**Chương 4 THIẾT KẾ HỆ THỐNG ĐIỀU KHIỂN DẪN HƯỚNG BẰNG TÍN HIỆU HÌNH ẢNH**

**4.1 Giới thiệu về giải thuật xử lí ảnh và bộ thư viện xử lí ảnh mã nguồn mở OpenCV**

**4.1.1 Giải thuật xử lí ảnh**

Hiện tại, cùng với sự phát triển vượt bậc của phần cứng máy tính, xử lí ảnh và đồ họa máy tính phát triển rất mạnh mẽ, có nhiều ứng dụng trong thực tiễn. Xử lí ảnh và đồ họa vì thế cũng có vai trò quan trọng đối với lĩnh vực tương tác người máy, máy học (Machine Learning) và trí tuệ nhân tạo (Artificial Intelligence).

Về cơ bản, quá trình xử lí ảnh xem như là thao tác xử lí ảnh đầu vào sao cho ra kết quả mong muốn, kết quả ở đây có thể là một ảnh “tốt hơn” hoặc là những kết luận (dựa trên những so sánh thực nghiệm).

***Hình 4.1.1. Quá trình xử lí ảnh***

Kết luận

XỬ LÍ ẢNH

Ảnh “tốt hơn”

Ảnh

Ảnh có thể được xem là tập hợp của những điểm ảnh (pixels), mỗi điểm ảnh được xem là đặc trưng cường độ sáng hoặc dấu hiệu nào đó tại một vị trí nào đó của đối tượng trong không gian, nó có thể xem như hàm n đến P(c1, c2, c3, … cn). Vì thế, ảnh trong quá trình xử lí ảnh có thể xem như ảnh n chiều.

Khung cảnh,   
Đối tượng   
Ảnh quang học

Miêu tả ảnh,

Đối tượng ảnh

Đặc trưng ảnh

Dữ liệu ảnh, Ảnh số

Tín hiệu ảnh, Ảnh liên tục

Xử lí phân tích ảnh

Số hóa ảnh

Thiết bị cảm biến (Camera, Sensor, Scanner)

Nén, lưu trữ và truyền ảnh

***Hình 4.1.2 Các bước cơ bản trong một qui trình xử lí ảnh***

Hình 4.1.2 biểu diễn sơ đồ tổng quát của một qui trình xử lí ảnh. Về cơ bản chúng có các tiến trình chính sau:

* **Thu nhận**: Bao gồm các thiết bị thu nhận Camera, Sensor ảnh,.. có nhiệm vụ tiếp nhận ảnh ngõ vào.
* **Tiền xử lí**: Khối này có nhiệm vụ tiền xử lí các tác vụ với ảnh nhằm nâng cao chất lượng ảnh như giảm nhiễu, tìm biên ảnh, phần vùng màu,…
* **Trích chọn đặc điểm**: Khối này có nhiệm vụ trích chọn các đặc trưng của ảnh đã được tiền xử lí để sử dụng trong Hệ quyết định,
* **Khối hậu xử lí**: Xử lí các đặc điểm đã trích chọn: đường biên ảnh, tọa độ tâm ảnh, mức xám,…. Các đặc trưng này có thể được lược bớt hoặc chuyển đổi để phù hợp hơn với các kĩ thuật lựa chọn cụ thể trong Hệ quyết định.
* **Khối hệ quyết định và lưu trữ**: Khối này có nhiệm vụ đưa ra những lựa chọn phù hợp với yêu cầu xử lí trên lượng dữ liệu lấy được trong khối lưu trữ. Khối lưu trữ có thể là một vùng nhớ trên RAM, ROM hoặc là một ổ cứng, bộ nhớ nhất định.
* **Khối kết luận**: Đưa ra quyết định dựa vào quyết định của khối quyết định.

Trên thực tế, để có thể thu gọn và hệ thống hóa các thao tác xử lí ảnh cũng như cách để tận dụng sức mạnh của phần cứng: máy tính cá nhân , máy tính nhúng, board nhúng,… thì cần có những thuật toán xử lí chi tiết. Các thuật toán này có thể đơn thuần là những thao tác như mở ảnh, mở camera,…hay thậm chí như chuyển màu ảnh từ màu RGB sang màu đơn sắc, khữ nhiễu ảnh răng cưa,….

