như một chuỗi, đường viền hay biểu đồ, ta phải thông qua một bộ nhớ hợp lệ đích lưu trữ của hàm.

* Hàm cv.Save – lưu trữ một đối tượng từ một file

Save (filename, structPtr, name=NULL, comment=NULL) -> None

Filename: Tên file

structPtr: Đối tượng lưu.

Name: Tên của đối tượng, nếu mà là NULL, tên sẽ có hình thức giống filename.

Coment: Tùy chọn những lời dẫn đưa vào trong file.

* Hàm: cv.HaarDetectObjects – nhận diện đối tượng trong hình ảnh

HaarDetectObjects (image, cascade, storage, scale\_factor=1.1, min\_neighbors=3, flags=0, min\_detected objects)

Image: Hình ảnh được sử dụng để nhận diện đối tượng.

Cascade: Haar phân tầng trong những đặc trưng nội bộ.

Storage: bộ nhớ lưu trữ để lưu trữ kết quả của các đối tượng ứng viên hình chữ nhật.

Scale\_factor: Các yếu tố mà cửa sổ tìm kiếm thu nhỏ sau mỗi lần quét, 1.1 nghĩa là tăng cửa sổ lên 10%.

Min-neighbors: số lượng tối thiểu (-1) của hình chữ nhật lân cận để tạo nên một đối tượng. Tất cả các đối tượng hình chữ nhật lân cận nhỏ hơn min\_neighbors -1 bị từ chối. Nếu min\_neighbors là 0 thì hàm này không có ở tất cả các nhóm và trả về tất các các nhận diện hình chữ nhật của ứng viên.

Flags: Chế độ hoạt động, flag duy nhất có thể xác định và nhận diện được là CV\_HAAR\_DO \_CANNY\_PRUNING. Nếu mà nó được cài đặt, hàm này sẽ sử dụng CANNY nhận diện cạnh để từ chối một số đối tượng có quá ít hoặc quá nhiều cạnh, do đó ko thể chứa các đối tượng tìm kiếm.

Min\_size: kích thước nhỏ nhất của cửa sổ. Theo mặc định nó được thiết lập cho kích thước mẫu phân loại (thường là 20x20 cho nhận diện khuôn mặt).

### *2.3.3. Chuyển đổi các không gian màu trong OpenCV*

Để chuyển đổi các không gian màu trong OpenCV ta dùng hàm sau:

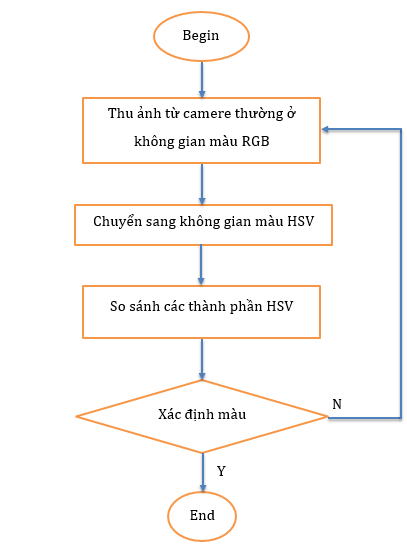
cvCvtColor(const CvArr \*src, CvArr \*dst, int code). Trong đó, src là ảnh đầu vào, dst là ảnh đích chứa kết quả chuyển đổi. code là mã chuyển đổi, ví dụ ta cần chuyển từ RGB sang HSV thì code là CV\_RGB2HSV ...

# CHƯƠNG 3: XỬ LÍ HÌNH ẢNH VÀ PHÂN LOẠI MÀU

## 3.1. Phát hiện đối tượng

### *3.1.1.* *Phương pháp phân loại vật dựa trên màu sắc*

Ảnh chụp thu thập được từ camera sẽ cho không gian màu của ảnh là RBG, để thuận tiện cho việc phân loại màu sắc ta phải chuyển sang không gian màu HSV.



