**Hệ thống cảnh báo xe chệch làn sử dụng biến đổi Hough và bộ lọc Kalman**

**Tóm tắt**

*Hệ thống cảnh báo chệch làn đường (Lane Departure Warning System – LDWS) là hệ thống tích hợp trên xe ô tô để phát hiện làn đường từ đó cảnh báo người lái khi xe chệch khỏi làn đường đang chạy giúp giảm thiểu tai nạn. Trong báo cáo này, chúng tôi trình bày một hệ thuống sử dụng tích hợp các giải thuật xử lý ảnh đơn giản giúp phát hiện giới hạn làn đường và đưa ra cảnh báo khi xe đi chệch làn. Giải thuật sẽ tiến hành phát hiện làn đường sử dụng thuật toán biến đổi Hough và theo dõi (tracking) bằng bộ lọc Kalman. Dựa trên các thông tin nhận được chúng tôi tính toán vị trí ô tô so với làn đường và đưa ra cảnh báo trạng thái tương ứng. Các kết quả thực nghiệm đã cho thấy giải thuật có thể được áp dụng trong những điều kiện phức tạp như điều kiện ban đêm, nhiều bóng râm.*

1. **Giới thiệu**

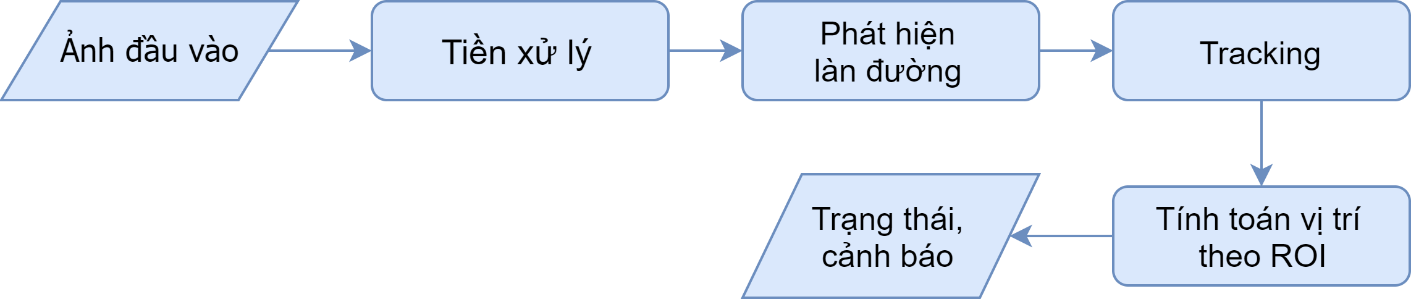
Hệ thống phát hiện làn đường và cảnh báo xe chệch làn đường dựa vào camera hành trình tho dõi các vạch kẻ đường. Chính vì thế khả năng phát hiện làn có thể bị hạn chế trong trường hợp trên đường không có vạch kẻ làn đường hoặc vạch kẻ bị mờ trong tầm ngắm của camera ,hoặc vạch kẻ đường không rõ ràng, trên mặt đường có nước hoặc cát che phủ và các đường cong từ trung bình đến lớn. Để khắc phục phần nào điều này, chúng tôi đề xuất một hệ thống bao gồm hai bước: phát hiện (detection) và theo dõi (tracking).

Ở bước thứ nhất, hệ thống phát hiện làn đường trên từng khung hình của camera hành trình. Dữ liệu khung hình được gửi về được qua các bước tiền xử lý như lọc trung vị (median blur), chọn vùng và màu sắc quan tâm (vàng và trắng), phát hiện cạnh. Ảnh sau khi tiền xử lý được đưa qua bộ biến đổi HoughLine để phát hiện làn đường chính xác. Kết quả phát hiện được biểu thị bằng hai đường thẳng cho làn trái và phải.

Ở bước theo dõi (tracking), hệ thống sử dụng bộ lọc Kalman để tính toán vị trí chính xác làn đường hiện tại dựa trên vị trí các đường thẳng ở những khung hình trước. Điều này giúp khử nhiễu và cải thiện đáng kể chất lượng nhận diện trong các điều kiện xấu.

Sau cùng, với vị tí vạch đường đã phát hiện, hệ thống sẽ tính toán trạng thái của xe dựa trên ị trí làn so với các vùng quan tâm (Region of Interest – ROI). Chúng tôi thiết đặt ba trạng trái chính để kiểm định kết quả: Không phát hiện được làn đường (No lane detected), xe trong làn (In lane) và xe ra khỏi làn (Out of lane). Toàn bộ chương trình viết bằng ngôn ngữ Python và thư viện OpenCV cùng các thư viện hỗ trợ.