Thêm vào đó, việc trình bày rõ ràng và giải thích chi tiết các đặc điểm xử lí cũng đòi hỏi khả năng tư duy toán học và mô hình hóa hệ thống cao. Chúng em đưa ra 2 giải pháp để nghiên cứu về giải thuật : tự viết giải thuật hoặc sử dụng các bộ thư viện có sẵn. Giải pháp tự viết giải thuật đem đến khả năng tự chủ về tư duy thiết kế hệ thống, khả năng linh hoạt, gọn nhẹ trong các dự án riêng biệt tuy nhiên đòi hỏi tư duy toán học và mô hình hóa cao. Giải pháp sử dụng các bộ thư viện mã nguồn có sẵn sẽ giúp giảm thiểu thời gian nghiên cứu giải thuật, chuẩn hóa các quy trình bởi các kĩ sư phần mềm trên cộng đồng, tuy nhiên nhược điểm cố hữu là các bộ thư viện luôn bao hàm nhiêu tác vụ, dung lượng nặng và đòi hỏi nhiều về tài nguyên phần cứng khi xử lí, do đó đem đến tốc độ xử lí chậm hơn nếu phần cứng không đủ tốt.

Trên cơ sở tìm hiểu thực tế và nhận định rõ tính chất phức tạp của quá trình xử lí ảnh, chúng em nhận thấy bộ thư viện xử lí ảnh mã nguồn mở OpenCV sẽ là một trợ thủ đắc lực trong các dự án đòi hỏi quá trình xử lí ảnh số, bộ thư viện này cũng được tin dùng và sử dụng trong các dự án số của Google, Microsoft, Intel, Honda, Toyota, Linux Foundation, Raspberry Foundation,… Do đó, việc lựa chọn các bộ thư viện mã nguồn có sẵn sẽ là cách tiếp cận hợp lí và khả thi hơn.

**4.1.2 Bộ thư viện xử lí ảnh mã nguồn mở OpenCV**

**4.1.2.1 Giới thiệu**

OpenCV (Open Source Computer Vision Library) là một thư viện phần mềm máy tính và thị giac máy tính mã nguồn mở1. Chúng được xây dựng để cung cấp một cơ sở hạ tầng chung cho các ứng dụng thị giác máy tính nhằm tăng tốc việc sử dụng nhận thức máy móc trong các sản phẩm thương mại. OpenCV là một sản phẩm được cấp phép BSD (Berkelay Software Distribution - giấy phép mã nguồn mở) và GPL (General Public License), giúp các doanh nghiệp dễ dàng sử dụng và sửa đổi mã nguồn. OpenCV có trang chủ chính thức tại *https://opencv.org/*, cho phép tải tài liệu và mã nguồn tự do theo nguyên tắc cộng đồng của BSD và GPL.

Hiện nay, OpenCV bao gồm hơn 2500 thuật toán được tối ưu hóa, trong đó bao gồm một bộ toàn diện của cả thuật toán máy học và thị giác máy tính cổ điển cũng như hiện đại. Các thuật toán này có thể được sử dụng để phát hiện và nhận dạng khuôn mặt, theo dõi chuyển động của máy ảnh kĩ thuật số, theo dõi đối tượng động, trích xuất mô hình 3D của đối tượng, tạo ra các đảm mấy điểm ảnh từ máy ảnh nổi, ghép các hình ảnh lại với nhau để tạo ra hình ảnh có độ phân giải tốt hơn hoặc tìm ra hình ảnh toàn bộ, loại bỏ hiện tượng mắt đỏ bởi đèn flash, theo dõi chuyển động của mắt người, nhận dạng phong cảnh và thiết lập các điểm đánh dấu để phủ nó bằng thực tế tăng cường,…OpenCV có hơn 47 nghìn người dùng cộng đồng, lượng kĩ sư và cộng đồng phát triển cũng tăng lên không ngừng nghỉ. Bộ thư viện này đã tung ra phiên bản mới nhất là OpenCV 4.5 và OpenCV Contrib 4.5, ước tính hiện tại có hơn 18 triệu lượt tải về tính trên tất cả phiên bản của mã nguồn. OpenCV cũng là một trong số những bộ thư viện an toàn để sử dụng rộng rãi trong các công ty, nhóm nghiên cứu và các cơ quan chính phủ.

Với số lượng thuật toán tối ưu lên đến hàng nghìn đơn vị, OpenCV cũng được xem là một thư viện vỡ lòng và rất đáng để nghiên cứu cho các sinh viên, nghiên cứu sinh, chuyên gia,.. trong lĩnh vực thị giác máy tính và trí tuệ nhân tạo.

Trong phạm vi giới hạn của đề tài, chúng em cũng đã vận dụng và sử dụng bộ thư viện mã nguồn mở để giải quyết các bước tính toán và rút ngắn thời gian, độ khó trong suốt quá trình xử lí ảnh số, có thể nói, nếu không có sự hỗ trợ từ bộ thư viện OpenCV, chúng em sẽ rất khó tiếp cận và hoàn thành được yêu cầu đặt ra của đề tài.

