

ỨNG DỤNG XỬ LÝ ẢNH NHẬN DẠNG ĐƯỜNG ĐI CHO Ô TÔ CHẠY TỰ ĐỘNG

APPLICATION OF IMAGE PROCESSING TO DETECT LANE FOR AUTONOMOUS VEHICLE

Hồ Văn Thu¹, Lê Thanh Phúc²

¹Trường Cao Đẳng Kỹ Thuật Cao Thắng,

²Trường Đại Học Sư Phạm Kỹ Thuật TP.HCM

Ngày tòa soạn nhận bài 30/12/2014, ngày phản biện đánh giá 6/3/2015, ngày chấp nhận đăng 10/3/2015

TÓM TẮT

Nhận dạng đường đi cho ô tô chạy tự động là một đề tài hỗ trợ cho hệ thống giao thông thông minh. Nghiên cứu này sử dụng phần mềm Matlab và các Toolbox của nó là: Image Processing, Image Acquisition System, Computer Vision System làm công cụ chính để thu thập và xử lý. Trên mô hình đường là đường nhựa với các giả định: vân của đường là đồng nhất, dấu phân cách làn đường tuân theo quy chuẩn của vạch kẻ đường, khoảng cách giữa các dấu phân cách là không đổi. Đề tài nghiên cứu sử dụng webcam làm bộ phận thu thập hình ảnh chính. Từ hình ảnh thu được sử dụng bộ lọc FIR 2D để lọc ảnh và biến đổi ảnh thành ảnh nhị phân, dùng biến đổi Hough để xác định các dấu phân cách đường, làn đường. Tính toán khoảng cách dựa trên các dấu phân cách thu được để đưa ra tín hiệu về đường đi cho phần điều khiển xe tự động thông qua chuẩn giao tiếp RS232. Đề tài đã xây dựng được một chương trình nhận dạng đường đi và đã thực thi có hiệu quả trên các đoạn đường thử nghiệm.

Từ khóa: Hough Transform, điều khiển, làn đường, MATLAB, nhận dạng, xe tự hành.

ABSTRACT

Image processing has many applications in intelligent transportation system. Lane detection and tracking for autonomous vehicle is the one of that. This study uses Matlab software and its Toolbox: Image Processing, Image Acquisition System and Computer Vision System as the main tool to collect and process. The road model is assumed that: texture of road are identical. Lane makers follow lane rules. The distance between the lane makers is constant. This project uses webcam to collect the images. From the images obtained by using the 2D FIR filter, grayscale image is achieved. The image are converted into binary images, then by using Canny method and Hough transform available in Matlab to determine the lane makers and lane department. Calculate the distance based on separator obtained to provide signal for autonomous vehicle via the standard RS232 interface. The experimental results on local streets show that the suggested program is very reliable.

Key words: Hough Transform, control, lane, MATLAB, detection, autonomous car.

I. GIỚI THIỆU

Ngày nay, chúng tôi nhận thấy rằng các nhà nghiên cứu tập trung vào việc gia tăng tính an toàn cho ô tô để giảm thiểu tối đa nguy cơ gây tai nạn giao thông. Đã có nhiều nghiên cứu về hệ thống hỗ trợ người lái về làn đường như cảnh báo lệch làn đường, hỗ trợ đỗ xe, cảnh

báo và chạm... dựa trên các hệ thống cảm biến được gắn trên ô tô. Trong đó, công nghệ xe tự lái được cho là an toàn hơn, đơn giản vì máy móc hoạt động nhanh hơn con người. Số khung hình mỗi giây mà mắt thường chúng ta nắm bắt ít hơn một máy quay tốc độ cao, não

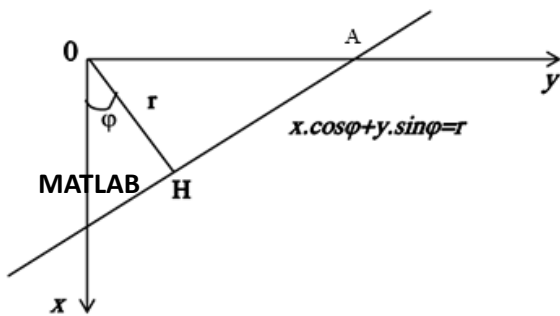
chúng ta xử lý dữ liệu và phản ứng chậm hơn CPU máy tính. Và chân tay con người thua xa về tốc độ làm việc so với những cơ cấu điện tử.

