

# THEO VẾT ĐỐI TƯỢNG TRONG VIDEO DỰA TRÊN ĐỘ LỢI THÔNG TIN

Ngô Dương Hà, Trần Như Ý, Lê Hữu Hà,  
Nguyễn Phương Hạc, Nguyễn Văn Tùng\*

Trường Đại học Công nghiệp Thực phẩm TP.HCM

\*Email: tungnv@hufi.edu.vn

Ngày nhận bài: 20/5/2020; Ngày chấp nhận đăng: 12/8/2020

## TÓM TẮT

Theo vết đối tượng trong video là một bài toán quan trọng cốt yếu trong lĩnh vực thị giác máy tính. Theo vết chuyển động của đối tượng có thể sử dụng trong các hệ thống an ninh, quan sát tự động, tích hợp vào robot, các thiết bị bay không người lái của quân đội,... Bài báo này trình bày hướng tiếp cận mô tả bài toán dưới dạng lọc một quá trình ngẫu nhiên Markov ẩn, sử dụng particle filter để lọc quá trình ngẫu nhiên và kết hợp giữa việc tính toán điểm tự tin từ gentle adaboost. Nhóm tác giả đề xuất thuật toán InfoPart cho phép theo vết đối tượng bằng phương pháp theo vết phần chứa nhiều thông tin đối tượng hơn (phần đầu, thân) là phần ít thay đổi theo thời gian và bỏ qua phần chứa ít thông tin (phần chân) là phần dao động nhiều. Kết quả thực nghiệm cho thấy độ chính xác trung bình của thuật toán InfoPart lớn hơn so với thuật toán GradNet.

*Từ khóa:* Bootstrap, mô hình Markov ẩn, gentle adaboost, particle filter, Haar like.

## 1. TỔNG QUAN

Bài toán theo vết đối tượng trong video được áp dụng trong nhiều lĩnh vực khác nhau, được sử dụng như một bài toán độc lập hay là một bài toán thành phần của bài toán lớn hơn. Ví dụ, trong lĩnh vực thể thao, việc theo dõi tự động cầu thủ chuyển động là một vấn đề quan trọng nhằm thuận tiện cho việc phân tích chiến thuật. Bài toán ở đây bao gồm việc phát hiện cầu thủ rồi sau đó theo vết chuyển động của cầu thủ này. Một ứng dụng cụ thể khác của bài toán là ứng dụng trong hệ thống xe lái tự động; Việc quan sát và theo vết các xe phía trước là một phần không thể thiếu để đảm bảo an toàn. Vì vậy, bài toán theo vết nếu kết hợp thêm với các hệ thống nhận diện, nhận dạng sẽ tạo ra một hệ thống có thể giải quyết nhiều vấn đề trong cuộc sống.

Hướng tiếp cận bộ lọc tương quan là công cụ mạnh mẽ trong xử lý tín hiệu số. Lọc thuật toán này xoay quanh việc khai thác các tính chất của biến đổi Fourier, tiêu biểu là tính chất biến phép tích chập trong miền không gian thành phép nhân hàm số trong miền Fourier [1-4]. Ý tưởng ban đầu của bộ lọc tương quan dùng để giải quyết bài toán định vị một vật trong ảnh. Nghĩa là, nếu vật được quan tâm có xuất hiện trong ảnh thì xác định vị trí của nó. Công cụ giải quyết bài toán này là Average of synthetic exact filters (ASEF) [1]. Bộ lọc tương quan tiếp theo là Minimum Output Sum of Squared Error (MOSSE) được nghiên cứu bởi David S. Bolme và các cộng sự [4]. Phương pháp theo vết này rất mạnh mẽ, có thể đối phó với các tình huống thay đổi ánh sáng, thay đổi kích cỡ, hình dáng của vật. Hơn thế nữa, tốc độ thực thi của phương pháp này rất ấn tượng khoảng 669 fps.