*Hình 3. 1 : Lưu đồ giải thuật nhận dạng*

Không gian màu RGB không thích hợp cho việc nhận dạng màu sắc, để phân biệt màu sắc người ta thường dùng không gian màu HSV. Trong OpenCV, H (Hue) có giá trị trong khoảng (0, 179) mang thông tin về màu sắc. S (Saturation) có giá trị (0, 255) cho thông tin về độ thuần khiết của màu. V (Value) có giá trị (0, 255) cho biết độ sáng của điểm ảnh. Chuyển đổi không gian màu từ RGB sang HSV:

*cvCvtColor(rgbimage,hsvimage,CV BGR2HSV);*

Ví dụ : Màu đỏ thường thành phần H có giá trị trong khoảng (0 - 7) & (174 - 179). Các thành phần S và V thường chọn trong lân cận 100 tùy theo điều kiện ánh sáng môi trường và yêu cầu về độ thuần khiết của màu.

Dựa vào đây ta có thể tìm ra mã màu HSV của từng màu và từ đó nhận dạng, phân biệt màu trong môi trường cụ thể.

## 3.2. Phát hiện tọa độ của đối tượng có màu cần tìm

### *3.2.1 Quá trình phát hiện tọa độ của đối tượng*

“Khi đã có mã màu HSV của các sản phẩm cần phân loại, việc phân loại màu sắc trở nên dễ dàng”

Sau khi tìm được các vùng màu trùng với mã màu HSV theo yêu cầu, ta cần chuyển về ảnh xám để tiện cho việc tìm đường bao (Contour )quanh các màu đó.

Contour thường thu được từ các cạnh, mục đích để tạo các đường cong khép kín để vẽ đường nét đối tượng , dùng thuật toán phát hiện cạnh (edge detection ) Canny để tìm cạnh của đối tượng, phân biệt với các vùng màu khác, sau đó liên kết các cạnh đó lại để tạo ra đường cong khép kín ta được Contour.

* Phương pháp Canny

Phương pháp này sử dụng hai mức ngưỡng cao và thấp. Ban đầu ta dùng mức ngưỡng cao để tìm điểm bắt đầu của biên, sau đó chúng ta xác định hướng phát triển của biên dựa vào các điểm ảnh liên tiếp có giá trị lớn hơn mức ngưỡng thấp. Ta chỉ loại bỏ các điểm có giá trị nhỏ hơn mức ngưỡng thấp. Các đường biên yếu sẽ được chọn nếu chúng được liên kết với các đường biên khỏe.

Phương pháp Canny bao gồm các bước sau:

Bước 1. Trước hết dùng bộ lọc Gaussian để làm mịn ảnh.

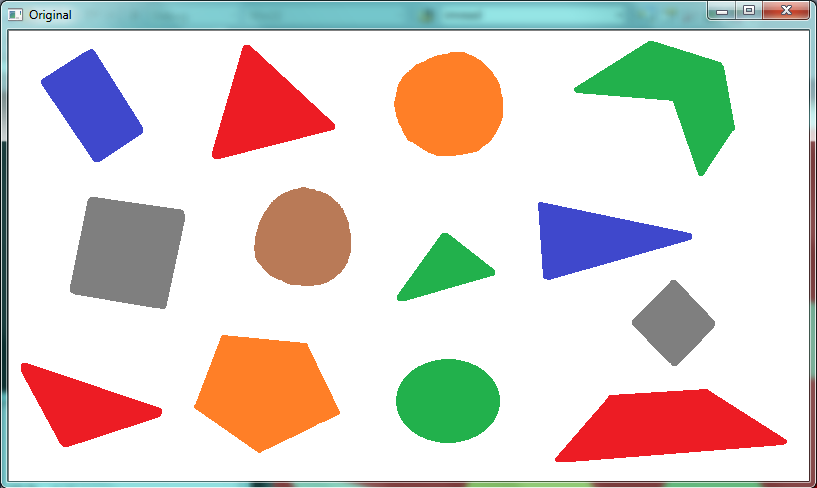
Bước 2. Sau đó tính toán gradient của đường biên của ảnh đã được làm mịn.