Ở Việt Nam, cơ sở hạ tầng giao thông đang dần được cải thiện. Việc xây dựng một hệ thống giao thông thông minh là vấn đề cấp thiết trong công cuộc công nghiệp hóa, hiện đại hóa đất nước. Nghiên cứu, xây dựng các công cụ hỗ trợ cho hệ thống giao thông thông minh trong đó có phương tiện giao thông thông minh là yêu cầu của thời đại và việc ứng dụng xử lý ảnh để nhận dạng đường đi cho ô tô chạy tự động cũng nhằm mục đích này.

II. PHƯƠNG PHÁP NGHIÊN CỨU

1. Cơ sở lý thuyết

Dùng biến đổi Hough [1] để xác định dấu phân cách làn đường trong ảnh:



Hình 1: Đường thẳng Hough trong tọa độ Đề-các [2]

Mỗi điểm (x,y) trong mặt phẳng được biểu diễn bởi cặp (r, φ) trong tọa độ cực.

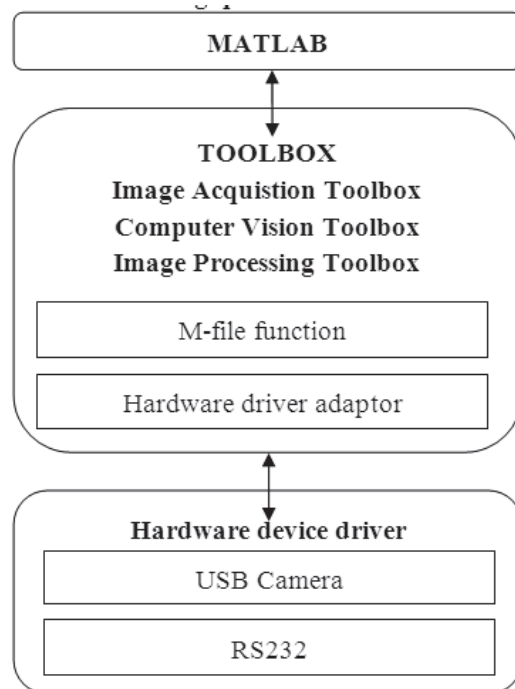
Tương tự mỗi đường thẳng trong mặt phẳng cũng có thể biểu diễn bởi một cặp (r, φ) trong tọa độ cực với r là khoảng cách từ góc tọa độ tới đường thẳng đó và φ là góc tạo bởi trục Ox với đường thẳng vuông góc với nó, hình 1 biểu diễn đường thẳng Hough trong tọa độ Đề-các. Ngược lại, mỗi một cặp (r, φ) trong tọa độ cực cũng tương ứng biểu diễn một đường thẳng trong mặt phẳng.

Giả sử $M(x,y)$ là một điểm thuộc đường thẳng được biểu diễn bởi (r,φ) , gọi $H(x,y)$ là hình chiếu của góc tọa độ O trên đường thẳng ta có $x = r\cos\varphi$, $y = r\sin\varphi$. Mặt khác, ta có: $OH.HA=0$.

Từ đó có mối liên hệ (x,y) và (r,φ) như sau: $x\cos\varphi + y\sin\varphi = r$. Xét n điểm thẳng hàng trong tọa độ Đề-các có phương trình $x\cos\varphi_0 + y\sin\varphi_0 = r_0$. Biến đổi Hough ánh xạ n điểm này thành n đường sin trong tọa độ cực mà các đường này đều đi qua điểm (r,φ) . Giao điểm (r,φ) của n đường sẽ xác định một đường thẳng trong hệ tọa độ Đề-các. Như vậy những đường thẳng đi qua điểm (x,y) sẽ cho duy nhất một cặp (r,φ) và có bao nhiêu đường đi qua (x,y) sẽ có bấy nhiêu cặp giá trị (r,φ) .

Trong bài toán của đề tài, xét các điểm ảnh thẳng hàng nằm trên dấu phân cách làn đường, dùng biến đổi Hough để xây dựng mảng tích lũy các giá trị (r,φ) . Tìm giá trị cực đại trong mảng tích lũy, giá trị cực đại đó tương ứng với dấu phân cách làn đường trong ảnh.