Sơ đồ khối hệ thống:



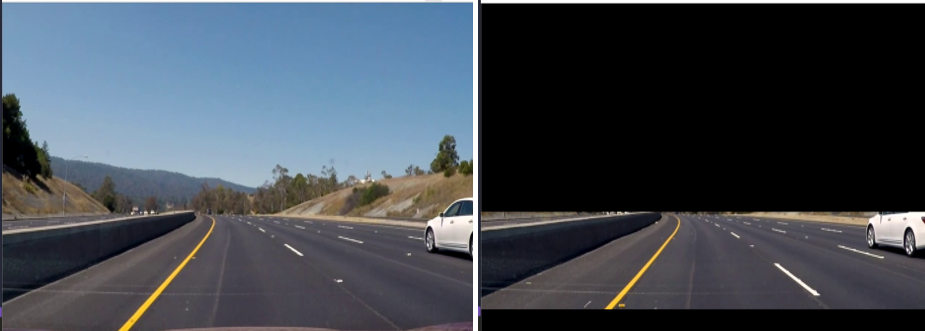
Hình 1. Sơ đồ khối hệ thống

**2. Thuật toán phát hiện làn đường**

Trong phần này, chúng tôi cố gắng sử dụng những thuật toán xử lý ảnh truyền thống để phát hiện làn đường trong ảnh. Trước tiên ảnh được đưa qua bộ tiền xử lý bao gồm: giới hạn vùng quan tâm (ROI), áp dụng các bộ lọc khử nhiễu, căn chỉnh gamma và phát hiện cạnh. Giải thuật phát hiện làn đường ý tưởng chính dựa trên phép biến đổi Hough Line (Hough Line Transform) tìm đường thẳng đồng thời tính toán xuất ra các hai làn đường phù hợp nhất.

