

# Cảnh báo xe chệch làn sử dụng biến đổi Hough, bird-eyes-view và bộ lọc Kalman

Nguyễn Thái Sơn  
Việt Pháp 2015  
Đại học Bách Khoa  
TP. HCM, Việt Nam  
1512847@hcmut.edu.vn

Trần Đình Phước Anh  
Việt Pháp 2015  
Đại học Bách Khoa  
TP. HCM, Việt Nam  
1510101@hcmut.edu.vn

Nguyễn Minh Hùng  
Việt Pháp 2015  
Đại học Bách Khoa  
TP. HCM, Việt Nam  
1511355@hcmut.edu.vn

Đinh Võ Hoàng Tuấn  
Việt Pháp 2015  
Đại học Bách Khoa  
TP. HCM, Việt Nam  
1513814@hcmut.edu.vn

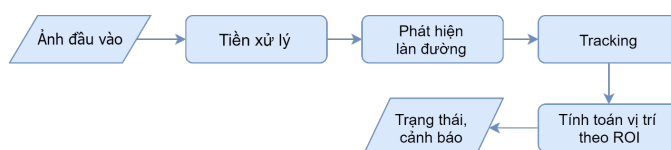
**Tóm tắt nội dung**—Hệ thống cảnh báo chệch làn đường (Lane Departure Warning System – LDWS) là hệ thống tích hợp trên xe ô tô để phát hiện làn đường từ đó cảnh báo người lái khi xe chệch khỏi làn đường đang chạy giúp giảm thiểu tai nạn. Trong báo cáo này, chúng tôi trình bày một hệ thống sử dụng tích 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.

**Từ khóa**—Lane detection . Lane tracking . Bird eyes view . Kalman filter . Hough transform

## I. GIỚI THIỆU

Hệ thống phát hiện làn đường và cảnh báo xe chệch làn đường dựa vào camera hành trình theo 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 trí vạch đường đã phát hiện, hệ thống sẽ tính toán trạng thái của xe dựa trên vị trí làn so với các vùng quan tâm (Region of Interest – ROI). Chúng tôi thiết đặt ba trạng thái 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

## II. LÝ THUYẾT

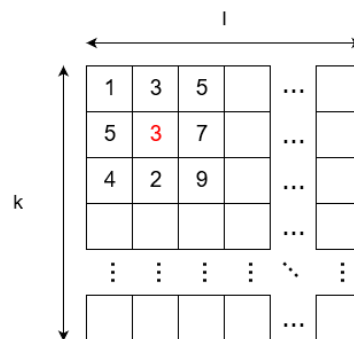
### A. Bộ lọc trung vị (Median Blur)

Bộ lọc không gian phi tuyến tính là bộ lọc trong đó các điểm ảnh có trong vùng xác định của bộ lọc sẽ được thống kê. Giá trị của điểm ảnh của trung tâm của bộ lọc sẽ được xác định bằng một xác quyết thống kê (trung vị, nhỏ nhất, lớn nhất...). Bộ lọc trung vị là bộ lọc không gian phi tuyến tính dùng xác quyết thống kê trung vị.

Xét một dãy giá trị  $X = \{x_1, x_2, \dots, x_N\}$  với  $x_i \leq x_j$  khi  $i < j$ ,  $1 \leq i, j \leq N$ .  $\alpha$  gọi là trung vị của  $X$  khi:

$$\alpha = X_m$$

với  $m = \frac{N+1}{2}$  khi  $N$  chẵn và  $m = \frac{N}{2}$  khi  $N$  lẻ. Xét một ảnh có kích thước  $k \times l$  với  $k, l \geq 1$ .



Hình 2: Ma trận mức xám của ảnh có kích thước  $k \times l$

Ta sẽ dùng một bộ lọc trung vị có kích thước  $3 \times 3$  để

lọc điểm ảnh tại vị trí màu đỏ. Từ định nghĩa của  $X$  và  $\alpha$  ta có:

$$X = \{1, 2, 3, 3, 4, 5, 5, 7, 9\}$$

và

$$\alpha = x_5 = 4$$

Bộ lọc trung vị sẽ thay giá trị của điểm ảnh được lọc bằng giá trị của  $\alpha$ .