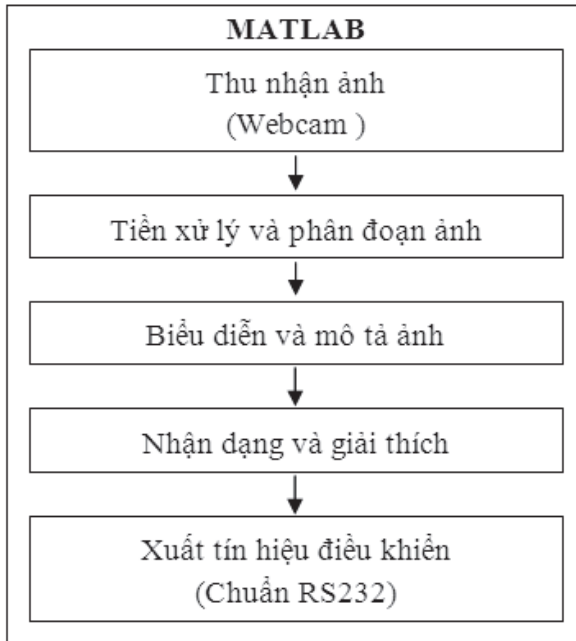
2. Mô hình tổng quát



Hình 2: Mô hình tổng quát [1]

Đề tài sử dụng mô hình xử lý tham khảo như hình 2. Phần mềm Matlab là công cụ xử lý chính, giao tiếp với thiết bị ngoại vi thông qua cổng usb và cổng COM. Trong đó, camera giao tiếp với máy tính bằng cổng usb, truyền tín hiệu đến phần cứng bằng cổng COM.

3. Mô hình của đề tài



Hình 3: Mô hình của đề tài

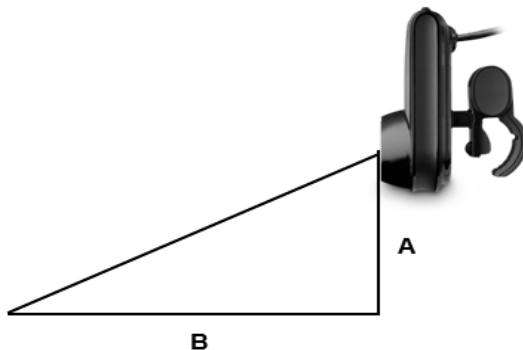
4. Thu nhận ảnh

a. Chọn và thiết lập camera

Đây là bước rất quan trọng bởi nó có ảnh hưởng rất lớn đến độ chính xác trong quá trình xử lý đối với các đề tài ứng dụng xử lý ảnh. Bước này đề cập đến việc thiết lập vị trí ghi hình. Độ phân giải của camera, tốc độ ghi hình.

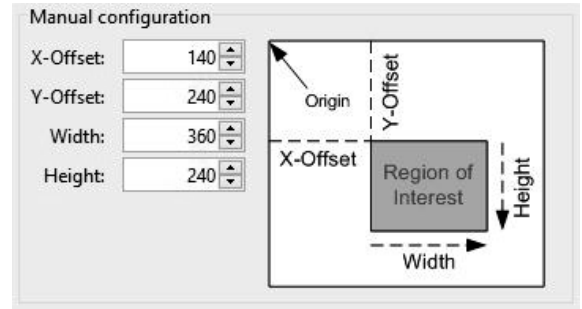
Camera được sử dụng trong đề tài là Logitech webcam 905. Có các thông số:

- Độ phân giải: 2MP (up to 8MP).
- Kích thước khung hình tối đa là: 1600x1200 pixels, ảnh RGB.
- Giao tiếp máy tính: USB.



Hình 4: Thiết lập góc quay

Kích thước khung hình thu thập từ thiết bị này nằm trong khoảng từ 160x90 cho đến 1600x1200. Thông số webcam được chọn trong đề tài là 'RGB24_640x480'. Tốc độ ghi hình là 30fps. Thiết lập góc quay như hình 4, với tỷ lệ



Hình 5: Vùng quan tâm [3]

b. Thu nhận ảnh từ camera

Sử dụng hàm `imaq.VideoDevice` trong thư viện Image Acquisition System Toolbox [3] để thu tập hình ảnh từ webcam. Trong đó vùng quan tâm ROI, với ý nghĩa như hình 5. Sau đó, trích xuất từng khung hình rồi tạo vòng lặp xử lý.