Hướng tiếp cận dựa trên phân loại nền và đối tượng sử dụng adaboost có công trình nghiên cứu của Shai Aviden [5] tại Mitsubishi Electric Research Labs xem xét việc theo vết đối tượng là bài toán phân loại nhị phân giữa các pixels nền và vật cần theo vết. Ý tưởng phương pháp là huấn luyện các hàm phân loại yếu để phân loại nền và vật rồi sau đó kết hợp lại để tạo thành một phân loại mạnh dựa trên cơ chế adaboost. Tuy nhiên, nhóm tác giả nhận thấy nếu vật không có dạng hình chữ nhật thì có những pixels thuộc khung hình chữ nhật chứa vật nhưng không thuộc vật sẽ được gán nhãn thuộc vật. Các pixels này được xem là các phần tử ngoại lai, adaboost thì lại nhạy cảm với các phần tử ngoại lai [6]. Ngoài ra, một số hạn chế khác của hướng tiếp cận: chưa giải quyết được tình huống vật bị che khuất hoàn toàn trong thời gian dài, vẫn phải đánh đổi giữa hiện tượng bị trôi đối tượng và khả năng thích nghi của mô hình theo thời gian, không gian đặc trưng được sử dụng trong thuật giải chưa tận dụng được thông tin về không gian của ảnh.

Hướng tiếp cận dựa trên lọc quá trình ngẫu nhiên đã được nghiên cứu trong thời gian dài trong lĩnh vực thống kê toán học và đã có rất nhiều kết quả ấn tượng được khám phá [7-9]. Đa số các thuật toán theo hướng tiếp cận này đều dựa trên lời giải tối ưu Bayes cho bài toán lọc quá trình Markov ẩn [10-12]. Nghĩa là xây dựng mô hình Markov ẩn đóng vai trò then chốt, mô hình càng chính xác với thực tế thì lời giải Bayes càng ước lượng được chính xác trạng thái của đối tượng. Công trình [11] có sử dụng đặc trưng về histogram màu sắc để xây dựng một particle filter để theo vết vật. Công trình [13] sử dụng gentle adaboost để xây dựng mô hình quan sát cập nhật theo thời gian.

Phần còn lại của bài viết được tổ chức như sau. Phần 2 trình bày một số công việc liên quan đến việc sử dụng Markov để lọc quá trình, phương pháp lấy mẫu, gentle adaboost cung cấp điểm tựa tin để phân loại và mô hình thực nghiệm particle filter. Phần 3 xây dựng thực nghiệm gồm nêu bài toán, đề xuất ý tưởng và mã giả theo vết đối tượng với việc sử dụng phần chứa nhiều thông tin cho quá trình theo vết. Kết quả thử nghiệm được trình bày trong phần 4. Kết luận và hướng phát triển được trình bày trong phần 5.

## 2. CƠ SỞ LÝ THUYẾT

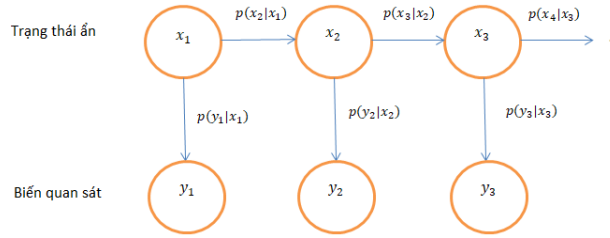
### 2.1. Lọc quá trình ngẫu nhiên Markov

Nếu trạng thái thứ  $n$  của quá trình ngẫu nhiên rời rạc chỉ phụ thuộc vào trạng thái thứ  $n-1$ , với mọi  $n$  thì ta có thể mô hình hóa dưới dạng quá trình ngẫu nhiên Markov.

Ví dụ về thời tiết ở Salzburg: *Các du khách người Ý rất thích thành phố Salzburg xinh đẹp, nơi được mệnh danh là “Rome của phía bắc”. Khi đến Salzburg họ nhanh chóng nhận ra thời tiết ở đây không ổn định như ở miền nam. Ở đây không bao giờ có 2 ngày liên tiếp mà thời tiết quang đãng. Nếu một ngày quang đãng, ngày tiếp theo sẽ mưa hoặc sẽ có tuyết rơi với xác suất như nhau. Một ngày mưa hoặc ngày có tuyết sẽ được theo sau bởi một ngày có cùng kiểu thời tiết hoặc thay đổi kiểu thời tiết với xác suất chia đôi; trong trường hợp thời tiết thay đổi, xác suất giữa 2 loại thời tiết để xảy ra sự thay đổi là như nhau.*

Với mô hình trên, du khách có thể đặt ra một số câu hỏi như: Hôm nay có mưa. Xác suất để 1 tuần nữa trời nắng đẹp là bao nhiêu? Kỳ vọng của số ngày mưa trong tháng tiếp theo là bao nhiêu?

Trong trường hợp không thể quan sát được trạng thái thực sự của quá trình ngẫu nhiên mà chỉ quan sát thông qua các biến quan sát thì mô hình dưới dạng Markov ẩn Hình 1 sẽ là giải pháp tốt.