**4.1.2.2 Cài đặt OpenCV trên hệ điều hành GNU/Linux**

OpenCV đều có thể được cài đặt trên rất nhiều nền tảng hệ điều hành (OS – Operation System) khác nhau như Windows, MacOS, các bản phân phối GNU/Linux,.. tuy nhiên để thuận tiện và thống nhất cho ứng dụng, chúng em chọn cách cài đặt OpenCV trên hệ điều hành GNU/Linux, cụ thể là bản phân phối Ubuntu Mate trên board máy tính nhúng Raspberry Pi và máy tính cá nhân để gỡ lỗi.

**a. Cài đặt môi trường và các gói cài cần thiết**

- Mở terminal, tiến hành cài các gói cài sau:

***$ sudo apt install build-essential cmake git pkg-config libgtk-3-dev \ libavcodec-dev libavformat-dev libswscale-dev libv4l-dev \ libxvidcore-dev libx264-dev libjpeg-dev libpng-dev libtiff-dev \ gfortran openexr libatlas-base-dev python3-dev python3-numpy \ libtbb2 libtbb-dev libdc1394-22-dev***

**b. Clone OpenCV và kho đóng góp OpenCV**

- Tạo ra folder **opencv\_build** và đến đường dẫn đó.

***$ mkdir opecv\_build && cd ~/opencv\_build***

Đối với tôi khi làm mình sẽ tạo ra đường dẫn như sau: **/home/covv/workspace/src/opencv\_build**

- Có hai cách lấy gói cài về máy tính: Clone trực tiếp từ github hoặc tải trực tiếp từ trang chủ OpenCV.

+ Clone trực tiếp:

***$ git clone https://github.com/opencv/opencv.git***

***$ git clone https://github.com/opencv/opencv\_contrib.git***

nếu bạn muốn cài các phiên bản cũ hơn của opencv và opencv\_contrib, bạn có thể dùng lệnh:

***$ git checkout <opencv-version>***

+ Tải trực tiếp từ trang chủ:

Truy cập [**https://opencv.org/releases**](https://opencv.org/releases/)và tải về các gói cài opencv.

Truy cập **https://github.com/opencv/opencv\_contrib** để tải về gói cài opencv\_contrib.

Mình hay dùng bản opencv 3.2 và opencv\_contrib 3.2, tuy hơi cũ nhưng bù lại vì độ ổn định và do quen thuộc với các chương trình của mình ^.^

**c. Sau khi tải về, tạo ra một folder build tạm và chuyển đến folder đó.**

- Copy hoặc di chuyển 2 gói cài opencv và opencv\_contrib vừa tải về vào thư mục opencv\_build vừa tạo ở bước 2.

- Thực hiện dòng lệnh:

***$ cd ~/opencv\_build/opencv***

***$ mkdir build && cd build***

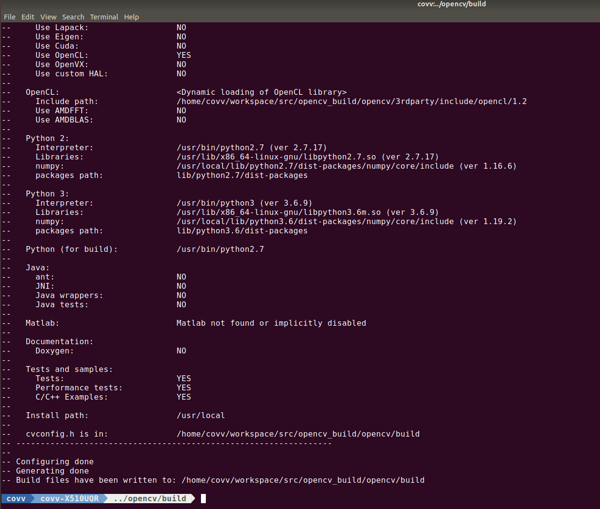
- Setup môi trường build OpenCV với CMake:

**$ cmake -D CMAKE\_BUILD\_TYPE=RELEASE \  
-D CMAKE\_INSTALL\_PREFIX=/usr/local \  
-D INSTALL\_C\_EXAMPLES=ON \  
-D INSTALL\_PYTHON\_EXAMPLES=ON \  
-D OPENCV\_GENERATE\_PKGCONFIG=ON \  
-D OPENCV\_EXTRA\_MODULES\_PATH=~/opencv\_build/opencv\_contrib/modules \  
-D BUILD\_EXAMPLES=ON ..**

Các bạn chú ý đến OPENCV\_EXTRA\_MODULES\_PATH phải là đường dẫn chính xác module của bạn.

Đối với máy tính của mình thì đường dẫn là: **/workspace/src/opencv\_build/opencv\_contrib/modules \**

Sau khi setup môi trường thành công, màn hình terminal thông thường sẽ hiển thị như sau:



**d. Tiến hành biên dịch**

Gõ dòng lệnh:

***$ make -j4*** (có thể là make -j8, -j16,..) tùy thuộc vào số thread CPU máy tính của bạn, số thread càng cao thì biên dịch càng nhanh.