Bước 3. Tiếp theo là loại bỏ những điểm không phải là cực đại.

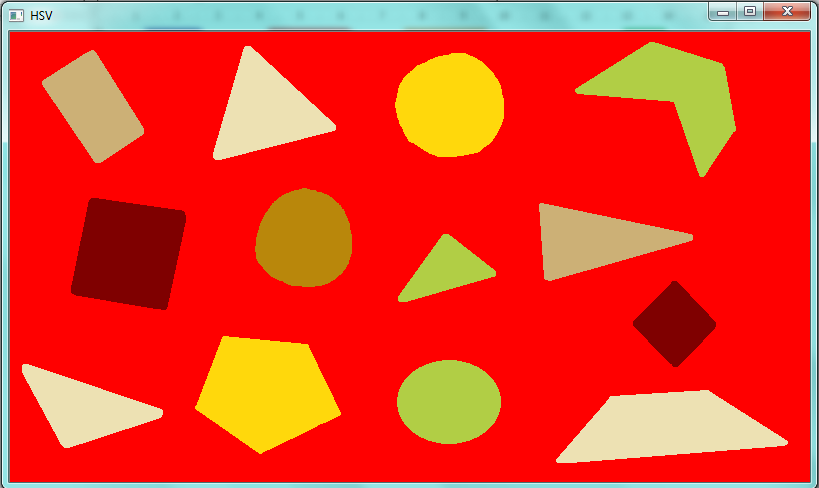
Bước 4. Bước cuối cùng là loại bỏ những giá trị nhỏ hơn mức ngưỡng.

Phương pháp này hơn hẳn các phương pháp khác do ít bị tác động của nhiễu và cho khả năng phát hiện các biên yếu. Nhược điểm của phương pháp này là nếu chọn ngưỡng quá thấp sẽ tạo ra biên không đúng, ngược lại nếu chọn ngưỡng quá cao thì nhiều thông tin quan trọng của biên sẽ bị loại bỏ.

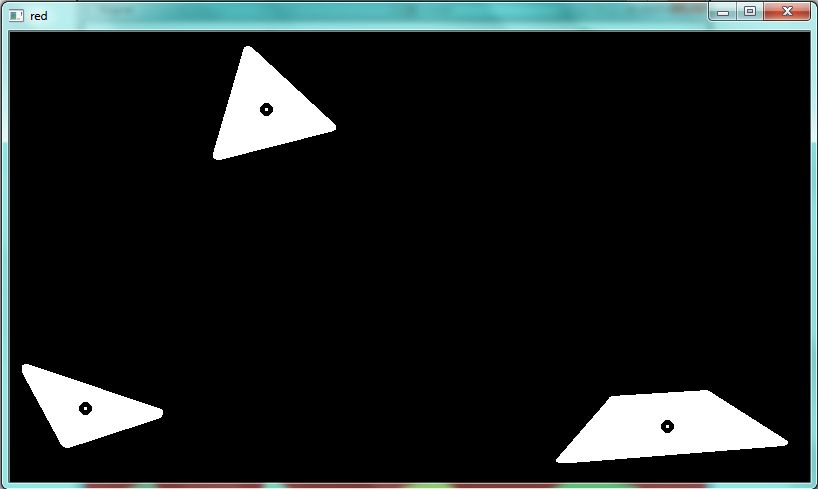
Hình ảnh sơ lượt về quá trình tìm toạ độ của các đối tượng theo màu