5. Tiền xử lý ảnh



Hình 6: Ảnh đầu vào

Ở bước này, ảnh sẽ được cải thiện về độ tương phản, lọc, xám hóa... với mục đích cho chất lượng ảnh tốt hơn.

Sau đó, chia ảnh đầu vào thành nhiều phần nhỏ và chọn những vùng ảnh cần quan tâm để xử lý. Ví dụ: Ảnh đầu vào như hình 6 thì vùng cần quan tâm chính là nửa dưới của bức ảnh,

nơi có chứa dấu chỉ làn đường.

Đề tài đã sử dụng bộ công cụ Computer Vision System Toolbox [4] để xử lý, cụ thể là các hàm:

- Hàm chuyển ảnh màu sang ảnh cường độ: `vision.ColorSpaceConverter`
- Hàm lọc ảnh: sử dụng bộ lọc FIR 2D [5] bằng hàm `vision.ImageFilter`

6. Biểu diễn và mô tả ảnh

Bước này chính là bước trích chọn đặc trưng của ảnh. Trong bài toán nhận dạng đường đi của đề tài thì đặc trưng quan trọng đó là dấu phân cách làn đường. Bằng phương pháp biến ảnh cường độ thành ảnh nhị phân sẽ làm nổi bật các dấu phân cách làn đường.

Sử dụng hàm chuyển ảnh cường độ sang ảnh nhị phân: `vision.Autothresher`



Hình 7: Ảnh nhị phân

7. Nhận dạng và giải thích

Bước này sử dụng biến đổi Hough để xác định các đường thẳng trong ảnh. Các đường thẳng ở đây chính là các dấu phân cách làn đường.

Từ đó tính toán xác định làn đường bên trái, bên phải.

Đề tài sử dụng các hàm của công cụ Computer Vision System Toolbox để xử lý đường thẳng trong ảnh, cụ thể là các hàm:

- `vision.HoughTransform`
- `vision.LocalMaximaFinder`,
- `vision.HoughLines`.

Nguyên tắc để xác định làn đường trái phải

và sự lệch tâm đường: Dấu phân cách làn đường trái phải được xác định dựa vào giao điểm của nó với đường bao phía dưới cùng của ảnh. Chương trình sẽ tính toán khoảng cách từ tâm của khung ảnh tới các dấu phân cách làn đường. Từ đó xuất tín hiệu cảnh báo nếu một trong các trường hợp sau xảy ra:

- Dấu phân cách làn đường phía bên phải khung hình giao với đường bao phía dưới cùng của khung hình. Sẽ cho tín hiệu lệch phải.

- Dấu phân cách làn đường phía bên trái khung hình giao với đường bao phía dưới cùng của khung hình. Sẽ cho tín hiệu lệch trái.

- Không có dấu phân cách nào giao với đường bao phía dưới của khung hình. Sẽ cho tín hiệu đúng làn.

Việc chọn vị trí đặt camera từ bước thu nhận ảnh và chọn vùng quan sát của ảnh có ảnh hưởng rất lớn đến độ chính xác của chương trình.

8. Xuất tín hiệu điều khiển

Nếu đi đúng làn đường, chương trình sẽ gửi một tín hiệu kiểu String 'C' tới chân TX của cổng COM.

Khi lệch tim đường, chương trình sẽ gửi một tín hiệu kiểu String 'L' nếu lệch trái, 'R' nếu lệch phải tới chân TX của cổng COM.

Trong phần giao diện, tín hiệu được thể hiện bằng hình ảnh trong khung Lane, và hiển thị dạng chuỗi trong vùng Receiver (RX) nếu chân RX và TX được nối tắt.

Sử dụng các hàm `serial`, `fopen`, `fclose` trong Matlab để xuất tín hiệu ra cổng COM.

Trong quá trình lập trình, người thực hiện đã dùng phần mềm Proteus 8 Professional và Configure Virtual Serial Port Driver để mô phỏng và kiểm tra tín hiệu xuất ra của cổng COM.

9. Thiết kế giao diện

Giao diện của chương trình được thiết kế như hình 8 bằng công cụ MATLAB Creating Gra-

physical User Interfaces [6] Với các nút điều khiển:

- Nút Start Webcam: Camera hành trình.
- Nút Detection & Tracking: Bắt đầu xử lý.
- Nút Stop: Dừng chương trình.
- Nút Exit: Thoát chương trình.
- Nút Connect: Kết nối và truyền dữ liệu ra cổng COM.
- Nút Send: Để gửi thử tín hiệu ra cổng COM.