Gõ: ***$ sudo nproc*** để biết được số thread.

Đoạn này các bạn có thể chuyển máy làm việc khác hoặc uống một tách cà phê vì sẽ mất thời gian đấy.

**đ. Tải xuống OpenCV**

**$ sudo make install**

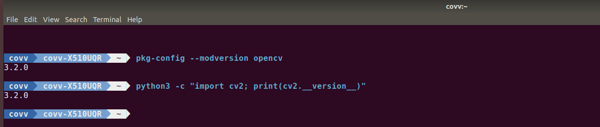
**f. Kiểm tra version đã được cài thành công trên máy tính.**

***$ pkg-config --modversion opencv***

**3.2.0 -->**Đây là bản mình cài cho máy tính của mình

***$ python3 -c "import cv2; print(cv2.\_\_version\_\_)"***

**3.2.0 -->** Đây là bản mình cài cho máy tính của mình



Như vậy về cơ bản chúng ta đã có thể sử dụng bộ thư viện trong các chương trình cơ bản trên máy tính như C/C++, Python, MATLAB mà không bị lỗi.

**4.2 Tiến trình xử lí hình ảnh và điều khiển**

**4.2.1 Quá trình xử lí hình ảnh**

Xét về số lượng hàm mục tiêu hỗ trợ trên OpenCV, chúng ta có hơn 2500 thuật toán đã đề cập, do đó đối với những nhóm phát triển khác nhau sẽ có những cách thức sử dụng và vận dụng khác nhau đối với từng yêu cầu cụ thể cho việc xử lí hình ảnh. Thêm vào đó, tùy vào chất lượng và sức mạnh xử lí của phần cứng, chúng ta nên sử dụng các phương pháp ứng dụng khác nhau.

Đối với phạm vi đề cập của đề tài, chúng em sử dụng các thuật toán cơ bản của OpenCV trong ứng dụng cụ thể xử lí ảnh và bám mục tiêu cố định của nhóm. Thuật toán nhận dạng mục tiêu là bước đầu tiên và khá quan trọng trong hệ robot tự động nhận dạng mục tiêu. Mục tiêu của quá trình là thực hiện các phép xử lí ảnh để tìm đối tượng cần quan tâm, xác định vị trí của đối tượng trong mặt phẳng ảnh hiện thời. Trong quá trình tìm kiếm chúng ta chia tập hợp các pixel của ảnh làm hai thành phần: Hoặc là của đối tượng hoặc là thuộc nền. Sau đây ta sẽ xem xét việc tìm kiếm các pixel thuộc đối tượng trong chuỗi ảnh thu về.

Có rất nhiều cách để nhận dạng một đối tượng: khi biết trước chính xác đặc điểm của đối tượng cần nhận dạng ta có thể áp đặt toàn bộ mô hình của đối tượng (màu sắc: xanh, đỏ, vàng,…; hình dạng: vuông, tròn, hình chữ nhật,..) lên toàn bộ ảnh, từ đó tìm ra vị trí của đối tượng cần nhận dạng trong ảnh. Phương pháp này có ưu điểm đơn giản, dễ thực hiện song lại không hiệu quả trong thực tế vì đòi hỏi khả năng xử lí và tiên đoán của cả phần mềm lẫn phần cứng cực kì cao.

Một phương pháp nhanh hơn, hiệu quả hơn đối với ảnh đen trắng là phương pháp so sánh. Phương pháp này dựa trên đặc điểm các điểm ảnh (pixel) thuộc nền thì cường độ sáng ít thay đổi hoặc nếu thay đổi thì rất nhỏ còn các pixel thuộc đối tượng cường độ sáng thay đổi lớn. Như vậy một phép so sánh giữa các pixel trong ảnh hiện thời với ảnh trước đó được thực hiện. Nếu camera tĩnh, sự thay đổi ánh sáng của môi trường rất nhỏ và nhiễu không đáng kể thì một pixel thuộc nền nếu độ sai khác bằng 0 và ngược lại sẽ thuộc đối tượng. Tuy nhiên trong thực tế các điều kiện trên khó có thể xảy ra, do đó để loại bỏ các điểm mà chúng ta phát hiện sai do nhiễu gây ra ta có thể dùng ngưỡng thích hợp. Nếu độ sai khác lớn hơn ngưỡng thì cho ra giá trị bằng 1, ngược lại cho ra giá trị 0, các giá trị luận lí này cũng giúp hệ vi mạch xử lí tín hiệu nhanh hơn. Nếu ngưỡng quá lớn, pixel thuộc đối tượng có thể lẫn với nền, nếu ngưỡng quá nhỏ, thì sự thay đổi ánh sáng trong môi trường sẽ tạo ra nhiều pixel có giá trị 1, mà các điểm này lại không phải là điểm thuộc đối tượng. Sau khi xử lý ngưỡng ta thu được ảnh nhị phân chỉ gồm các pixel với hai trạng thái 0 và 1. Quá trình phân đoạn ảnh thành từng đối tượng riêng rẽ được thực hiện thông qua thuật toán đánh nhãn liên tiếp. Quá trình này sẽ tạo ra vô số đối tượng khác nhau, trong đó chỉ có một số đối tượng gần với đối tượng cần quan tâm. Một quá trình quét tiếp theo sẽ được thực hiện để loại bỏ các đối tượng không hợp lý so với đối tượng mẫu. Việc còn lại là xác định vị trí trọng tâm của mục tiêu trên ảnh tương đối đơn giản, thông qua xác định momen bậc nhất.