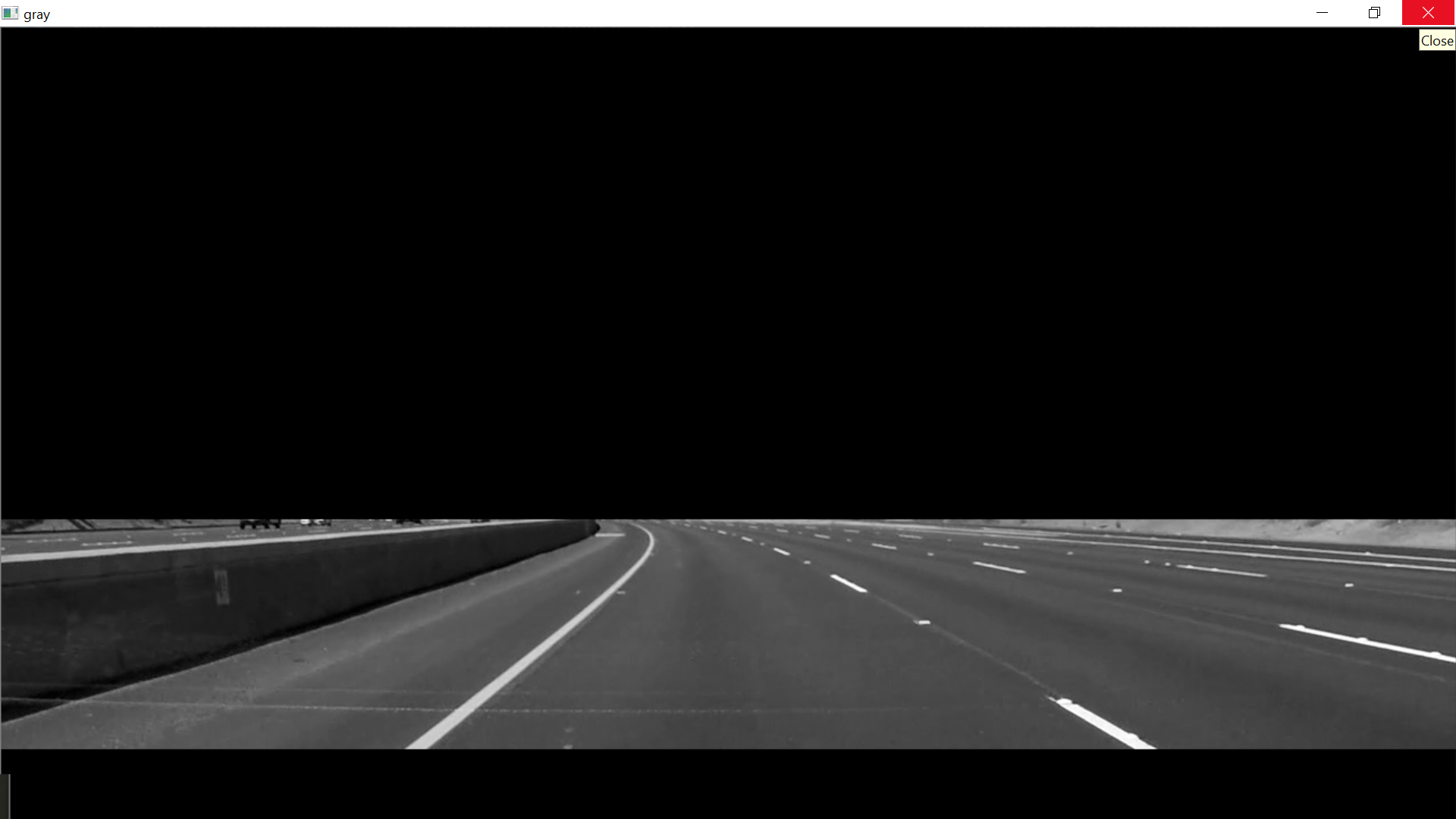
**2.1 Tiền xử lý**

**2.1.1 Giới hạn vùng trung tâm và lọc ảnh**

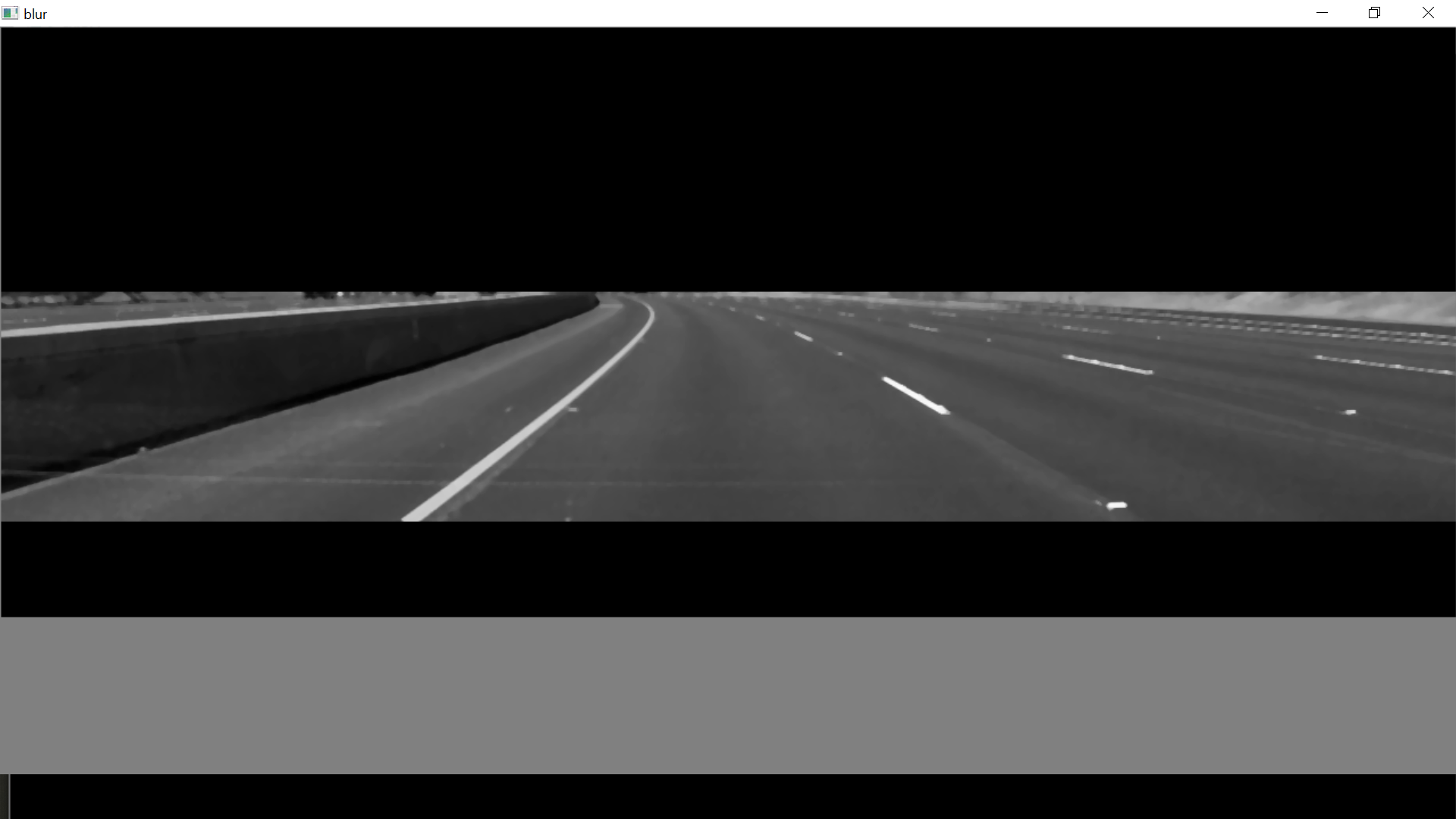
Để hạn chế nhiễu ngoại cảnh, ta giới hạn một vùng quan tâm (ROI) cụ thể trên ảnh. Cần đảm bảo vùng này chắc chắn chứa đầy đủ các vạch kẻ đường. Ở các bước tiếp theo chúng ta chỉ xử lý trên ROI này.

