

# TỰ ĐỘNG NHẬN DẠNG VÀ ĐỊNH VỊ HƯ HỎNG MẶT ĐƯỜNG DỰA TRÊN CAMERA HÀNH TRÌNH TÍCH HỢP GPS

## AUTOMATIC ROAD DAMAGE DETECTION AND LOCATING BASED ON BUILD-IN GPS CAMERA MOUNTED ON VEHICLES

Trần Trung Nguyên<sup>1</sup>, Nguyễn Công Hà<sup>2</sup>, Bùi Trọng Hiếu<sup>3</sup>, Lê Văn Quốc Anh<sup>4</sup>

<sup>1</sup>Học viên Cao học, ĐH GTVT TP.HCM

<sup>2,3,4</sup>Khoa Công nghệ thông tin, ĐH GTVT TP.HCM

**Tóm tắt:** Thu thập thông tin về ô gà trên mặt đường đô thị là một nhiệm vụ khá quan trọng và có ý nghĩa lớn không chỉ cho mục đích bảo trì, sửa chữa và quản lý đường bộ mà còn cho mục đích cung cấp các dịch vụ trong các hệ thống giao thông thông minh (HTGTTM) (ví dụ: tự động đưa thông tin cảnh báo các ô gà đến người điều khiển phương tiện giao thông hay cung cấp dữ liệu về tình trạng đường sá cho các đơn vị cung cấp dịch vụ bảo trì). Bài báo này đưa ra một giải pháp mới và hiệu quả để giải quyết bài toán phát hiện và định vị ô gà tự động qua camera hành trình gắn trên phương tiện giao thông. So với các phương pháp hiện có, những ưu điểm đáng kể của tiếp cận này là: (1) chi phí thực hiện thấp, (2) quy trình thực hiện tương đối đơn giản, và (3) độ chính xác cao. Chúng tôi đưa ra một quy trình hoàn chỉnh từ bước đầu tiên là tiền xử lý hình ảnh video đến bước cuối cùng là trực quan hóa các ô gà trên bản đồ. Kết quả thực nghiệm cho thấy giải pháp được đề xuất trong bài báo này có thể áp dụng được ngay trên các tuyến đường ở Thành phố Hồ Chí Minh cùng các tỉnh lân cận nói riêng và Việt Nam nói chung.

**Từ khóa:** Kỹ thuật trừ nền, ô gà, phát hiện ô gà, nhận dạng ô gà, định vị ô gà.

**Abstract:** Collecting information about potholes in city roadways is a very important and meaningful task not only for the purpose of road maintenance but also for providing intelligent transportation systems (ITS) services (e.g., to warn drivers of potholes on their ways or to provide road condition data for maintenance services). In this paper, we propose a novel approach to the problem of automatically detecting and locating potholes by using video camera mounted on vehicles. Comparison to existing methods, our approach has the following advantages: (1) low cost, (2) simple in deployment and (3) high accuracy. We provide a comprehensive framework consisting of several stages, from the first stage of video image pre-processing to the final stage of pothole visualization on the map. Experiments on real data show that our approach can be applied to roadways in Ho Chi Minh City as well as other cities in Vietnam.

**Keywords:** Background subtraction, potholes, potholes detecting, potholes locating

### 1. Giới thiệu

Tình trạng xuống cấp, hư hỏng mặt đường như ô gà, lún, lồi, nứt, gãy (hình 1) xuất hiện trên nhiều tuyến đường bộ ở Việt Nam nói chung và Thành phố Hồ Chí Minh nói riêng đã trở thành mối quan tâm lớn đối với người điều khiển phương tiện giao thông cũng như đối với các đơn vị bảo trì, sửa chữa và quản lý đường bộ. Các ô gà có thể là nguyên nhân xảy ra các vụ tai nạn giao thông gây thiệt hại về người và tài sản. Các hư hỏng mặt đường như ô gà cần được sửa chữa một cách kịp thời. Tuy nhiên, hiện nay, việc phát hiện ô gà được tiến hành một cách thủ công và tốn nhiều nhân lực [1]. Các nhân viên được phân công phụ trách theo từng con đường. Họ theo dõi, phát hiện và báo cáo các hư hỏng trên mặt đường theo định kỳ.