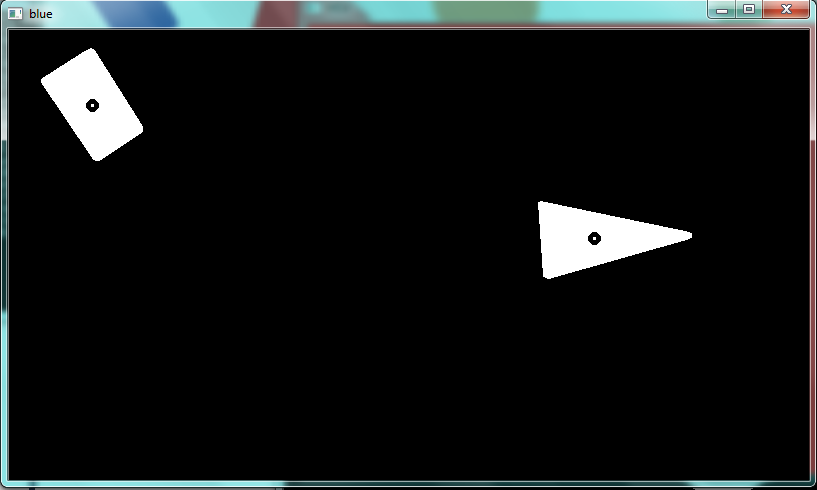
*Hình 3. 2 : Hình gốc dùng để nhận dạng màu*



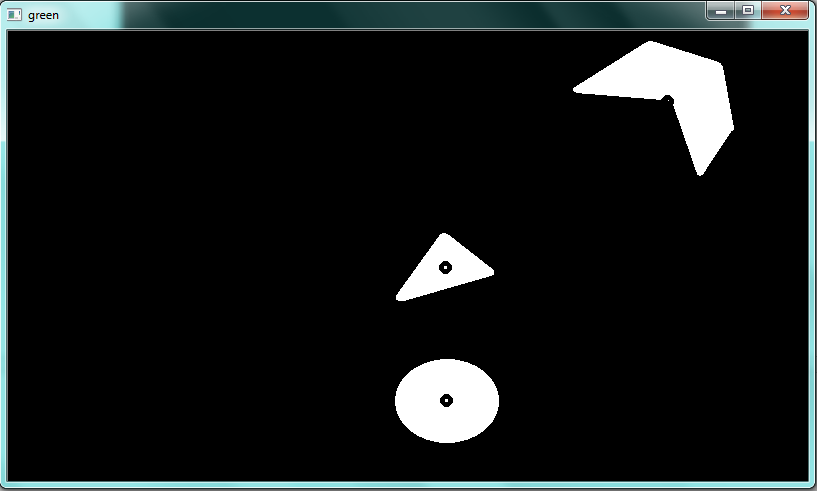
*Hình 3. 3 : Hình sau khi được chuyển sang mã màu HSV*



*Hình 3. 4 : Hình thể hiện vùng có màu đỏ(chấm đen là tâm của ảnh mỗi vật , toạ độ của chấm den được coi như là toạ độ của vật đó trong hình )*



*Hình 3. 5 : Hình thể hiện vùng có màu xanh da trời (chấm đen là tâm của ảnh mỗi vật , toạ độ của chấm den được coi như là toạ độ của vật đó trong hình )*