Phương pháp chính mà nhóm chúng em sử dụng trong đề tài là thuật toán Camshift. Thuật toán này có hiệu quả rõ rệt khi thu ảnh từ camera là ảnh màu và đối tượng cần phát hiện có màu sắc đặc trưng riêng so với nền.

Hơn thế, đối với yêu cầu thực tế tại nhà màng, lượng ánh sáng và vật mục tiêu hoàn toàn được khống chế sao cho đối tượng mục tiêu có đặc trưng rõ rệt so với những mục tiêu không cần thiết (tạm gọi là nhiễu hình ảnh không mong muốn), phần này sẽ được trình bày rõ hơn ở chương 6 của báo cáo.

Phương pháp chính được giới thiệu trong bài báo này là thuật toán Camshift. Thuật toán này đặc biệt hiệu quả với ảnh thu được từ camera là ảnh màu và đối tượng cần phát hiện và bám có màu sắc đặc trưng riêng so với nền.

**4.2.2 Giải thuật CamShift**

**4.2.2.a Cơ sở toán học của thuật toán CamShift**

Camshift (Continously Adaptive Meanshift ) được phát triển từ thuật toán Meanshift, nó bao gồm thuật toán cơ sở Meanshift với thích ứng các bước thay đổi kích cỡ của vùng. Nhân là một hàm step đơn giản nằm trong một bản đồ vỏ xác suất (skin-probability). Vỏ xác suất của mỗi một điểm ảnh được tính toán dựa trên màu sắc bằng cách sử dụng một phương pháp gọi là Histogram backprojection. Màu sắc được đưa ra như Hue từ mô hình màu HSV. Kể từ nhân là một hàm step, Mean Shift ở mỗi một lần lặp thì chỉ đơn giản là giá trị trung bình x và y của vỏ xác suất đóng góp trong vùng hiện tại. Điều này được xác định bằng cách chia moment thứ nhất của vùng này cho moment thứ không ở mỗi lần lặp và dịch chuyển vùng đến trọng tâm xác suất. Sau khi Mean Shift hội tụ đến một vị trí (x,y), phạm vi của vùng được cập nhật dựa trên giá trị hiện tại của moment thứ 0. Giá trị của bản cập nhật này có thể thay đổi, phạm vi tuyến tính của nó được giả định là tỉ lệ thuận với căn bậc 2 của tổng của các khoản đóng góp xác suất của khu vực hiện tại (ví dụ. Moment thứ không). Chiều rộng và chiều dài tại thời điểm thứ i sẽ được tính thông qua giá trị tại thời điểm thứ i-1, nghĩa là giá trị trước đó.

Trong thuật toán CamShift, một bức ảnh về xác suất phân bố màu sắc của bức ảnh trong chuỗi video được tạo ra. Đầu tiên nó tạo ra một mẫu mô tả hue sử dụng một biểu đồ màu sắc (color Histogram) và sử dụng không gian màu Hue Saturation Value (HSV) được tham chiếu từ không gian màu tiêu chuẩn RGB. Vì phân bố màu sắc của các bức ảnh trong video thay đổi theo thời gian, nên thuật toán CamShift đã được sửa đổi để dễ dàng thích ứng với sự thay đổi phân bố xác suất màu sắc của mục tiêu nó theo dõi.

**4.2.2.b. Các bước thực thi thuật toán CamShift:**

B1: Chọn vị trí ban đầu của cửa sổ tìm kiếm (frame camera)

B2: Thuật toán Meanshift (lặp một hoặc nhiều lần), lưu moment thứ 0.

B3: Kích thước của cửa sổ tìm kiếm được đặt bằng một hàm của moment thứ 0 vừa được tìm trong bước 2.