Hình 2. Giới hạn vùng quan tâm ROI

Nhằm thuận tiện cho tính toán và xử lý, ảnh màu BGR được chuyển sang ảnh mức xám (Gray-scale). Sau đó, ta dùng một bộ lọc trung vị (median-blur) lên ảnh mức xám này để lọc nhiễu. Bộ lọc trung vị áp dụng một cửa số 5x5 quét qua ma trận ảnh, sắp xếp sau đó lấy trung vị các giá trị phần tử trong ma trận gán cho điểm giữa ma trận cửa sổ.



Hình 3. Ảnh chuyển sang mức xám



Hình 4. Ảnh sau khi dùng bộ lọc Trung vị với cửa sổ 5x5



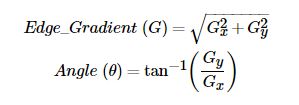
Hình 5. Trong điều kiện thiếu sáng, trước và sau bộ lọc trung vị - hình bên phải là ảnh sau khi qua bộ lọc

**2.1.2 Phát hiện biên dùng thuật toán Canny**

Để tăng hiệu quả việc phát hiện đường thẳng, trước tiên ta tách đường biên trong ảnh bằng giải thuật Canny. Giải thuật gồm 4 bước:

**Lọc nhiễu**

**Tính Gradient và hướng gradient**: ta dùng bộ lọc Sobel X và Sobel Y (3x3) để tính được ảnh đạo hàm Gx và Gy. Sau đó, ta tiếp tục tính ảnh Gradient và góc của Gradient theo công thức. Mỗi pixel trên ma trận này thể hiện độ lớn của biến đổi mức sáng ở vị trí tương ứng trên ảnh gốc. Tương tự, ma trận Angle cũng có cùng kích thước ảnh gốc, mỗi pixel trên Angle thể hiện góc, hay nói cách khác là hướng của cạnh. Giá trị góc này chuẩn hóa về 4 hướng tương ứng với 0, 45, 90, 180 độ.



**Non-maximum Suppression** (viết tắt NMS): loại bỏ các pixel ở vị trí không phải cực đại toàn cục. Ở bước này, ta dùng một filter 3x3 lần lượt chạy qua các pixel trên ảnh gradient. Trong quá trình lọc, ta xem xét xem độ lớn gradient của pixel trung tâm có phải là cực đại (lớn nhất trong cục bộ - local maximum) so với các gradient ở các pixel xung quanh. Nếu là cực đại, ta sẽ ghi nhận sẽ giữ pixel đó lại. Còn nếu pixel tại đó không phải là cực đại lân cận, ta sẽ set độ lớn gradient của nó về zero. Ta chỉ so sánh pixel trung tâm với 2 pixel lân cận theo hướng gradient.



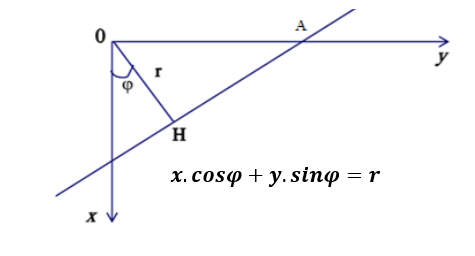
**Lọc ngưỡng**: ta sẽ xét các pixel dương trên mặt nạ nhị phân kết quả của bước trước. Nếu giá trị gradient vượt ngưỡng **max\_val** thì pixel đó **chắc chắn là cạnh**. Các pixel có độ lớn gradient nhỏ hơn ngưỡng **min\_val** sẽ bị loại bỏ. Còn các pixel nằm trong khoảng 2 ngưỡng trên sẽ được xem xét rằng nó có nằm liên kề với những pixel được cho là "chắc chắn là cạnh" hay không. Nếu liền kề thì ta giữ, còn không liền kề bất cứ pixel cạnh nào thì ta loại.



Hình 6.Dò biên ảnh với thuật toán Canny với max\_val = 120, min\_val = 60

**2.2 Phát hiện làn đường dùng biến đổi Hough**

Mỗi điểm trong mặt phẳng được biểu diễn bởi cặp trong tọa độ cực. Tương tự mỗi đường thẳng trong mặt phẳng cũng có thể biểu diễn bởi một cặp như hình sau:

****

Ta có mối liên hệ và như sau Biến đổi Hough ánh xạ n điểm này thành n đường sin trong tọa đô 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 sẽ cho duy nhất một cặp và có bao nhiêu đường đi qua sẽ có bấy nhiêu cặp giá trị .



Hình 7. Phát hiện làn đường với biến đổi Hough

**3. Thuật toán theo dõi làn đường**

Trong những điều kiện đặc biệt như trời tối, làn đường bị khuất lấp bởi vật chắn,v..v độ chính xác hệ thống bị ảnh hưởng mạnh. Để khắc phục nhiễu từ những điều này, chúng tôi xây dựng một bộ theo tracking vạch kẻ đường sử dụng bộ lọc Kalman.