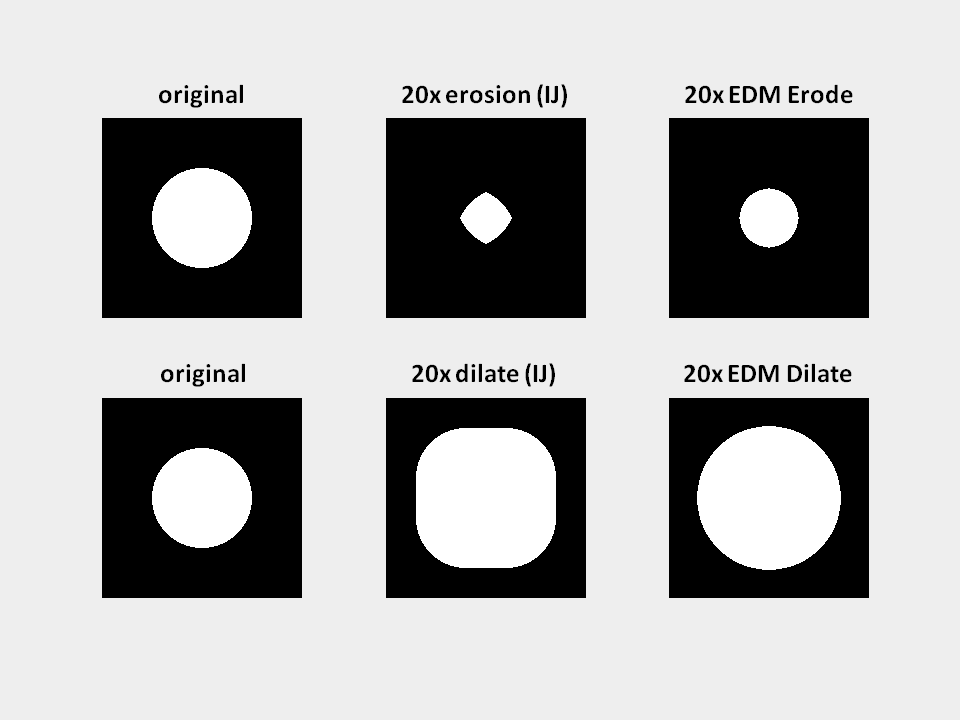
*Hình 3. 6 : Hình thể hiện vùng có màu xanh lá cây (chấm đen là tâm của ảnh mỗi vật , toạ độ của chấm den được coi như là toạ độ của vật đó trong hình )*

### *3.2.2 Loại bỏ nhiễu cho ảnh*

Trong trường hợp một bức hình thu được từ camera có các đối tượng đồng thời cũng có những điểm nhỏ trong hình có cũng màu với các đối tượng, để loại bỏ được các điểm không cần thiết và gây ảnh hưởng đến với quá trình phát hiện vật ta dùng phép biến đổi ảnh hình thái cơ bản là dilation( giãn nở ) và erosion( xói mòn ), chúng được ứng dụng trên phạm vi rộng như lọc nhiễu, tách bạch các đối tượng, nối các đối tượng riêng rẽ trên ảnh, và còn thường được dùng để tìm những nơi có cường độ độ sáng lớn hay nhỏ ( các hố, lỗ ) trên ảnh để tìm chiều hướng của bức ảnh.

Giãn nở và ăn mòn là hai phép toán hình thái tổng quát và đây là hai quá trình ngược nhau. Trong phép giản nở, các đường biên của một đối tượng trong ảnh được cộng thêm các pixel trong khi đó ở phép ăn mòn các đường biên của một đối tượng nào đó trong ảnh lại bị bỏ bớt đi. Số lượng các pixle được cộng thêm hay bớt đi khỏi đối tượng ở trong ảnh phụ thuộc vào kích thước và hình dáng của cấu trúc phần tử được dùng để xử lý. Sự giãn nở có xu hướng mở rộng các đường biên, đường viền, các vùng, trong khi sự ăn mòn có xu hướng giảm hoặc thậm chí loại bỏ các vùng nhỏ.

Quá trình ăn mòn và giãn nở có thể kết hợp lại với nhau hoặc có thể thực hiện một cách song hành để tạo nên những phương pháp xử lý hình thái phức tạp hơn. Vì cả sự ăn mòn lẫn giãn nở đều là phép toán phi tuyến, nên chúng không có nghịch đảo, vì vậy có thể thực hiện phép này sau phép kia trên cùng một vùng của một ảnh. Nếu thực hiện quá trình ăn mòn trước rồi tiếp đến quá trình giãn nở thì ta có phép mở. Trường hợp nếu ảnh là nhị phân thì phép toán mở có khuynh hướng làm mất các đối tượng có kích thước nhỏ ở trong ảnh mà không làm thay đổi kích thước và hình dạng các đối tượng lớn. Nếu sự giãn nở thực hiện trước, tiếp đến sự ăn mòn thì quá trình được gội là đóng. Phép toán đóng liên kết các đối tượng xít lại với nhau và do vậy nó có khuynh hướng lấp đầy các lỗ hổng nhỏ và làm nhẵn các đường nét của một đối tượng bằng cách lấp đầy những khe nhỏ.