Hình 1. Một số dạng hư hỏng mặt đường dạng lún, nứt, ô gà.

Một số phương pháp và công nghệ hiện nay có thể hỗ trợ các đơn vị quản lý giao thông để giảm thiểu hại liên quan đến ô gà đã được công bố trên thế giới, ví dụ [2], [3], [4]. Trong đó, sử dụng laser để quét bề mặt đường là một phương pháp đặc biệt mới đã được sử dụng ở Thành phố New York. Tuy nhiên, những phương pháp như vậy rất tốn kém và đòi hỏi một chiếc xe chuyên dụng, với các thiết bị trị giá hàng trăm ngàn đô la [2].

Chúng tôi đưa ra một giải pháp với chi phí thấp hơn để giúp các đơn vị bảo trì, sửa chữa và quản lý có thể quyết định nhanh chóng và kịp thời cho việc sửa chữa các hư hỏng mặt đường nhất là các dạng ổ gà, gãy, lún. Ý tưởng của giải pháp là dựa vào các phương tiện giao thông như xe buýt, xe taxi, ... được gắn camera hành trình tích hợp GPS kết nối Wifi, 3G hoặc GPRS, các phương tiện này, khi lưu thông trên đường gặp các vị trí mặt đường có hư hỏng, hệ thống sẽ nhận dạng và định vị tọa độ của các ổ gà từ đó hiển thị đánh dấu trên bản đồ giám sát. Đồng thời việc phát hiện ổ gà hệ thống sẽ tự động đưa thông tin để cảnh báo ổ gà tức thời cho tài xế đang điều khiển các phương tiện này.

Như đã nêu ở trên, giải quyết bài toán tự động nhận dạng và định vị ổ gà sẽ là một đóng góp có ý nghĩa quan trọng không chỉ cho mục đích bảo trì, sửa chữa và quản lý đường bộ mà còn cho mục đích cung cấp các dịch vụ trong các hệ thống giao thông thông minh (HTGTTM). Đóng góp này sẽ cải thiện an toàn và nâng cao hiệu quả hoạt động xã hội trong lĩnh vực giao thông vận tải.

Trong nghiên cứu này, chúng tôi đề xuất sử dụng giải pháp tự động nhận dạng và định vị ổ gà trên mặt đường dựa trên camera hành trình tích hợp GPS của phương tiện giao thông. Phần còn lại của bài báo được tổ chức như sau: trong phần tiếp theo, chúng tôi trình bày cơ sở lý thuyết được áp dụng trong nghiên cứu này. Phần ba mô tả quy trình tự động nhận dạng và định vị các ổ gà. Phần bốn báo cáo kết quả thực nghiệm của giải pháp đề xuất. Phần cuối trình bày kết luận và các hướng phát triển trong tương lai liên quan đến nghiên cứu này.

## 2. Cơ sở lý thuyết

Hướng tiếp cận được đề xuất trong bài báo này cốt lõi dựa trên các kỹ thuật xử lý ảnh, trong đó phương pháp trừ nền và lọc nhiễu được sử dụng trong bước tiền xử lý. Điểm độc đáo của giải pháp đề xuất trong nghiên cứu là kết hợp với kỹ thuật co giãn điểm ảnh để nâng cao độ chính xác của quá trình phát hiện ổ gà.

### 2.1. Phương pháp trừ ảnh nền

Hình ảnh video là một chuỗi các khung hình liên tiếp được thể hiện trong một khoảng thời gian. Ảnh nội hay tiền cảnh (foreground) trong bài toán xử lý hình ảnh video được xem là các đối tượng cần được tách ra khỏi cảnh nền (background). Với bài báo, ảnh nội chính là ổ gà và cảnh nền là mặt đường. Đối với trường hợp ảnh nội là đối tượng chuyển động, việc tách ảnh nội ra khỏi cảnh nền được thực hiện bằng thao tác đơn giản là so sánh các khung hình liên tiếp của dữ liệu video, từ đó bắt vết được các đối tượng chuyển động. Phương pháp tách cảnh nền (trừ ảnh nền) được sử dụng để bước đầu phát hiện ổ gà trên mặt đường từ hình ảnh video.