B4: Lặp lại bước 2 và bước 3 cho đến khi hội tụ (nghĩa là vị trí dịch chuyển nhỏ hơn ngưỡng thiết lập trước đó).

**4.2.2.c Lưu đồ thuật toán Camshift**

Đầu tiên là một Histogram được tạo ra, Histogram này chứa các thuộc tính liên quan đến màu sắc. Sau khi tạo ra histogram, tiếp theo tâm và kích cỡ của mục tiêu được tính toán để theo dõi mục tiêu khi hình dạng và kích cỡ của nó thay đổi. Tính xác suất phân bố mục tiêu căn cứ vào Histogram nhận được. Dịch chuyển đến vị trí mới với mỗi khung hình vừa nhận được từ video. Camshift sẽ dịch chuyển đến vị trí mà nó ước lượng trên đối tượng tập trung nhiều điểm sáng nhất trong bức ảnh xác suất. Tìm vị trí mới bắt đầu từ vị trí trước đó và tính toán giá trị trọng tâm vừa tìm được. Cả hai thuật toán trên đều có thư viện các hàm tính toán trong OpenCV.

Đối với các khâu xử lí tính toán trong báo cáo, chúng em sử dụng thao tác dịch chỉnh Histogram (dịch chỉnh H, S, V) để thực hiện lấy mẫu thử nhiều lần cho vật chuẩn là màn hình LED sáng từ điện thoại. Quá trình thử và lấy mẫu dựa trên dịch chỉnh các hệ số nhiều lần, phần này sẽ được trình bày cụ thể trong chương 6.

Chương trình xử lý ảnh viết trên Qt Creator có sử dụng thư viện OpenCV của Intel được biên dịch chéo trên máy tính Linux sau đó được truyền xuống board nhúng chạy độc lập là Raspberry Pi B+.

CamShift

Xác suất phân bố màu

Tính toán Histogram

Chuyển đổi không gian màu

X, Y

Thiết lập lại vùng tính toán

Thiết lập tâm và kích thước cửa sổ hiển thị ảnh

**4.2.3 Lưu đồ giải thuật của hệ thống**

(Lưu đồ)

**4.3 Phần cứng bộ xử lí**

**4.3.1 Máy tính nhúng Raspberry Pi B+ và camera Pi NoIR**

**4.3.1.a Máy tính nhúng Raspberry Pi 3 Model B+**

Raspberry Pi là máy tính nhúng gái rẻ chỉ từ 35USD, có kích cỡ siêu nhỏ gọn và chạy hệ điều hành Linux. Mục tiêu chính của Raspberry là giảng dạy máy tính cho trẻ em. Raspberry được phát triển bởi Raspberry Pi Foundation – một tổ chức phi lợi nhuận với tiêu chí xầy dựng hệ thống mở cho người dùng với nhiều tùy biến khác nhau.

Raspberry hiện tại được sản xuất bởi 3 OEM: Sony, Qsida, Egoman, được phân phối chính hãng bởi Element14, RS Componets và Egoman.

Raspberry Pi là một máy tính rẻ tiền có khả năng lập trình được dành cho học sinh, sinh viên nhưng hiện tại được sử dụng bởi nhiều đối tượng khác nhau với nhiều mục đích khác nhau.

Tuy nhiên, Raspberry không thể thay thế hoàn toàn cho hệ thống máy tính để bàn hoặc máy tính xách tay. Chúng cũng không hỗ trợ Windows mà chỉ hỗ trợ các bản phân phối của GNU/Linux. Raspberry tuy có giá thành rẻ nhưng hoàn hảo cho những hệ thống điện tử, DIY (Do It Yourseft),..

Nhóm em sử dụng phiên bản Raspberry Pi 3 Model B+ cho mục đích xử lí tính toán hình ảnh thu thập được từ camera và truyền về board nhúng theo như sơ đồ.

Thông số kĩ thuật:

* Vi xử lí CPU Broadcom BCM2837B0 4 nhân A53 (ARMv8) 64-bit xung nhịp 1.4GHz
* Chip xử lí đồ họa GPU Broadcom Videocore-IV
* Bộ nhớ RAM 1GB LPDDR2 SDRAM
* Giao tiếp mạng: Gigabit Ethernet (thông qua cổng USB), hỗ trợ WiFi 802.11b/g/n/ac băng tần 2.4GHz và 5GHz
* Tích hợp giao tiếp Bluetooth 4.2, Bluetooth Low Energy (BLE)
* Bộ nhớ: Micro-SD
* Giao tiếp ngoại vi GPIO: 40 chân cắm header GPIO
* Cổng kết nối: HDMI, jack âm thanh analog 3.5, 4x USB 2.0, Ethernet, Camera Serial Interface (CSI), Display Serial Interface (DSI)
* Kích thước: 82mm x 56mm x 19.5mm, 50g

Với tác vụ tính toán nghiêng về đồ họa nặng cho thư viện OpenCV như hiện nay, có thể nâng cấp lên các máy tính nhúng cao cấp hơn như Raspberry Pi 4, NVidia Jetson Nano Developer Kit,….để tăng chất lượng xử lí hình ảnh.