Bộ lọc Kalman được dùng để đảm kết quả từ việc trích các tọa độ ứng với làn đường từ video không bị ảnh hưởng quá nhiều bởi nhiễu qua đó cho phép dự đoán chính xác hơn vị trí làn đường.

Xét chuyển động với vận tốc không đổi của một vật thể  trong không gian 2 chiều gồm theo hai phương tọa độ  và . Vị trí của  là một vector gồm 2 thành phần. Vector trạng thái của  tại một thời điểm là  phương trình trạng thái của  là:



Với

 ,  ,  là một quá trình ngẫu nhiên tuân theo phân phối Gauss có phương sai bằng  và trung bình bằng 0.

Phép đo tọa độ tức thời của vật thể tại một thời điểm cho kết quả chịu tác động của nhiễu Gauss trắng với phương sai bằng  và trung bình bằng 0.



Với

 , và 

Giải thuật lọc Kalman cho kết quả phép đo vị trí của vật thể này như sau:

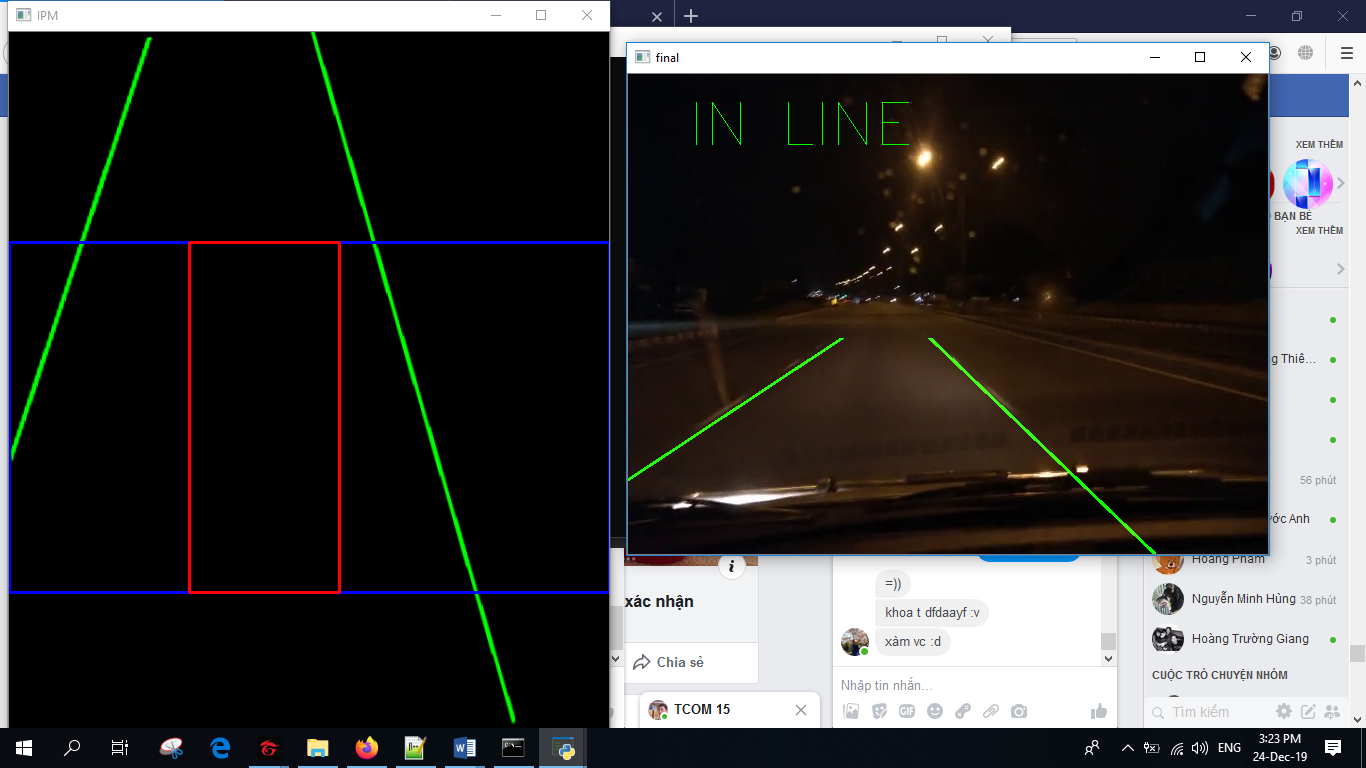
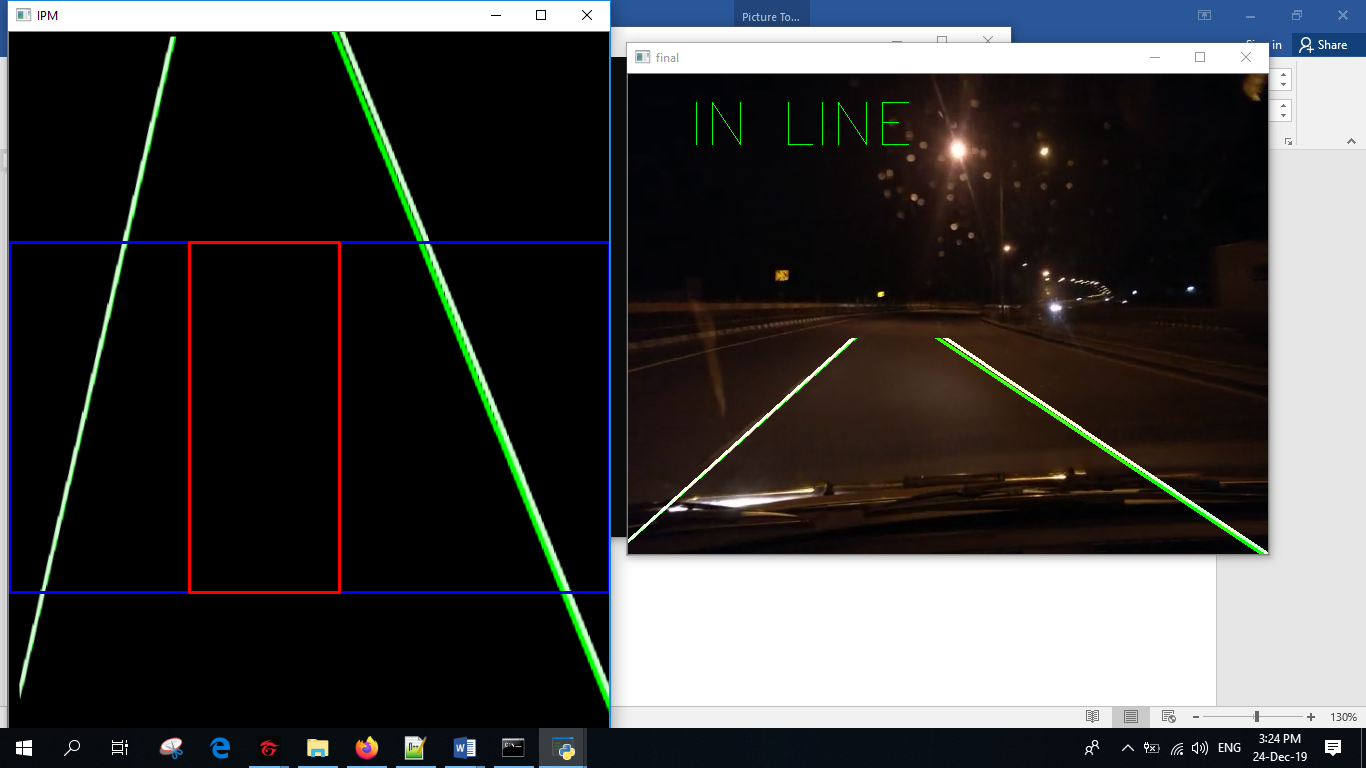