Ý tưởng chung của các phương pháp trừ ảnh nền hiện có là so sánh cường độ sáng (màu) của một điểm ảnh với cường độ sáng tại điểm đó trong mô hình nền (background model). Mô hình nền ở đây được tính toán từ chuỗi các khung hình trước đó. Có thể tóm gọn điều kiện để phân biệt một điểm là nền bằng công thức sau [4]:

$$|I_n(x) - B_n(x)| > \Delta \quad (1)$$

Trong đó:

$I_n(x)$ : Cường độ sáng của điểm ảnh tại vị trí  $x$  và tại thời điểm khung hình thứ  $n$ ;

$B_n(x)$ : Giá trị cường độ sáng của ảnh nền tương ứng cho vị trí  $x$  được ước lượng qua chuỗi hình ảnh từ  $I_0$  đến  $I_{n-1}$ ;

$\Delta$ : Ngưỡng, cho trước hoặc được ước lượng từ các khung hình trước đó.

Ban đầu mô hình nền được khởi tạo bằng chính khung hình đầu tiên  $B_0 = I_0$ . Ở các khung hình tiếp theo, mô hình nền được cập nhật đơn giản như sau:

$$B_{n+1}(x) = \alpha B_n(x) + (1-\alpha) I_n(x) \quad (2)$$

Trong đó, tham số  $\alpha$  được gán trước nằm trong khoảng  $[0,1]$ .

Công thức (1) và (2) được sử dụng để tạo ra tập các điểm cấu thành các vùng ảnh nội đại diện cho đối tượng chuyển động. Đầu ra của quy trình này là một ma trận trong đó các điểm ảnh nội sẽ có giá trị là 1 và các điểm ảnh nền sẽ có giá trị là 0. Đối với video

quay ô gà và mặt đường thì quá trình trừ nền hoàn tất sau khoảng 5 khung hình ( $\alpha = 0.9$ ).

Hình 2 minh họa kết quả của việc trừ ảnh nền đối với một video quay ô gà thực.



*Hình 2. Minh họa dưới đây mô tả cho kết quả thuật toán trừ ảnh.*

## 2.2. Lọc nhiễu

Như minh họa ở hình 2, sau khi trừ nền, vùng ảnh nổi sau khi được phát hiện còn rất nhiều nhiễu và điều này gây khó khăn cho việc nhận dạng đối tượng ô gà. Do đó các kỹ thuật lọc nhiễu cần được áp dụng để thuận lợi cho việc xác định đối tượng cùng với các thuộc tính của chúng. Trong bài báo này, chúng tôi sử dụng bộ lọc trung vị (median filter) [6] vì tính hiệu quả và hiệu năng của nó cho các giải pháp thời gian thực.

Lọc trung vị là một kỹ thuật lọc phi tuyến. Bộ lọc này thực hiện khá hiệu quả đối với hai loại nhiễu: nhiễu đóm (speckle noise) và nhiễu muối tiêu (salt-pepper noise), thường xuất hiện khá nhiều sau khi tách nền cho ô gà từ hình ảnh video. Kỹ thuật lọc này được chúng tôi kết hợp với kỹ thuật co-giãn ảnh, sẽ trình bày ở phần sau, cho kết quả khá tốt đối với bài toán nhận dạng ô gà.

Ý tưởng chính của thuật toán lọc trung vị áp dụng để xử lý ảnh là cho một bộ lọc quét qua lần lượt các điểm ảnh. Bộ lọc được định nghĩa từ trước dưới dạng một pattern (ví dụ dạng hộp hay chữ thập). Thông thường ma trận  $3 \times 3$  được sử dụng. Bộ lọc lần lượt quét qua các điểm ảnh, và tại vị trí mỗi điểm ảnh, các điểm ảnh lân cận che phủ bởi bộ lọc được sử dụng để tính toán giá trị của các điểm ảnh này. Cách tính như sau: sắp xếp các điểm ảnh trong cửa sổ này theo thứ tự (tăng dần hoặc giảm dần), gán điểm ảnh nằm chính giữa (trung vị) của dãy giá trị điểm ảnh đã được sắp xếp ở trên cho giá trị điểm ảnh đang xét của ảnh đầu ra.