**4.3.1.b Camera Pi NoIR V2 8MP**

Đây là camera được nhóm sử dụng để làm camera chính trong việc thu thập hình ảnh từ mục tiêu về độ ổn định cũng như được OMD chính hãng từ đối tác của Raspberry Foundation.

Camera Raspberry Pi NoIR V2 8MP là phiên bản Camera Module dành cho Raspberry Pi mới nhất sử dụng cảm biến ảnh IMX219 8-megapixel từ Sony thay cho cảm biến cũ là OV5647. Với cảm biến IMX219 8-megapixel từ Sony, Camera Module cho Raspberry Pi đã có được sự nâng cấp vượt trội về cả chất lượng hình ảnh, video cũng như độ bền.

Camera Raspberry Pi NoIR V2 8MP có thể sử dụng với Raspberry Pi để chụp hình, quay phim trong điều kiện thiếu sáng với chất lượng HD 1080p30, 720p60 hoặc VGA90, cách sử dụng cũng như lập trình với Camera Module trên Raspberry Pi cũng rất dễ dàng, chỉ cần cắm vào cổng Camera CSI trên Raspberry Pi và Config 1 chút là có thể dùng được.

Camera Raspberry Pi NoIR V2 8MP có thể sử dụng tương thích với tất cả các dòng Raspberry Pi 1, 2 và 3 từ trước đến nay, có thể điều khển thông qua MMAL và V4L APIs, có rất nhiều bộ thư viện được cộng đồng Raspberry Pi phát triển trên C/C++, Python giúp cho việc tìm hiểu và sử dụng trở nên dễ dàng hơn rất nhiều.

Thông số kĩ thuật:

* Camera Raspberry Pi NoIR V2 8MP dùng cho máy tính nhúng Raspberry Pi trong các điều kiện thiếu sáng.
* Cảm biến IMX219 Sony.
* Số điểm ảnh: 8MP.
* Lens: Fixed focus
* Độ phân giải camera: 3280 x 2464 pixel
* Độ phân giải video quay được: HD 1080p30, 720p60 and 640x480p90
* Kích thước: 25mm x 23mm x 9mm
* Khối lượng: ~3g
* Loại chân kết nối: Chân kết nối Ribbon
* Giao tiếp: CSI

Thông số sản phẩm chi tiết có thể tham khảo tại: https://www.raspberrypi.org/products/pi-noir-camera-v2/?resellerType=home

**4.3.2 Board nhúng Arduino Mega 2560**

Board nhúng Arduino Mega là board phát triển nhúng đơn giản, phù hợp cho những người mới, học sinh, sinh viên nghiên cứu các sản phẩm ứng dụng cho điện tử, công nghệ máy tính theo hướng đơn giản hóa câu lệnh và phương thức sử dụng. Board chính hãng của Arduino.cc được sản xuất và phân phối từ Italia. Arduino Mega chính hãng có giá thình tuy không rẻ nhưng có độ bền bỉ và khả năng ứng dụng tốt, do đó được rất nhiều người dùng ưa thích. Địa chỉ webstite trang chủ của nhà sản xuất: https://www.arduino.cc/

Thông số kĩ thuật:

* Vi điều khiển chính: ATmega2560
* IC nạp và giao tiếp UART: ATmega16U2.
* Nguồn nuôi mạch: 5VDC từ cổng USB hoặc nguồn ngoài cắm từ giắc tròn DC (nếu sử dụng nguồn ngoài từ giắc tròn DC.
* Số chân Digital I/O: 54 (trong đó 15 chân có khả năng xuất xung PWM)
* Số chân Analog Input: 16
* Dòng điện DC Current trên mỗi chân I/O: 20mA
* Dòng điện DC Current chân 3.3V: 50mA
* Flash Memory: 256 KB trong đó 8 KB sử dụng cho bootloader.
* SRAM: 8 KB
* EEPROM: 4 KB
* Clock Speed: 16 MHz
* LED\_BUILTIN: 13
* Kích thước: 101.52 x 53.3 mm

Với mức ứng dụng của đề tài, chúng em có thể sử dụng các dòng board nhúng cao cấp với các MCU mạnh mẽ hơn như STM32F4 Discovery Board (STMicroelectronics), Tiva Launchpad TI Board (Texas Instruments),…. Tuy nhiên, với độ khó của toàn bộ đề tài, lựa chọn sử dụng board Arduino Mega giúp giảm thiểu gánh nặng về lập trình, tập trung tốt hơn cho giải thuật điều khiển.