Dữ liệu được lọc trong bài toán này là một vector chứa tọa độ của các điểm được xác định thuộc về làn bên trái và làn bên phải 

Giải thuật Kalman được xây dựng tích hợp trong thư viện OpenCV các ma trận  khai báo lần lượt là sự mở rộng của trường hợp tọa độ hai chiều. Phương sai của nhiễu trong quá trình xử lý  là  = 0.1, phương sai của nhiễu tác động vào phép đo  là  = 15.

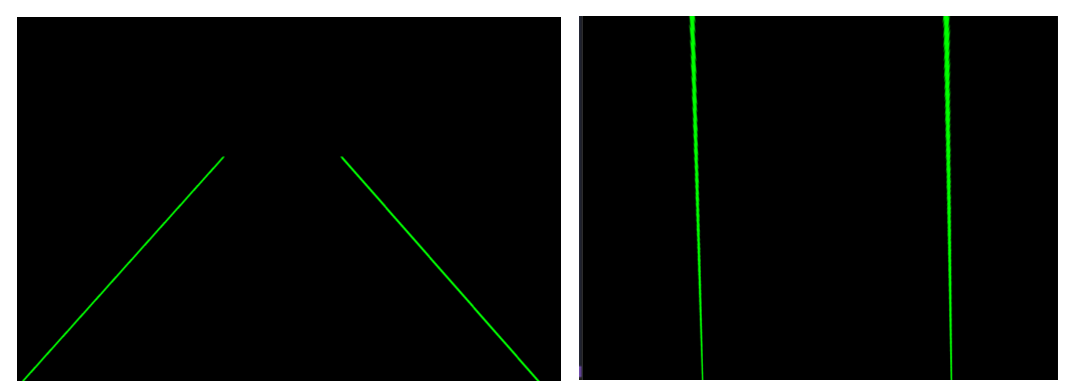
Hình 8.a Hình 8.b

Hình 8. Kết quả dự đoán của bộ lọc Kalman(màu xanh) và dữ liệu đọc được(màu trắng)

Thử nghiệm cho thấy ở *hình 8.b* thì bộ lọc Kalman đã dự đoán được gần đúng làn đường so với dữ liệu đo được. Hơn nữa ở *hình 8.a* khi không thể thu được dữ liệu về làn đường từ video thì bộ lọc Kalman vẫn cho phép hệ thống dự đoán được vị trí làn đường từ đó đưa ra quyết định chính xác hơn.