Qua thực nghiệm, chúng tôi phát hiện rằng bộ lọc trung vị phát huy khá hiệu quả

đối với các loại nhiễu muối tiêu và đóm. Tuy nhiên, quá trình lọc cũng làm đứt đoạn đường biên bao quanh các ô gà. Một số trường hợp ô gà bị tách thành các cụm điểm nhỏ, rời rạc. Đối với bài toán phát hiện và cảnh báo ô gà thì điều này không trở thành vấn đề lớn. Tuy nhiên, với mục đích thu thập đầy đủ các thuộc tính, như kích thước, hình dáng, cùng với tọa độ tâm của ô gà thì quá trình tiền xử lý phải có kết quả tốt hơn, có nghĩa là đường bao ô gà sẽ liền mạch sau giai đoạn lọc nhiễu. Để giải quyết điều này, chúng tôi kết hợp bộ lọc trung vị và kỹ thuật co-giãn ảnh, trình bày trong phần sau đây.

## 2.3. Phép mở ảnh

Với mục đích loại bỏ cấu trúc và loại bỏ phần nhô, phần thừa ra đồng thời vẫn giữ nguyên cấu trúc và kích thước vùng ảnh, chúng tôi sử dụng phép mở ảnh. Về cơ bản, phép mở ảnh được thực hiện lần lượt hai bước sau: Thứ nhất là tiến hành co ảnh (erosion) và thứ hai là giãn ảnh (dilation).

Đối với ảnh biểu diễn bởi ma trận nhị phân, phép co hay giãn ảnh sẽ sử dụng phép toán erosion  $\ominus$  và dilation  $\oplus$  trên các điểm ảnh, qua một phần tử cấu trúc (thường sử dụng ma trận  $3 \times 3$ ). Độc giả quan tâm đến chi tiết hai phép toán này tham khảo trong [7]. Minh họa hai bước co-giãn ảnh qua ví dụ ở hình 3.

Bước 1: Co ảnh.

0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0
0 1 1 1 1 1 1 1 0	0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0

Máy trộn điểm ảnh

0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0

Phản tử cấu trúc

0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0

Kết quả

Bước 2: Giãn ảnh.

0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0

Máy trộn điểm ảnh

0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0

Phản tử cấu trúc

0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0

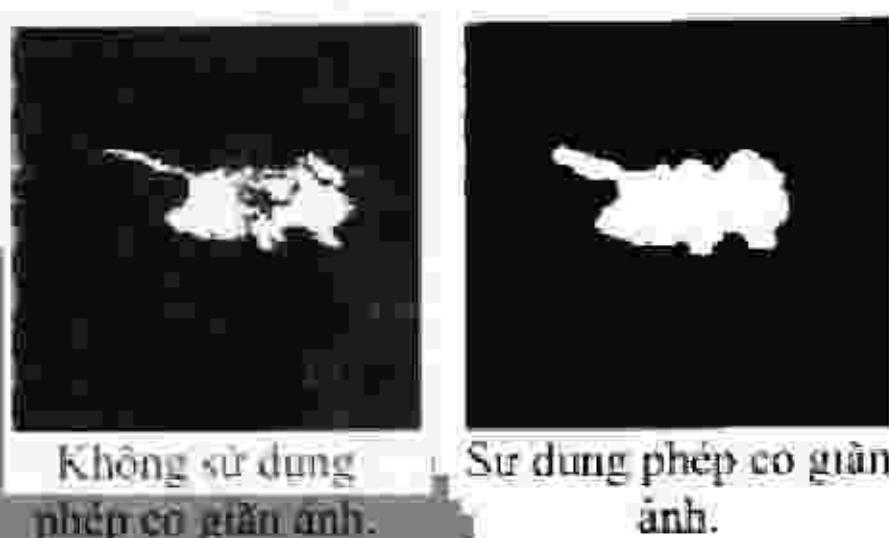
Kết quả

*Hình 3. Ví dụ minh họa phép co và giãn ảnh.*

Ví dụ minh họa ở trên cho thấy qua hai bước co và giãn ảnh thì các phần nhỏ nhô ra đã được loại bỏ, còn phần cấu trúc và kích

thuộc của vùng ảnh nổi (các điểm ảnh l) vẫn được giữ nguyên.