Hình 1. Mô hình Markov ẩn

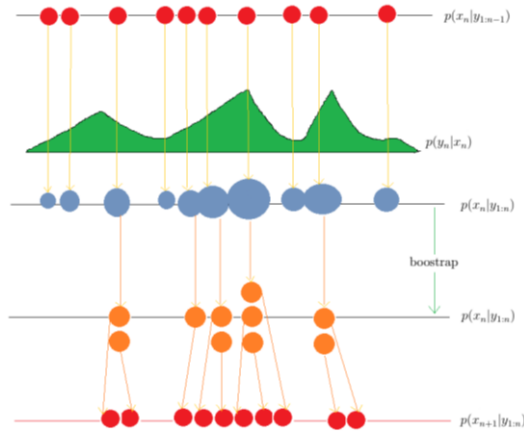
Mô hình Markov ẩn gồm 2 mô hình:

Mô hình chuyển trạng thái:  $p(x_k|x_{k-1})$ .

Mô hình quan sát:  $p(y_k|x_k)$ .

Dựa vào Markov ẩn gồm 2 mô hình trên sẽ tính xác suất  $p(x_k|y_k)$  xuất hiện của trạng thái  $x_k$  với biến quan sát  $y_k$ .

## 2.2. Mô hình particle filter



Hình 2. Minh họa cơ chế hoạt động của particle filter

Dựa vào cơ chế hoạt động của particle filter ở Hình 2, trình bày việc xấp xỉ hàm mật độ hậu nghiệm  $p(x_k|y_{1:k})$  cho mẫu particle ở mỗi thời điểm  $k$ . Sau khi tìm được hàm mật độ hậu nghiệm cho các mẫu particle, sử dụng bootstrap để loại bỏ và lựa chọn lại các mẫu particle cho lần lặp thứ  $k+1$ .

Quay trở lại với lời giải tối ưu Bayes cho hậu nghiệm ở mức thứ  $k$ :

$$p(x_k|y_{1:k}) = \frac{p(y_k|x_k) * p(x_k|y_{1:k-1})}{p(y_k|y_{1:k-1})}$$

Nhận xét đại lượng  $p(y_k|y_{1:k-1})$  chỉ là một hằng số khi ta xét hàm  $p(x_k|y_{1:k})$  theo biến  $x_k$ , do đó ta có công thức xấp xỉ:

$$p(x_k|y_{1:k}) \propto p(y_k|x_k) * p(x_k|y_{1:k-1}) \quad (1)$$

Ta có dự đoán tiên nghiệm:

$$p(x_k|y_{1:k-1}) = \int p(x_k|x_{k-1}) * p(x_{k-1}|y_{1:k-1}) dx_{k-1} \quad (2)$$

Tiếp tục thay thế  $p(x_k|y_{1:k-1})$  theo công thức (1) ta thu được:

$$p(x_k|y_{1:k}) \propto p(y_k|x_k) * \int p(x_k|x_{k-1}) * p(x_{k-1}|y_{1:k-1}) dx_{k-1} \quad (3)$$

Mục tiêu là xấp xỉ các hàm mật độ xác suất  $p(x_k|y_{1:k})$  với  $k \in N$ .

Giả sử biết hàm mật độ xác suất khởi đầu  $p(x_1)$  là phân phối đều. Ta sẽ trình bày một cơ chế quy nạp để từ một xấp xỉ cho  $p(x_{k-1}|y_{1:k-1})$  có thể tính tiếp một xấp xỉ cho  $p(x_k|y_{1:k})$ .

Giả sử  $\{\omega_{k-1}^l, x_{k-1}^l\}_{l=1}^L$  là một xấp xỉ cho  $p(x_{k-1}|y_{1:k-1})$ . Theo Monte-Carlo ta có:

$$\sum_{l=1}^L \omega_{k-1}^l \delta_{x_{k-1}^l} \approx p(x_{k-1}|y_{1:k-1}) \quad (4)$$

Được viết dưới dạng tương đương:

$$\forall g \in D(R^n), \int g(x_{k-1}) * p(x_{k-1}|y_{1:k-1}) dx_{k-1} \approx \sum_{l=1}^L \omega_{k-1}^l * g(x_{k-1}) \quad (5)$$

Ta chọn  $g(x_{k-1}) = p(x_k|x_{k-1})$  (biến  $x_{k-1}$ , cố định  $x_k$ ) thì ta thu được:

$$\int p(x_k|x_{k-1}) * p(x_{k-1}|y_{1:k-1}) dx_{k-1} \approx \sum_{l=1}^L (\omega_{k-1}^l * p(x_k|x_{k-1}^l)) \quad (6)$$