*Hình 3. 7 : Ăn mòn và giãn nở ảnh*

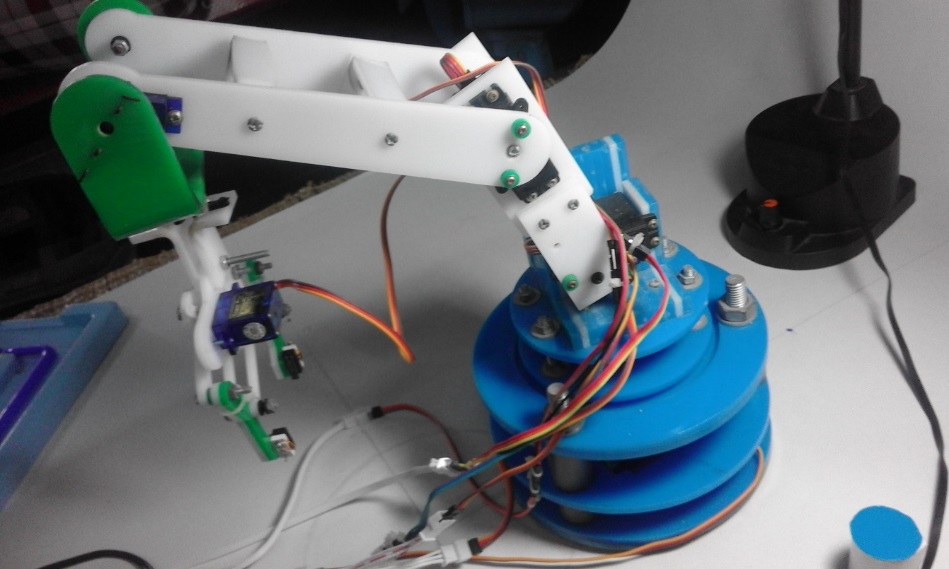
Ở đây ta thực hiện quá trình ăn mòn trước rồi tiếp đến quá trình giãn nở.

# CHƯƠNG 4: THIẾT KẾ VÀ THI CÔNG HỆ THỐNG

## 4.1. Thiết kế mô hình

Mô hình gồm:

* Khung xe mô hình
* Module điều khiển động cơ L298
* Bệ gắn camera
* Module wifi ESP 8266
* Pin nguồn
* Máy tính
* Điện thoại android