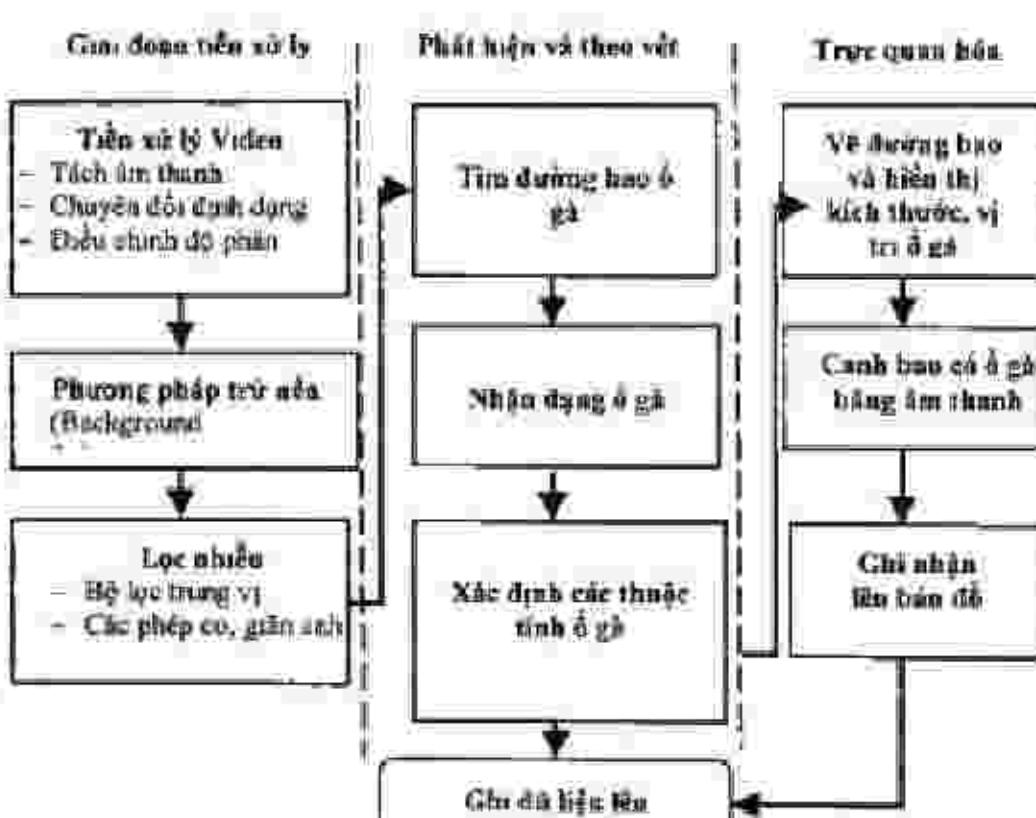
Lần lượt thử nghiệm trên hai ảnh nhị phân của một ô gà, so sánh hai ảnh này chúng tôi thấy các điểm không liên nhau trong được nối liền và các khoảng trống ở giữa đã được lấp đầy nhờ sự co giãn ảnh (hình 4).



**Hình 4. So sánh chất lượng ảnh khi sử dụng và không sử dụng phép co giãn ảnh.**

### 3. Quy trình phát hiện ô gà

Hình 5 trình bày quy trình được chúng tôi đề xuất để tự động nhận dạng và định vị ô gà qua video giao thông. Quy trình này được chia thành ba giai đoạn: Một là tiền xử lý, hai là phát hiện và theo dõi tượng và ba là trực quan hóa đối tượng.