Kết hợp lại công thức (3) và (6) ta thu được công thức (7):

$$p(x_k|y_{1:k}) \propto p(y_k|x_k) * \sum_{l=1}^L (\omega_{k-1}^l * p(x_k|x_{k-1}^l)) = \sum_{l=1}^L (\omega_{k-1}^l * p(y_k|x_k) * p(x_k|x_{k-1}^l)) \quad (7)$$

Cuối cùng, vì các mô hình quan sát  $p(y_k|x_k)$ , chuyển trạng thái  $p(x_k|x_{k-1})$  do HMM cung cấp ngay từ đầu, nên đến đây ta đã có một hàm xấp xỉ cho hậu nghiệm  $p(x_k|y_{1:k})$ . Nhưng để tiếp tục sử dụng cho bước sau (dùng để tiếp tục tính  $p(x_{k+1}|y_{1:k+1})$ ) ta lấy mẫu từ  $p(x_k|y_{1:k})$ . Ta có công thức sau:

$$p(x_k|y_{1:k}) = \frac{1}{Z} \sum_{l=1}^L (\omega_{k-1}^l * p(y_k|x_k) * p(x_k|x_{k-1}^l)) \quad (8)$$

Lấy tích phân theo  $dx_k$  hai vế và  $\int p(x_k|y_{1:k}) dx_k = 1$  ta có công thức sau:

$$Z = \int \sum_{l=1}^L (\omega_{k-1}^l * p(y_k|x_k) * p(x_k|x_{k-1}^l)) dx_k \quad (9)$$

Nhận xét rằng  $\frac{1}{Z} \sum_{l=1}^L (\omega_{k-1}^l * p(y_k|x_k) * p(x_k|x_{k-1}^l))$  là một dạng trộn của hàm mật độ, ý tưởng ở đây là không áp dụng IS trên toàn bộ hàm trên, nhưng áp dụng cho từng hàm thành phần. Tức là ta lấy mẫu cho từng hàm mật độ  $\frac{1}{Z_l} * p(y_k|x_k) * p(x_k|x_{k-1}^l)$ , với  $Z_l$  là nhân tử sao cho hàm trên là hàm mật độ, cụ thể  $Z_l = \int p(x_k|x_{k-1}^l) dx_k$ .

Phần tiếp theo trình bày việc lấy mẫu dựa trên IS trên từng hàm thành phần và chứng minh bộ mẫu đó có thể đại diện cho hàm tổng ban đầu.

Áp dụng IS với  $\pi(x_k) = \frac{1}{Z_l} * p(y_k|x_k) * p(x_k|x_{k-1}^l)$ ,  $q(x_k) = p(x_k|x_{k-1}^l)$ .

Lấy  $x_k^l \sim p(x_k|x_{k-1}^l)$  thì  $(x_k^l, \frac{1}{Z_l} p(y_k|x_k^l))$  đại diện cho  $\pi(x_k)$  (trường hợp này thì  $N = 1$ ).

Ta chứng minh  $\{x_k^l, \frac{1}{Z} \omega_{k-1}^l * p(y_k|x_k^l)\}_{l=1}^L$  đại diện cho  $\frac{1}{Z} \sum_{l=1}^L (\omega_{k-1}^l * p(y_k|x_k) * p(x_k|x_{k-1}^l))$

Thật vậy điều này tương đương:

$$\frac{1}{Z} \omega_{k-1}^l * p(y_k|x_k^l) * \delta_{x_k^l} \approx \frac{1}{Z} \sum_{l=1}^L (\omega_{k-1}^l * p(y_k|x_k) * p(x_k|x_{k-1}^l)) \quad (10)$$

Lấy hàm  $g(x_k) \in D(R^n)$ , thực hiện biến đổi như sau:

$$\begin{aligned} \int g(x_k) * \frac{1}{Z} \sum_{l=1}^L (\omega_{k-1}^l * p(y_k|x_k) * p(x_k|x_{k-1}^l)) dx_k \\ = \frac{1}{Z} \sum_{l=1}^L \omega_{k-1}^l * \int g(x_k) p(y_k|x_k) p(x_k|x_{k-1}^l) dx_k \end{aligned}$$

$$\begin{aligned}
 &= \frac{1}{Z} \sum_{l=1}^L \omega_{k-1}^l * Z_l \int \frac{1}{Z_l} g(x_k) p(y_k | x_k) p(x_k | x_{k-1}^l) dx_k \\
 &\approx \frac{1}{Z} \sum_{l=1}^L \omega_{k-1}^l * Z_l * \frac{1}{Z_l} p(y_k | x_k^l) g(x_k^l) \\
 &= \frac{1}{Z} \sum_{l=1}^L \omega_{k-1}^l * p(y_k | x_k^l) g(x_k^l)
 \end{aligned}$$