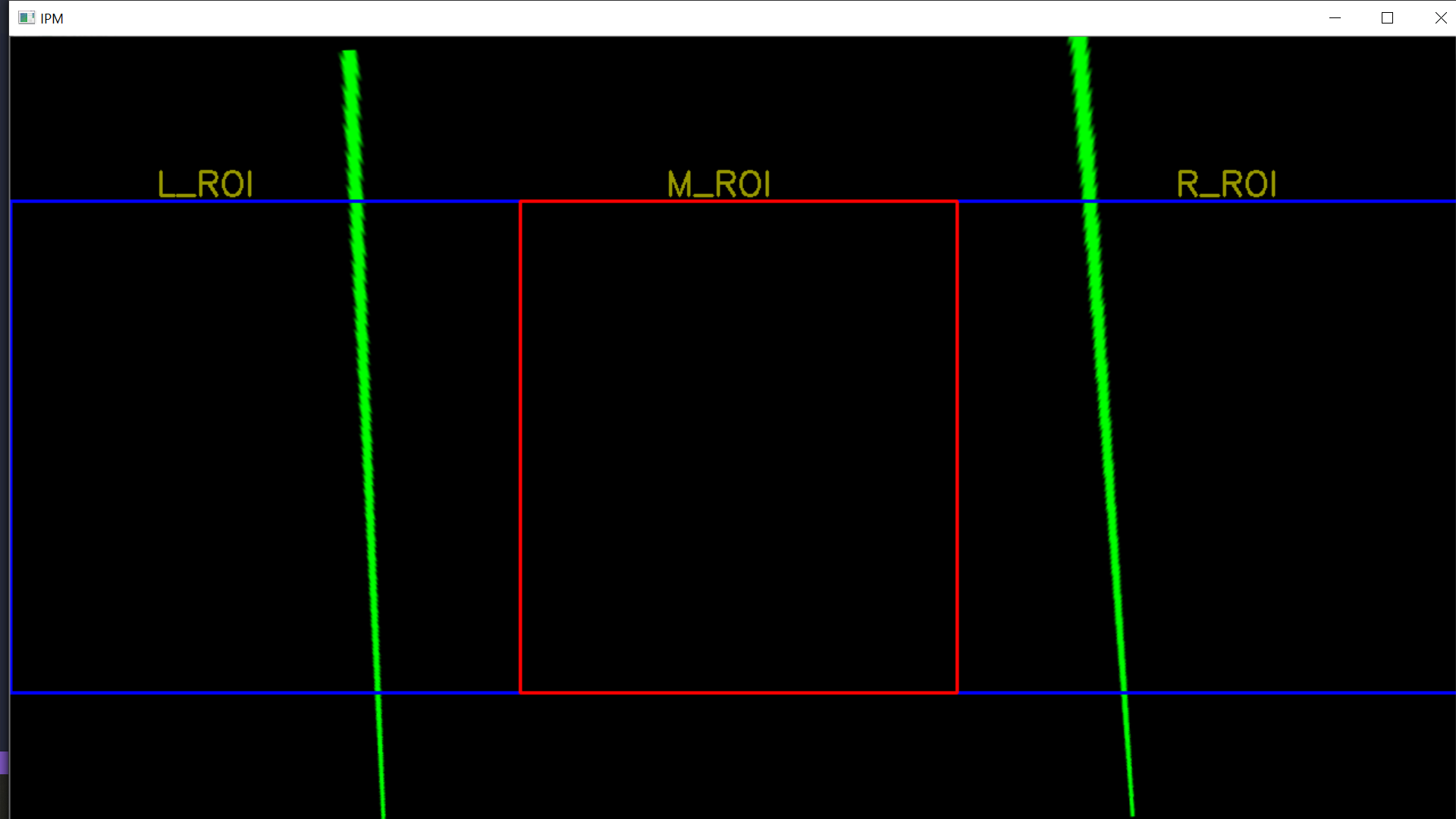
**4. Tính toán trạng thái và cảnh báo**

Sau khi có dữ liệu vạch kẻ đường, ta tính toán trạng thái xe so với làn đường. Để thiết đặt luật cảnh báo chính xác, ta thực hiện phép biến đổi kết quả làn đường về góc nhìn từ trên xuống hay “bird-eye view”. Sử dụng biến đổi Perspective-transform với bốn điểm ảnh gốc và bốn điểm đích mong muốn. Perspective-transform là một phép biến đổi tuyến tính ánh xạ từng điểm ảnh trong mặt phẳng phối cảnh này sang mặt phẳng phối cảnh khác. Thực hiện phép biến đổi bằng hàm cv2.getPerspectiveTransform với đầu vào là ma trận bốn điểm gốc và ma trận 4 điểm chuyển đổi tương ứng.