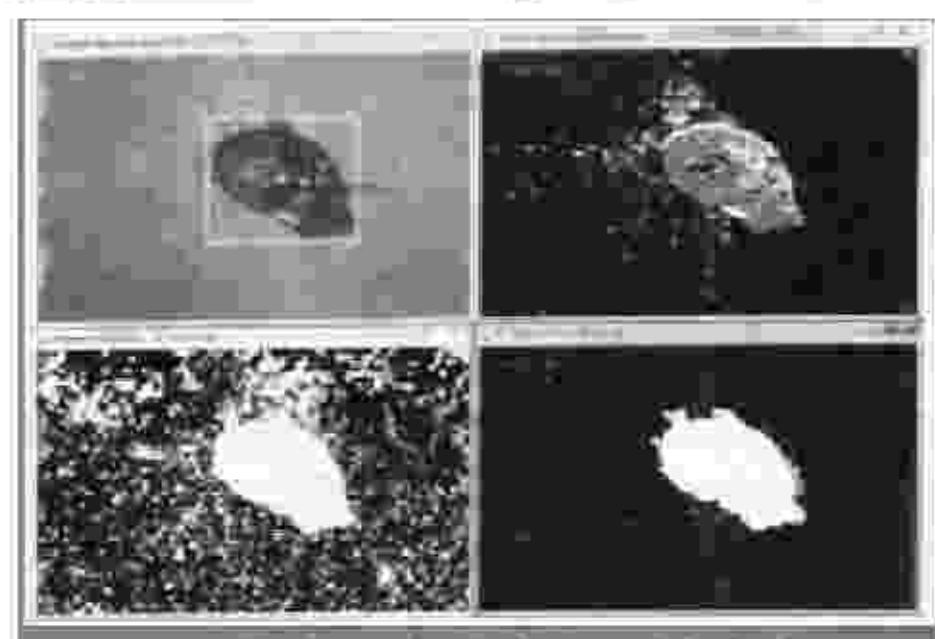
**Hình 5. Framework phát hiện ô gà.**

Giai đoạn tiền xử lý gồm các bước: tiền xử lý video, thực hiện phương pháp trừ nền, và xử lý lọc nhiễu.

Giai đoạn phát hiện và theo dõi đối tượng gồm các bước: tìm đường bao ô gà, nhận dạng ô gà, và xác định các thuộc tính của ô gà. Chúng tôi sử dụng bộ thư viện OpenCV để cài đặt các giải thuật xử lý ảnh được áp dụng trong bài báo này. Hàm findContours sẽ tìm đường bao quanh của các đối tượng đã được nhị phân hóa và lưu

đường bao này vào một vector là các điểm nằm trên đường bao đó. Hàm boundingRect sẽ tìm ra một hình chữ nhật bao một tập điểm của một đường bao được tìm ra từ hàm findContours.

Kết hợp hai hàm trên, chúng tôi cài đặt các hàm xử lý hiển thị trên màn hình ứng dụng các thuộc tính của ô gà như: đường bao quanh ô gà, hình chữ nhật bao quanh ô gà, kích thước, vị trí tâm của ô gà với các màu sắc khác nhau để dễ dàng nhận biết (hình 6).



**Hình 6. Ô gà được phát hiện.**

Ngoài ra, ứng dụng của chúng tôi cũng đưa ra tín hiệu âm thanh (tiếng bip) tức thời để cảnh báo người điều khiển phương tiện giao thông tại thời điểm phát hiện ô gà.

Sau ghi nhận dạng ô gà các dữ liệu về vị trí, kích thước, diện tích, hình ảnh của ô gà, ... sẽ được gửi về máy chủ thông qua tín hiệu GPRS hoặc 3G. Máy chủ sẽ xử lý dữ liệu và hiển thị lên bản đồ số (hình 7).



**Hình 7. Hiển thị ô gà lên bản đồ số**

Trong phần tiếp theo, chúng tôi sẽ trình bày một số kết quả thực nghiệm của giải pháp này.

### 4. Một số kết quả thực nghiệm

Để thực hiện việc kiểm nghiệm giải pháp đề xuất, chúng tôi thu thập dữ liệu thực trên các đường phố khu vực Quận Bình Thạnh, Thành phố Hồ Chí Minh. Các video mà chúng tôi thu thập có chứa một số đoạn (gồm các khung hình liên tiếp) có ô gà, và nhiệm

vụ của chương trình mà chúng tôi xây dựng phải phát hiện tự động và lọc ra các khung hình có ô gà này.