Hình 8: Giao diện

III. THỰC NGHIỆM VÀ KẾT QUẢ

1. Dữ liệu thực nghiệm

❖ Địa điểm thực nghiệm

- **Phòng thí nghiệm:** Thí nghiệm được thực hiện trên xe mô hình như hình 9, và mô hình đường như hình 10.

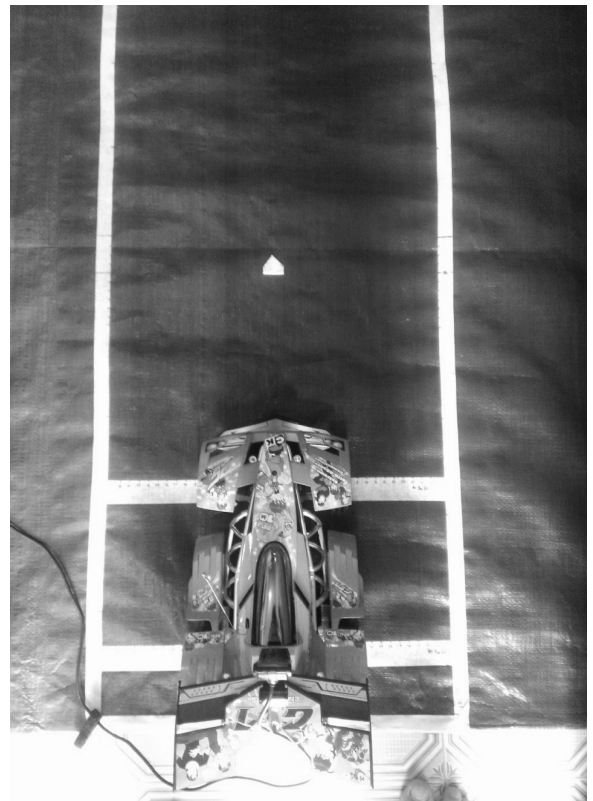


Hình 9: Xe mô hình

Kích thước xe mô hình: Chiều rộng cơ sở: 14.3cm; Chiều rộng toàn bộ: 17.5cm; Chiều dài cơ sở: 16.5cm; Chiều dài toàn bộ: 33cm; Chiều cao xe: 13.8cm. Khoảng cách từ vị trí camera tới cản trước: 22cm.



Hình 10: Mô hình đường



Hình 11: Mô hình thử nghiệm

- **Trên đường:** Toàn bộ dữ liệu thực nghiệm của đề tài được thực hiện bằng xe máy trên nhiều đoạn đường, nhưng cho kết quả tốt nhất là khi thực nghiệm trên các đoạn đường vắng xe (Đường Phạm Văn Đồng, Quận Thủ Đức; Đường Mai Chí Thọ, Quận 2).

❖ Về tốc độ xử lý

Chương trình thực thi của đề tài này được

thử nghiệm trên máy tính có cấu hình:

Bộ vi xử lý: Intel Core 2 Duo CPU T5800 @ 2.00GHz ; RAM : 2.00 GB ; HDD: 120 GB ; Hệ điều hành : Windows 8.1 Pro (32 bit).

Ngôn ngữ lập trình được sử dụng là Matlab với các bộ công cụ chính là Computer Vision System, Image Acquisition System và Image Processing System [7], MATLAB Creating Graphical User Interfaces [6].

Độ phân giải của camera ghi hình là 640x480, tốc độ ghi là 30fps.

2. Hình ảnh về kết quả thực nghiệm

a. Trong phòng thí nghiệm

- Xe đi đúng làn đường: Hình 12



Hình 12: Xe đi đúng làn đường

Khi xe đi đúng làn đường, chương trình sẽ gửi một chuỗi ký tự C đến cổng COM, đồng thời thể hiện bằng một hình ảnh màu xanh tại vị trí Center trong khung Lane của giao diện.

- **Khi xe lệch phải:** Hình 13

Khi xe lệch phải, chương trình sẽ gửi một chuỗi ký tự R đến cổng COM, đồng thời thể hiện bằng một hình ảnh màu vàng tại vị trí Right trong khung Lane của giao diện.