Vậy ta đã tạo được một mẫu có trọng số đại diện cho hàm mật độ hậu nghiệm ở thời điểm thứ k:

$$\{x_k^l, \frac{1}{Z} \omega_{k-1}^l * p(y_k | x_k^l)\}_{l=1}^L \sim p(x_k | y_{1:k}) \quad (11)$$

### 2.3. Gentle adaboost

Là một thuật toán phân loại đồng thời cung cấp thêm điểm tự tin của phân loại [10]. Ý tưởng chính của gentle adaboost như sau:

Cho một bộ huấn luyện gồm n vectors  $x_1, x_2, \dots, x_n$ , mỗi vector có m chiều, mỗi vector được gán nhãn  $y_i = \pm 1$ . Ta tiến hành huấn luyện một phân loại yếu có dạng:

$$f(x) = a * \delta(x^{(k)} > \theta) + b$$

với x là vector input và  $x^{(k)}$  là thành phần thứ k của x;  $\delta$  là hàm Kronecker;  $\theta$  là ngưỡng; a, b là các hệ số hồi quy sao cho hàm lỗi đạt giá trị nhỏ nhất.

$$J = \sum_{i=1}^n \omega_i (y_i - f(x_i))^2$$

với  $\omega_i$  là trọng số được cập nhật sau mỗi lần phân loại yếu cho phân loại phần tử  $x_i$  đúng hay sai.

Điều này đồng nghĩa với việc tìm bộ bốn số  $(a, k, \theta, b)$  sao cho J đạt giá trị nhỏ nhất.

Lặp lại quá trình trên s lần, ta thu được s phân loại yếu trở thành phân loại mạnh theo công thức:

$$F(x) = \sum_{i=1}^s f_i(x)$$

Dựa vào công thức trên cho phép ta phân loại một phần tử dựa trên dấu của  $F(x)$  và điểm tự tin của phân loại dựa trên  $|F(x)|$ .

Quá trình huấn luyện gentle adaboost chính là tìm  $(a, k, \theta, b)$  sao cho J đạt giá trị nhỏ nhất.

$$(a, k, \theta, b) = \underset{i=1}{\operatorname{argmin}} \sum \omega_i (y_i - a\delta(x^{(k)} > \theta) - b)^2$$

Để tìm giá trị nhỏ nhất của J, ta cố định k,  $\theta$  trước, sau đó giải hệ gồm hai phương trình:

$$\frac{\partial J}{\partial a} = 0 \text{ và } \frac{\partial J}{\partial b} = 0$$

Ta thu được công thức tường minh của a, b là:

$$b = \frac{\sum_{j=1}^n \omega_j y_j \delta(x_j^{(k)} \leq \theta)}{\sum_{j=1}^n \omega_j \delta(x_j^{(k)} \leq \theta)} \text{ và } a = \frac{\sum_{j=1}^n \omega_j y_j \delta(x_j^{(k)} > \theta)}{\sum_{j=1}^n \omega_j \delta(x_j^{(k)} > \theta)} - b$$

Thay đổi  $k = 1, 2, \dots, m$  với mỗi  $k$  thay đổi  $\theta = x_1^{(k)}, x_2^{(k)}, \dots, x_n^{(k)}$  rồi tính  $a, b$  theo công thức trên, lấp bộ 4 số vào hàm lỗi  $J$ , ghi nhận lỗi và cuối cùng chọn bộ 4 có lỗi thấp nhất.

## 2.4. Đặc trưng Haar-like

Đặc trưng Haar-like được tính bằng tổng cường độ pixels vùng trắng trừ tổng cường độ pixels vùng đen. Các hình chữ nhật ở Hình 3 được trượt ở các vị trí khác nhau với các kích thước khác nhau, thay đổi trên 3 kênh màu trên vùng cần lấy đặc trưng và ta thu được một vector đặc trưng của vùng.



Hình 3. Đặc trưng Haar-like

## 3. XÂY DỰNG THỰC NGHIỆM

**Bài toán:** Cho biết vị trí vật cần theo vết ở tọa độ  $(x_1, y_1)$  và có kích thước  $(\omega_1, h_1)$ . Lọc trạng thái  $(x_k, y_k, \omega_k, h_k)$  của vật ở những khung ảnh tiếp theo.