1	3	5
5	3	7
4	2	9

 $\longrightarrow$ 

1	3	5
5	4	7
4	2	9

Hình 3: Giá trị vùng lọc trước(trái) và sau khi lọc(phải)

### B. Giải thuật Canny

1) *Cải thiện độ nét ảnh bằng toán tử Gradient thông qua các cửa sổ trượt Sobel*: Cho một hàm  $f(x, y)$ , gradient của  $f$  tại  $(x, y)$  được định nghĩa:

$$\nabla f = \text{grad}(f) = \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}$$

Độ lớn của vector  $\nabla f$ ,  $M(x, y)$  được định nghĩa:

$$M(x, y) = \text{mag}(\nabla f) = \sqrt{g_x^2 + g_y^2}$$

$M(x, y)$  thể hiện mức độ thay đổi của hàm số  $f(x, y)$  theo hướng của vector gradient  $\nabla f$ . Trong xử lý ảnh, nếu ta dùng toán tử gradient lên một ảnh  $A$  có kích thước  $k \times l$  thì kết quả sẽ là một ảnh  $A'$  cũng có kích thước  $k \times l$  và  $A'$  sẽ được gọi là gradient của  $A$ .

Do  $M(x, y)$  không phụ thuộc vào chiều lấy đạo hàm riêng của  $f(x, y)$  mà phụ thuộc vào độ lớn của kết quả này nên để đơn giản hóa việc tính toán  $M(x, y)$  trong một vài trường hợp ta có thể dùng biểu thức gần đúng:

$$M(x, y) \approx |g_x| + |g_y|$$

Toán tử Sobel là một bộ lọc sử dụng 2 cửa sổ trượt  $3 \times 3$  lên ảnh để thực hiện phép tính gần đúng của  $M(x, y)$ .

-1	-2	-1
0	0	0
1	2	1

-1	0	1
-2	0	2
-1	0	1

Hình 4: Hai cửa sổ trượt của toán tử Sobel

Xét một ma trận  $3 \times 3$  biểu thị mức xám trong một vùng có

$x_1$	$x_2$	$x_3$
$x_4$	$x_5$	$x_6$
$x_7$	$x_8$	$x_9$

Hình 5: Vùng có kích thước  $3 \times 3$  với  $x_i, i = 1..9$  là các mức xám của điểm ảnh tương ứng

kích thước tương ứng của một ảnh.

Lúc này độ lớn của gradient tại  $x_5$ ,  $M(x_5)$  sẽ bằng:

$$M(x, y) = M(x_5) \approx |(x_7 + 2x_8 + x_9) - (x_1 + 2x_2 + x_3)| + |(x_3 + 2x_6 + x_9) - (x_1 + 2x_4 + x_7)|$$

2) *Non-maximum suppression*: Sau khi sử dụng toán tử gradient lên ảnh  $A$  và thu được ảnh  $A'$  là gradient của  $A$ . Một cửa sổ trượt  $3 \times 3$  sẽ được trượt qua các điểm ảnh của  $A'$ . Với luật được định nghĩa như sau:

Xét một vùng ảnh  $3 \times 3$  với  $x_i, i = 1..9$  là các giá trị mức xám tương ứng của các điểm ảnh trong vùng:

$x_1$	$x_2$	$x_3$
$x_4$	$x_5$	$x_6$
$x_7$	$x_8$	$x_9$

Hình 6: Vùng có kích thước  $3 \times 3$  với  $x_i, i = 1..9$  là các mức xám của điểm ảnh tương ứng

Ta định nghĩa  $M$  là tập các giá trị cực đại theo hàng và theo cột ứng với điểm trung tâm của vùng ảnh.

$$M = \{m_1, m_2\}$$

với  $m_1 = \max(x_2, x_5, x_8)$  và  $m_2 = \max(x_4, x_5, x_6)$ .

Khi dùng cửa sổ trượt non-maximum suppression lên vùng ảnh trên thì giá trị của  $x_5$  sẽ được quyết định theo luật:

$$x_5 = \begin{cases} x_5, & \text{nếu } x_5 \in M. \\ 0, & \text{nếu } x_5 \notin M. \end{cases}$$

Kết quả của phép toán này là một ma trận  $A''$  có kích thước bằng  $A$  và  $A'$ .