Hình 13: Xe lệch phải

- **Khi xe lệch trái:** Hình 14

Khi xe lệch trái, chương trình sẽ gửi một chuỗi ký tự L đến cổng COM, đồng thời thể hiện bằng một hình ảnh màu đỏ tại vị trí Left trong khung Lane của giao diện.



Hình 14: Xe lệch trái

b. Thực nghiệm trên đường

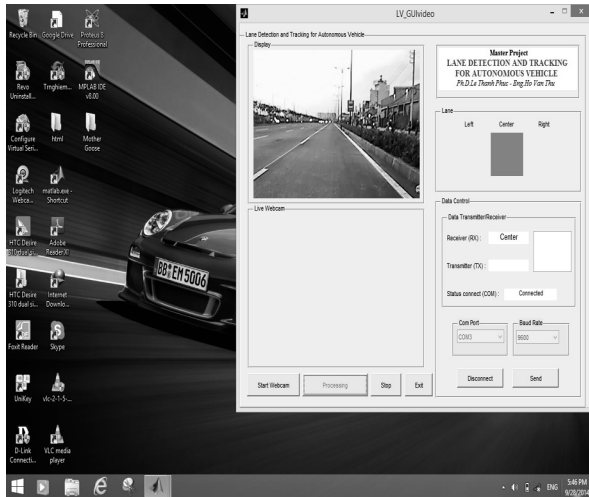
Các hình ảnh được ghi lại trên đường thử vào ban ngày, trời không mưa.

- **Khi xe đi đúng làn đường:** Hình 15

VI. ĐÁNH GIÁ KẾT QUẢ

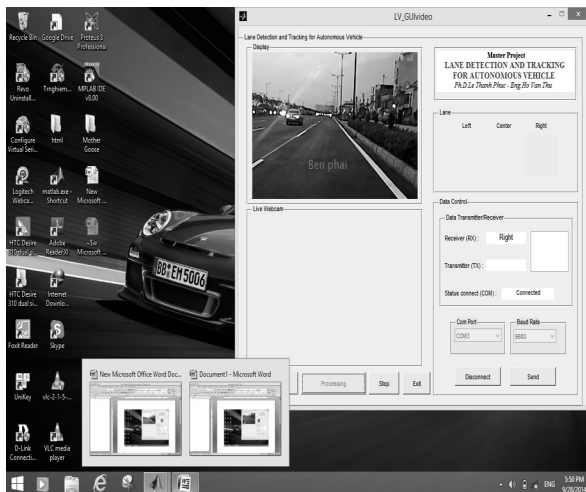
Bảng 1: So sánh về tốc độ xử lý của các nghiên cứu.

Nghiên cứu	Khả năng xử lý	Cấu hình
M. Bertozzi và A. Broggi [8]	128x128, 5fps	CPU 1.7GHz, sử dụng Matlab
ZuWhan Kim [9]	176x120, 2-3fps	P4 3GHz, C++ sử dụng OpenCV
Y. Wang, E.K. Teoh, and D. Shen [10]	128x128, 4spf	128MB Ram
Luận văn	640x480, 30fps	Core 2 Duo, 2.0GHz, Ram 2GB. Sử dụng Matlab.



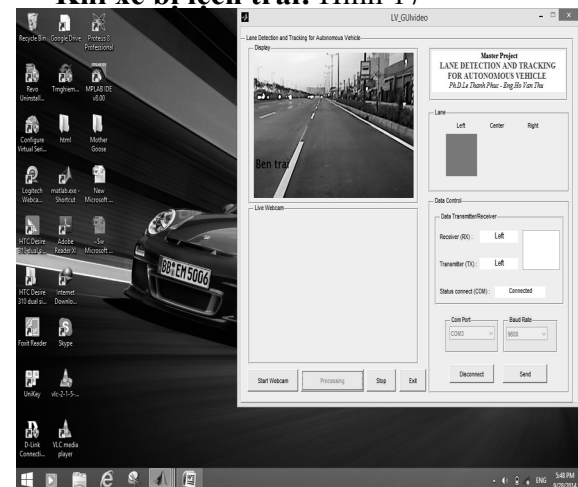
Hình 15: Xe đi đúng làn đường

- Khi xe bị lệch phải: Hình 16



Hình 16: Xe lệch phải

- Khi xe bị lệch trái: Hình 17



Hình 17: Xe lệch trái

Bài toán nhận dạng đường đi có nhiều hạn chế trong việc đánh giá kết quả. Bởi chưa có một tiêu chí chung nào cho việc đánh giá, so sánh kết quả giữa các nghiên cứu. Đồng thời mục tiêu đưa ra của mỗi nghiên cứu cũng không giống nhau nên việc đánh giá cũng còn nhiều hạn chế.