### 3.1. Xây dựng particle filter

Mô hình chuyển trạng thái ẩn: Nếu trạng thái của vật ở thời điểm thứ  $k - 1$  là  $(x_{k-1}, y_{k-1}, \omega_{k-1}, h_{k-1})$  thì chúng ta hoàn toàn không biết về vị trí và kích thước vật ở thời điểm  $k$ . Vật có thể xuất hiện trên, dưới, trái, phải,.. so với vị trí cũ, vật có thể tăng giảm chiều dài chiều rộng so với trạng thái cũ,.. Do đó, hàm chuyển trạng thái phải đảm bảo những tính chất tổng quát này.

Tại thời điểm ảnh thứ 2, tác giả dùng particle filter để ước lượng trạng thái mới của vật bằng cách chọn hàm mật độ quan trọng trùng với hàm chuyển trạng thái. Ngoài ra, trong quá trình thực nghiệm tác giả nhận thấy nếu xem  $w$  và  $h$  độc lập nhau thì quá trình theo vết không ổn định. Do đó, xem  $h$  phụ thuộc hoàn toàn vào  $w$ , nghĩa là tỷ lệ  $h/w$  luôn bảo toàn và bằng tỷ lệ  $\eta = h/w$  của khung hình chữ nhật chứa vật ở ảnh ban đầu.

**Thuật toán 1:** Particle filter cho quá trình ngẫu nhiên  $(x_n, y_n, \omega_n, h_n)$

**Input:** Bộ mẫu particles pf (dựa vào **thuật toán 5**), ảnh thứ  $k$  (trong đó  $k$  bắt đầu từ ảnh thứ 2)

**Output:** Bộ particles mới đại diện cho  $p((x, y, \omega, h) \mid \text{ảnh } k)$ , ước lượng trạng thái của vật ở ảnh thứ  $k$ .

Bước 1: Gán  $\epsilon = 10^{-20}$

Bước 2:

for  $i=1$  to  $N_s$  do (với  $(x_i, y_i, \omega_i, h_i)$  là particle thứ  $i$ )

Lấy  $x_{new} \sim x_i + N(0, \delta_x^2)$

Lấy  $y_{new} \sim y_i + N(0, \delta_y^2)$

Lấy  $\omega_{new} \sim \omega_i + N(0, \delta_\omega^2)$

Tính  $h_{new} = \eta * \omega_{new}$

Tính  $likelihood = p(\text{ảnh } 2 \mid (x_{new}, y_{new}, \omega_{new}, h_{new}))$  theo **thuật toán 2**

Cập nhật trọng số cho particle thứ  $i$ :  $weight_i = weight_i * likelihood$

endfor

Bước 3:

Tính tổng  $sw = \sum_{j=1}^{N_s} weight_j + \epsilon$

for  $i=1$  to  $N_s$  do

Chuẩn hóa trọng số  $weight_i = \frac{weight_i + \epsilon}{sw}$

endfor

Bước 4: Tính  $N_{eff} = \frac{1}{\sum_{j=1}^{N_s} weight_j^2}$

Bước 5:

if  $N_{eff} < N_s/2$  then

$\{(x_i, y_i, \omega_i, h_i), weight_i\}_{i=1}^{N_s} = RESAMPLE(\{(x_i, y_i, \omega_i, h_i), weight_i\}_{i=1}^{N_s})$  sử dụng bootstrap

endif

Bước 6: Ước lượng trạng thái của vật ở ảnh thứ  $k$  bằng việc tính trung bình của bộ particles mới

$$Trạng\ thái\ ước\ lượng = \sum_{i=1}^{N_s} weight_i * particle[i]$$

**Thuật toán 2:** Tính likelihood của ảnh trên giả thuyết về trạng thái của vật

**Input:** Ảnh quan sát, giả thuyết  $(x, y, \omega, h)$ , vector  $HOG_1$  của vật ở ảnh thứ nhất, phân loại mạnh  $F$ .