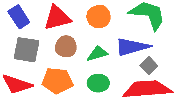
*Hình 4. 1 : Mô hình thực tế*

## 4.2. Hệ thống điều khiển



Thu nhận ảnh

Truyền thông tin điều khiển



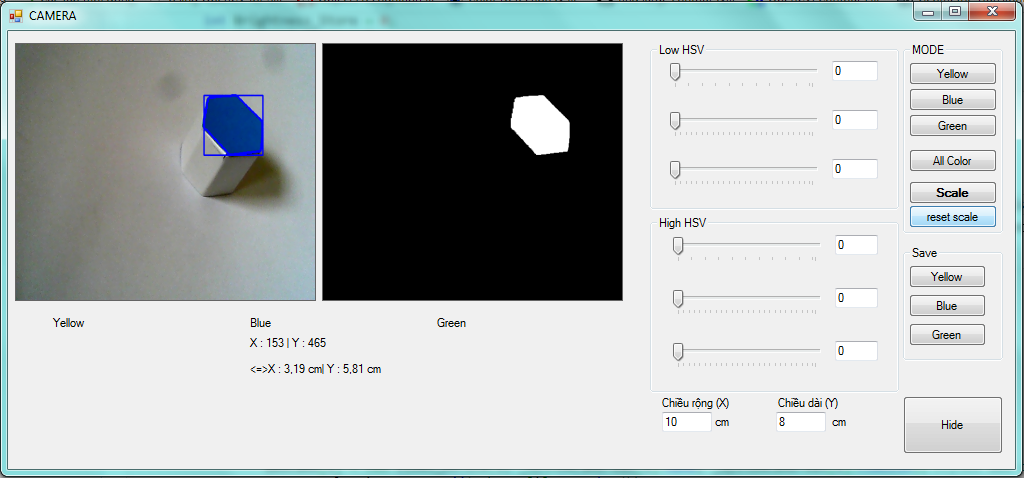


Xuất tín hiệu điều khiển cho xe

*Hình 4. 2 : Tổng quan hoạt động*



*Hình 4. 3 : Giao diện phần mềm*



*Hình 4. 4 : Giao diện xử lý ảnh*

* Nguyên lí hoạt động:
* Máy tính, điện thoại, module wifi Esp 8266 được kết nối chung với một mạng Lan. Hình ảnh được thu từ camera điện thoại trên phần mêm IP Webcam nền tảng Androi truyền tới máy tính.
* Máy tính thu nhận hình ảnh webcam và thực hiện các bước tiền xử lý. Khi có mỗi khung hình mới ( frame) ở hệ màu RGB chuyển sang HSV rồi lọc các màu gần nhau với màu của vật mình đã khóa từ đó thu được một mặt nạ ( mask)
* Ta nhận diện mặt nạ đó bằng cách gọi hàm lấy đường viền (contour). Trong thực tế nó rât nhiễu, có nhiều mẩu màu nhỏ màu trùng nhau nên để đơn giản ta lấy cái nào to nhất bằng hàm tính diện tích tìm ra cái to nhất rồi tìm tâm x,y của vật so sánh tâm video.
* Khi vật lệch với ngưỡng do ta đặt ra, nếu lệch bên trái hoặc bên phải thì chương trình trả về tín hiệu.
* Tín hiệu truyền về động cơ để để thực hiện quá trình rẽ trái hoặc rẽ phải.

# KẾT LUẬN.

* Kết quả đã đạt được những nội dung cụ thể sau:
* Xe cơ bản nhận diện được màu sắc của vật, đi theo vật.

Bên cạnh đó, sản phẩm vẫn còn một số hạn chế như:

* Do hình ảnh truyền về có độ trễ nhất định nên việc xử lý gặp khó khăn dẫn đến việc xe phải đi giật không được đều.
* Việc xử lý hoàn toàn phụ thuộc máy tính.
* Chưa nhận diện được các màu.
* Về mặt thẩm mĩ xe vẫn còn thô sơ.
* Bị nhiễu khá nhiều
* Hướng phát triển của sản phẩm:
* Tích hợp bộ xử lý hình ảnh trên mô hình xe để có thể di động dễ dàng, linh hoạt với khoảng cách xa hơn.
* Xây dựng
* Thiết kế mô hình xe thẩm mĩ nhiều hơn để có thể đưa sản phẩm ra sản xuất công nghiệp.
* Đưa sản phẩm vào ứng dụng trong việc do thám, quân sự mà không cần phải dùng nhân lực thực tế
* Mặc dù đã hết sức cố gắng để có thể hoàn thiện sản phẩm đúng hạn nhưng đồ án tốt nghiệp khó tránh khỏi những thiếu sót. Em rất mong nhận được sự góp ý, nhận xét từ quý thầy cô để sản phẩm có thể hoàn thiện và có tính thực tiễn cao hơn.

Em xin chân thành cảm ơn.

# TÀI LIỆU THAM KHẢO

## Danh mục tài liệu tham khảo

1. Bài giảng “Hệ thống nhúng” - 2010 - TS. Nguyễn Ngọc Minh - Học viện công nghệ bưu chính viễn thông.
2. Bài giảng “Xử lý ảnh” - 2006 – PGS.TS. Nguyễn Quang Hoang - Học viện công nghệ bưu chính viễn thông.

## Danh mục website tham khảo

1. [www.adafruit.com](http://www.adafruit.com)
2. [www.esp8266.com](http://www.esp8266.com)
3. www.adruino.vn