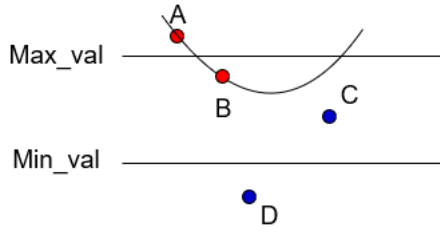
3) *Lọc ngưỡng*: Đầu vào của bước này là ảnh  $A''$ , một ngưỡng trên  $\text{maxval}$  và ngưỡng dưới  $\text{minval}$  sẽ được thiết đặt. Gọi  $P_1$  tập các điểm ảnh thuộc  $A''$  và thỏa

$$P_1 = \{p | p \leq \text{maxval} \vee (\text{minval} \leq p \leq \text{maxval} \wedge \text{lân cận của } p \in P_1)\}$$

Đối với một điểm ảnh  $p_i$  trong  $A''$  ta có:

$$p_i = \begin{cases} 255, & \text{nếu } p_i \in P_1. \\ 0, & \text{nếu } p_i \in P_2. \end{cases}$$

Lúc này ảnh thu được là  $A'''$  sẽ là kết quả của giải thuật Canny.



Hình 7: A, B là các điểm thỏa điều kiện cạnh còn C, D sẽ bị loại



Hình 8: Ảnh ban đầu (trên) và ảnh sau khi đã xử lý bằng giải thuật Canny (dưới)

### C. Bộ lọc Kalman

Xét chuyển động của một vật thể  $O$  trong không gian 2 chiều theo hai hướng  $x$  và  $y$ . Vị trí của  $O$  là một vector gồm 2 thành phần  $X = [x \ y]^T$ . Vector trạng thái của  $O$  tại một thời điểm là  $Y = [x \ v_x \ y \ v_y]^T$ , phương trình trạng thái của  $O$  là:

$$Y_n = AY_{(n-1)} + B\mu$$

với

$$A = \begin{bmatrix} 1 & dt & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & dt \\ 0 & 0 & 0 & 1 \end{bmatrix}, B = \begin{bmatrix} \frac{1}{2}dt^2 & 0 \\ dt & 0 \\ 0 & \frac{1}{2}dt^2 \\ 0 & dt \end{bmatrix},$$

$\mu$  là một quá trình ngẫu nhiên tuân theo phân phối Gauss có phương sai bằng  $\sigma_\mu^2$  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  $\sigma_w^2$  và trung bình bằng 0.

$$X_{measure} = HY + W$$

với

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, W = \begin{bmatrix} w \\ w \end{bmatrix},$$

Giải thuật lọc Kalman để cho kết quả gần đúng với quỹ đạo thật của vật thể dựa trên kết quả đo như sau:

$$Y_{n|n-1} = A \times Y_{n-1|n-1}$$

$$R_{Y_{n|n-1}} = A \times R_{Y_{n-1|n-1}} \times A^T + B \times R_\mu \times B^T$$

$$R_{Y_{n|n}} = H \times R_{Y_{n|n-1}} \times H^T + R_w$$

$$K = \frac{R_{Y_{n|n-1}} \times H^T}{R_{X_{n|n}}}$$

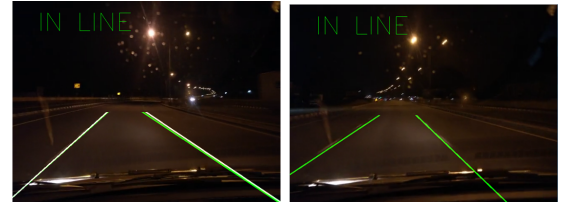
$$Y_{n|n} = Y_{n|n-1} + K \times (X_{measure} - H \times Y_{n|n-1})$$

$$R_{Y_{n|n}} = (I - K \times H) \times R_{Y_{n|n-1}}$$

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

$$X_{pos} = [x1_l \ y1_l \ x2_l \ y2_l \ x1_r \ y1_r \ x2_r \ y2_r]$$

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  $\sigma_\mu^2 = 0.1$ , phương sai của nhiễu tác động vào phép đo  $\sigma_w^2 = 15$ .



(a) Kết quả bộ lọc Kalman (xanh) và kết quả dò làn đường (trắng) (b) Kết quả bộ lọc Kalman khi không phát hiện được làn đường

Hình 9: Kết quả của việc dùng bộ lọc Kalman

Thử nghiệm cho thấy ở hình 9a thì bộ lọc Kalman đã dự đoán được gần đúng làn đường (màu xanh) so với dữ liệu đo được (màu trắng). Hơn nữa ở hình 9b thì 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.

### III. THIẾT KẾ PHẦN MỀM

#### KẾT LUẬN