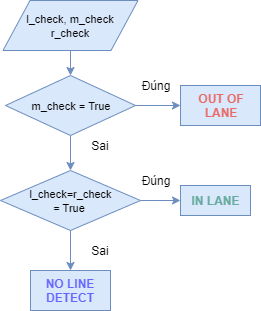
Hình 9. Làn đường phát hiện ban đầu và sau biến đổi phối cảnh sang góc nhìn “bird-eye”

Sau khi biến đổi, ảnh được đưa về góc nhìn từ trên xuống thuận tiện cho việc tham chiếu. Để xác định trạng thái xe so với làn đường, ta chia mặt phẳng tham chiếu thành 3 ROI khác nhau. Vị trí xe được xác định dựa theo vị trí hai làn đường so với ROI.



Hình 10. Ba vùng ROI trên mặt phẳng tham chiếu

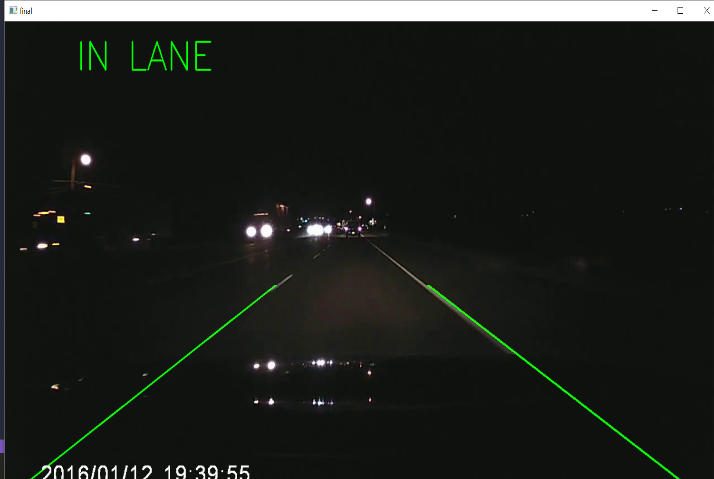
Ta đặt các biến l\_check, m\_check, r\_check là các biến kiểm tra sự xuất hiện của làn đường ở các ROI tương ứng. Tiến hành thiết đặt luật như sau:



Hình 11. Luật xác định trạng thái xe dựa vào vị trí làn đường

**5. Kết quả**

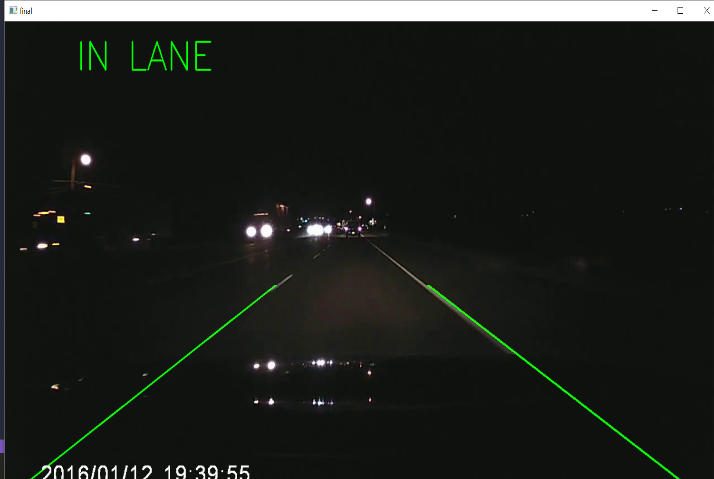
Giải thuật phát hiện và cảnh báo xe chệch làn được kiểm nghiệm với dữ liệu từ camera hành trình xe ô tô trong 2 điều kiện: ban ngày và ban đêm. Kết quả nhận diện và tracking khá ổn định. Đặc biệt, trong điều kiện trời tối chất lượng ảnh khá xấu nhưng vẫn cho kết quả tracking khả quan. Hệ thống có thể chạy real-time với tốc độ xử lý …



b)

a)

Hình 12. a) Kết quả nhận diện vào buổi sáng. b) Kết quả nhận diện vào buổi sáng



**6. Kết luận và đề xuất**

Trong báo cáo này, nhóm đề xuất một hệ thống cảnh báo xe chệch làn trong các điều kiện ngoại cảnh khác nhau dựa trên các thuật toán xử lý ảnh truyền thống. Cốt lõi giải thuật dựa trên việc nhận diện vạch kẻ đường dùng biến đổi Hough và tracking bằng bộ lọc Kalman. Tốc độ của hệ thống là ….

Một số điểm mạnh của hệ thống

Nhược điểm