**Output:** Likelihood  $p(\text{ảnh quan sát} | \text{giả thuyết } (x, y, \omega, h))$

Bước 1: Trích vùng ảnh trong hình chữ nhật  $(x, y, \omega, h)$  ra,  $h = crop(\text{ảnh}, (x, y, \omega, h))$

Bước 2: Resize ảnh  $h$  về kích thước của vật ở ảnh đầu tiên

Bước 3:

Trích đặc trưng HOG trên  $h$ ,  $HOG_2 = getHOG(h)$

Trích đặc trưng Haar-like trên  $h$ ,  $Haar_2 = getHaarlike(h)$

Bước 4: Tính điểm phân loại  $conf = F(Haar_2)$  theo **thuật toán 3**

Bước 5: Tính likelihood =  $\exp(\alpha * conf - \gamma ||HOG_1 - HOG_2||^2)$

**Thuật toán 3:** Tính điểm phân loại

**Input:** Phân loại mạnh  $F$  là ma trận  $4 \times s$  (tìm phân loại mạnh dựa vào **thuật toán 4**), vector đặc trưng Haar-like  $x$  của vùng cần tính điểm

**Output:** Điểm phân loại  $conf$

Bước 1: Gán điểm  $conf = 0$

Bước 2:

for  $i=1$  to  $s$  do

Lấy  $a, b, \theta, k$  là phân loại yếu thứ  $i$

Cập nhật  $conf = conf + a \times (x[k] > \theta) + b$

endfor

**Thuật toán 4:** Gentle Adaboost

**Input:** Tập huấn luyện  $(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)$  với  $y_i \in \{1, -1\}$  là nhãn của  $x_i$

**Output:** Hàm phân loại mạnh  $F$

Bước 1: Khởi tạo hệ số  $\omega_1 = \omega_2 = \dots = \omega_N = \frac{1}{N}$  cho tập huấn luyện

Bước 2:

for t=1 to s do

for j=1 to m do

Huấn luyện phân loại yếu  $h_j$ , tức là tính  $(a, k, \theta, b)$  theo công thức

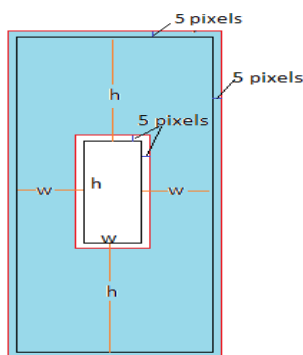
endfor

Chọn phân loại yếu có lỗi thấp nhất, đặt là  $f_t$

Cập nhật phân loại mạnh  $F(x) = F(x) + f_t(x)$

Cập nhật trọng số  $\omega_i = \omega_i \times \exp(-y_i f_t(x_i))$

endfor



Hình 4. Phần giới hạn cho việc lấy mẫu âm.

**Thuật toán 5:** Lấy mẫu để huấn luyện gentle adaboost

**Input:** Ảnh tại thời điểm ban đầu, vị trí và kích thước vật  $(x_0, y_0, \omega_0, h_0)$

**Output:** Bộ dữ liệu  $D$  kích thước  $N_+ + N_-$ , gồm  $N_+$  mẫu dương,  $N_-$  mẫu âm.

Bước 1: Khởi tạo một ngăn xếp features để chứa các vector đặc trưng.

Bước 2:

Lấy đặc trưng Haarlike của vùng  $(x_0, y_0, \omega_0, h_0)$ ,  $v = \text{getHaarlike}(\text{vùng}(x_0, y_0, \omega_0, h_0))$

Đưa  $v$  vào ngăn xếp *features*, *features.push(v)*

Bước 3:

for i=1 to  $N_+ - 1$  do

Lấy ngẫu nhiên vùng chữ nhật  $S$  từ ảnh với vị trí  $\pm 5$  pixels so với vật

Lấy vector đặc trưng Haarlike trên vùng  $S$ ,  $v = \text{getHaarlike}(S)$

Đưa vector  $v$  vào ngăn xếp *features*, *features.push(v)*

endfor

Bước 4:

for i=1 to  $N_- - 1$  do

Lấy ngẫu nhiên vùng chữ nhật  $S$  thuộc phần màu xanh Hình 4

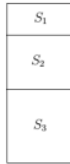


```

    Lấy vector đặc trưng Haarlike trên vùng  $S$ ,  $v = \text{getHaarlike}(S)$ 
    Đưa vector  $v$  vào ngăn xếp  $features$ ,  $features.push(v)$ 
endfor
    
```