**4.3.3 Mạch cầu H SmartPID**

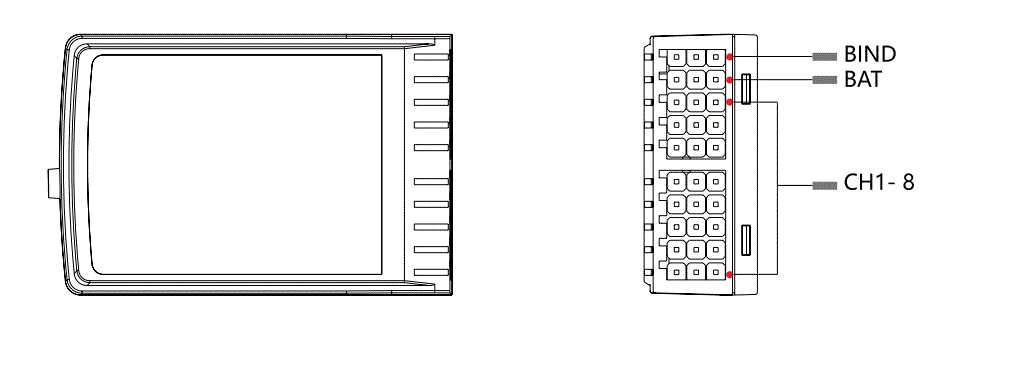
**4.3.4 Bộ tay cầm điều khiển từ xa FlySky FS-TH9X**

Tay cầm điều khiển từ xa FlySky FS-TH9X được phát triển cho những người có sở thích và đam về điều khiển vô tuyến radio. Bộ điều khiển bao gồm tay cầm (Handler) và đầu thu (Receiver). Sản phẩm có thể được ứng dụng điều khiển hầu hết dạng robot, xe địa hình, máy bay không người lái, thuyền, cano không người lái.

FS-TH9X có 8 kênh mặc định, tuy nhiên có thể nâng cấp lên 14 kênh bằng mô đun RF (Radio Frequency). Hệ thống này được phát triển đặc biệt cho tất cả các mô hình điều khiển Radio, cung cấp khả năng chống nhiễu siêu chủ động và thụ động, tiêu thụ điện năng rất thấp và độ nhạy thu cao. Với sự kiểm tra nghiêm ngặt của các kỹ sư và nghiên cứu thị trường trong nhiều năm, FLYSKY hiện được coi là một trong những hệ thống tốt nhất hiện có trên thị trường. Địa chỉ trang chủ của nhà sản xuất: https://www.flysky-cn.com/

* Sản phẩm hỗ trợ chế độ nạp mã nguồn mở, cho phép các lập trình viên nhúng tự do phát triển theo phần cứng đã cung cấp.
* Tần số sóng không dây: 2.4GHz
* Năng lượng truyền dẫn: < 20dBm
* Khoảng cách thu phát: 500 ~ 1500m (trong không khí)
* Chuẩn giao tiếp dữ liệu: Phone Jack (PPM)
* Module RF cũng có thể tháo rời hoặc thay thế một cách dễ dàng.
* Hỗ trợ tối đa 14 kênh RF
* Hỗ trợ mô đun RF AFHDS/AFHDS 2A
* Sử dụng 8 pin AA cũng cấp nguồn
* Tiêu chuẩn đáp ứng: CE, RCM, FCC ID

Trong đề tài, chúng em sử dung 4 kênh tín hiệu của tay cầm đề điều khiển các động cơ và bộ phận sau: Động cơ tải chính, Động cơ bẻ lái, Động cơ nâng hạ dàn phun và động cơ máy bơm.



Relay 5V

Động cơ máy bơm thuốc

Driver mạch cầu BTS7960

Động cơ dẫn hướng DC servo

Driver mạch cầu BTS7960

Drive TB6560

Động cơ Step nâng hạ dàn phun

Động cơ DC tải chính

---------------------------------------------------------

**4.3.5 Các board mạch và phần cứng khác**: Ngoài các board mạch chính sử dụng trong bộ phận xử lí, chúng em cũng sử dụng các board mạch driver như TB6560, mạch cầu H BTS7960, mạch giảm áp 24V – 5V DC, acquy,…

**4.4 Sơ đồ kết nối thiết bị**

Camera Pi NoIR

Raspberry Pi 3 B+

Relay 5V – 24V

Driver mạch cầu H BTS7960

Động cơ bước Nema 23

Driver động cơ bước TB6560

Arduino Mega 2560

Động cơ máy bơm

Động cơ DC giảm tốc

Receiver FS-TH9X

Driver mạch cầu H BTS7960