**Bảng 1. Kết quả thực nghiệm trên một số video.**

TT	Mô tả	Kết quả nhận dạng
1	Phát hiện ô gà có kích thước nhỏ (12 cm x 13 cm)	
2	Phát hiện ô gà có kích thước lớn (42 cm x 26 cm)	
3	Phát hiện hai ô gà có kích thước nhỏ gần nhau	

Trước khi đưa vào chương trình xử lý, các dữ liệu mẫu được tách âm thanh, chuyển đổi sang định dạng MJPG (Motion JPEG), điều chỉnh độ phân giải với kích thước 640x360 nhằm thống nhất kích thước khung hình. Với độ phân giải này thì tốc độ xử lý khung hình là 4 fps, sử dụng máy tính có cấu hình vừa phải (CPU Core i5, 8GB RAM, Card màn hình tích hợp của Intel). Đối với bài toán phát hiện ô gà thì tốc độ này là chấp nhận được vì một ô gà sẽ xuất hiện trên nhiều khung hình liên tục (10-30 frames, tùy tốc độ di chuyển của xe), và chỉ cần xử lý một vài khung hình để phát hiện ra ô gà này.

Để đánh giá độ chính xác của phương pháp, chúng tôi thu thập video khá đa dạng về chủng loại (ô gà kích thước lớn, ô gà kích thước bé, hai ô gà gần nhau) và điều kiện ngoại cảnh. Bảng 1 trình bày một số kết quả thực nghiệm với các loại ô gà và điều kiện ngoại cảnh khác nhau. Kết quả thực nghiệm cho thấy mức độ nhận dạng ô gà bằng phương pháp của chúng tôi để xuất khá tốt. Với những dữ liệu mà chúng tôi thu thập được thì độ chính xác xác suất trên 85%.

## 5. Kết luận

Bài báo này đã đưa ra một giải pháp tự động nhận dạng và định vị ô gà trên mặt đường dựa vào camera hành trình tích hợp GPS. Dựa trên nguồn dữ liệu video chúng tôi

thu thập được, giải pháp này đã được kiểm nghiệm. Kết quả thực nghiệm cho thấy mức độ nhận dạng khá tốt và hiệu quả.

Những thông tin về giao thông thu thập được có giá trị cho nhiều ứng dụng, trong đó có các ứng dụng liên quan đến HTGTTM. Trong các nghiên cứu sắp tới, chúng tôi sẽ sử dụng cơ sở dữ liệu ô gà thu thập được để xây dựng các ứng dụng hữu ích, chẳng hạn, ứng dụng cảnh báo ô gà trên thiết bị di động hay tìm đường thông minh.

## Lời cảm ơn

Nghiên cứu này được hỗ trợ từ nguồn kinh phí nghiên cứu khoa học của Trường Đại học Giao thông vận tải TP. HCM (MS KH1503) □

## Tài liệu tham khảo

- [1] Mashrur A. Chowdhury & Adel W. Sadek (2003). *Fundamentals of Intelligent Transportation Systems Planning*, Artech House Publishers.
- [2] Christoph Mertz. *Continuous road damage detection using regular service vehicles*, Technical Report, Carnegie Mellon University.
- [3] J. Eriksson, L. Girod, B. Hull, R. Newton, S. Madden, and H. Balakrishnan (2008), *Pothole Patrol: Using a Mobile Sensor Network for Road Surface Monitoring*, Proc. of The 6th International Conference on Mobile Systems, Applications, and Services.
- [4] Hsiu-Wen Wang, Chi-Hua Chen, Ding-Yuan Cheng, Chun-Hao Lin, and Chi-Chun Lo (2015), *A Real-Time Pothole Detection Approach for Intelligent Transportation System*, Mathematical Problems in Engineering.
- [5] Sen-Ching S. Cheung and Chandrika Kamath (2004). *Robust techniques for background subtraction in urban traffic video*. Applied Scientific Computing.
- [6] Yannick Beneteth, Pierre-Marc Jodoin, Bruno Emile, Helene Laurent, Christophe Rosenberger (2010). *Comparative study of background subtraction algorithms*. Journal of Electronic Imaging, Society of Photo-optical Instrumentation Engineers.
- [7] G.R. Arce (2005), *Nonlinear Signal Processing: A Statistical Approach*, Wiley:New Jersey, USA.
- [8] Pierre Soille (2003), *Morphological Image Analysis: Principles and Applications*, Signals & Communication, ISBN 3-540-65671-5.

Ngày nhận bài: 01/03/2016

Ngày hoàn thành sửa bài: 22/03/2016

Ngày chấp nhận đăng: 30/03/2016