Tuy nhiên, trong các bài toán thu thập và xử lý dữ liệu phục vụ cho ô tô chạy tự động thì tiêu chí về tốc độ và độ chính xác luôn được xem xét trước tiên. Tiếp đó là tiêu chí về sự đổi mới so với các nghiên cứu trước đó.

Trong nghiên cứu này, người thực hiện đã thực nghiệm lặp lại nhiều lần và ghi nhận kết quả chủ yếu dựa trên tiêu chí về tốc độ để so sánh với các nghiên cứu trước đó.

Tốc độ xử lý của chương trình là khá tốt. Chương trình có thể xử lý trong thời gian thực dựa trên hình ảnh thu thập từ webcam với tốc độ ghi hình là 30fps, tốc độ xử lý và xuất tín hiệu trung bình là 10fps, tốc độ này phụ thuộc rất lớn vào tốc độ ghi hình và cấu hình của máy tính.

Như vậy, nếu xét về tốc độ xử lý thì chương trình của luận văn khá hiệu quả so với các nghiên cứu trước đó.

V. KẾT LUẬN

Mặc dù đề tài đáp ứng yêu cầu đặt ra ban đầu nhưng kết quả đạt được vẫn chưa như kỳ vọng. Sau khi hoàn thành, đề tài thể hiện những kết quả đạt được như sau:

- Ứng dụng được phần mềm Matlab để thu thập, nhận diện và xử lý ảnh ở mức cơ bản dựa vào hình ảnh thu thập được bằng webcam.

- Xuất được tín hiệu điều khiển và cảnh báo cho người lái biết vị trí của xe so với vị trí mục tiêu.

- Làm cơ sở để thực hiện các đề tài tiếp theo có liên quan đến xử lý ảnh và nhận dạng làn đường, điều khiển xe tự động.

Tuy nhiên, đề tài chỉ thực hiện ở mức độ cơ bản nên chưa thể ứng dụng vào thực tế, để ứng dụng được vào thực tế, cần phải thực hiện thêm: (1) Nhận dạng làn đường trên những đoạn đường cong nghiêm ngặt, có chướng ngại vật. (2) Sử dụng camera hồng ngoại để quan sát vào ban đêm. Nâng cao tốc độ ghi hình và xử lý. (3) Nhận dạng và xử lý làn đường có nhiều tín hiệu giao thông: Đường cấm, biển báo bắt buộc rẽ phải, rẽ trái, giới hạn tốc độ, đường giao nhau... (4) Nhận dạng và xử lý trên những làn đường không có vạch kẻ đường.

TÀI LIỆU THAM KHẢO

- [1] Oge Marques, *Practical Image and video processing using Matlab*, Wiley, 2011.
- [2] TS.Đỗ năng Toàn, TS.Phạm Việt Bình, *Giáo trình Xử lý ảnh*, Đại học Thái Nguyên, 2007.
- [3] The MathWorks Inc, *Image Acquisition Toolbox User's Guide*, 2013.
- [4] The MathWorks Inc, *Computer Vision System Toolbox User's Guide*, 2013.
- [5] TS.Hồ Văn Sung, *Xử lý ảnh số - Lý thuyết và thực hành với Matlab*, Nxb Khoa học kỹ thuật, 2013.
- [6] The MathWorks Inc, *MATLAB Creating Graphical User Interfaces*, 2013.
- [7] The mathWork Inc, *Image Processing Toolbox*, 2013.
- [8] M.Bertozzi and A.Broggi, "GOLD: A parallel real-time stereo vision system for generic obstacle and lane detection", *IEEE Transaction on Image Processing*, 1998, pp.199-213.
- [9] ZuWhan Kim, "Realtime Lane Tracking of Curved Local Road", in *IEEE Intelligent Transportation Systems*, Toronto, Canada, 2006, pp.1149-1155.
- [10] Y. Wang, E.K.Teoh, and D.Shen, "Lane detection and tracking using BSnake," *Image and Vision Computing*, vol. 22, no.4, 2004, pp. 269-280.