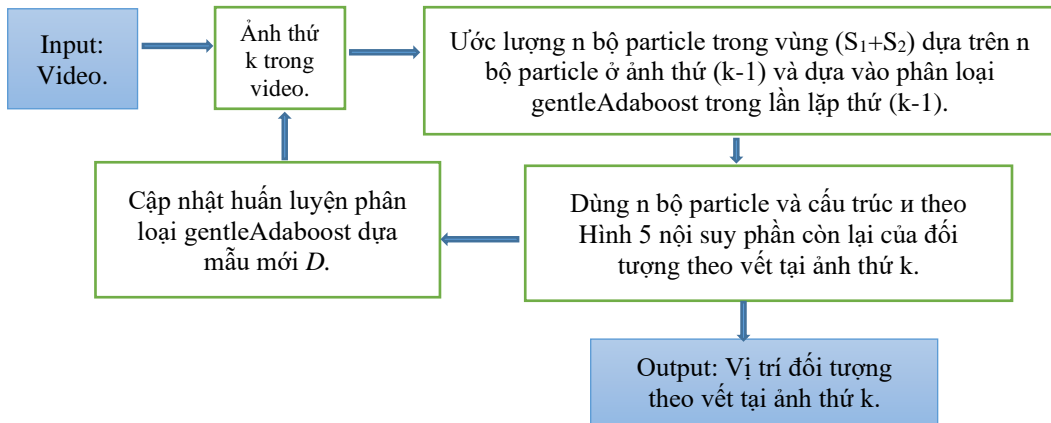
### 3.2. Thuật toán theo vết phần chứa nhiều thông tin (InfoPart)

Thay vì phải theo vết cả đối tượng dựa trên mô hình particle filter, nhóm tác giả đề xuất theo vết một phần đối tượng chứa nhiều thông tin và ít thay đổi dựa trên mô hình particle filter. Sau đó dựa vào kết quả theo vết một phần đối tượng nội suy ra toàn bộ đối tượng. Ví dụ, trường hợp đối tượng là người đi bộ thì 5/8 phần trên (gồm phần đầu và thân) là phần ít biến đổi theo thời gian và “đặc” hơn phần chân.



Hình 5. Cấu trúc  $n$  phân hoạch đối tượng gồm  $S_1$  là đầu,  $S_2$  là thân và  $S_3$  là chân.  
Vùng  $(S_1+S_2)$  là phần lợi thông tin trong đối tượng.

Sơ đồ hệ thống bài toán theo vết đối tượng:



Hình 6. Sơ đồ hệ thống bài toán theo vết đối tượng

**Thuật toán 6** theo vết đối tượng với phần nhiều thông tin dựa trên mô hình particle filter (InfoPart):

**Bước 1:** Khởi tạo  $n$  bộ particles  $pf_1, pf_2, \dots, pf_n$  trong vùng  $(S_1+S_2)$  hình 5 cho phép theo vết phần lợi thông tin trong đối tượng tại ảnh thứ 1.

**Bước 2:**

Lấy mẫu  $D$  cho vùng  $(S_1+S_2)$  tại ảnh thứ 1 theo **thuật toán 5**.

Huấn luyện phân loại mạnh  $F = \text{gentleAdaboost}(D)$  theo **thuật toán 3**.

**Bước 3:**

while video chưa kết thúc do

  Beginwhile

    Nhận ảnh thứ  $k$  quan sát đối tượng theo vết

Dùng  $n$  bộ particle ước lượng trạng thái vùng  $(S_1+S_2)$  tại ảnh thứ  $k$  theo **thuật toán 1**

Dùng phần lợi thông tin vùng  $(S_1+S_2)$  và cấu trúc  $n$  theo Hình 5 nội suy phần còn lại của đối tượng theo vết tại ảnh thứ  $k$  (*Nghĩa là ước lượng trạng thái của đối tượng theo vết ở ảnh thứ  $k$  dựa trên  $n$  bộ particle*).

Lấy mẫu mới  $D^*$  dựa trên phần lợi thông tin trong đối tượng theo vết tại ảnh thứ  $k$ .

Cập nhật phân loại mạnh  $F$ .

endwhile

#### 4. KẾT QUẢ THỰC NGHIỆM

Môi trường cài đặt: Tác giả thực nghiệm trên máy tính sử dụng hệ điều hành Windows 10 Pro bản 64 bit, RAM 8 GB, Chip Intel Core (TM) 5i-3210M CPU @ 2.5GHz. Ngôn ngữ lập trình Matlab phiên bản R2016a.

##### 4.1. Bộ dữ liệu

Năm 2013, nhóm tác giả Yi-Wu, Jongwoo Lim, Ming-Hsuan Yang [15] đã tổng hợp nhiều nguồn video liên quan đến theo vết và đã tiến hành tạo groundtruth cho các video này để tạo thành bộ dữ liệu TB-100.

Vì TB-100 là bộ dữ liệu tổng hợp từ nhiều nguồn nên ngữ cảnh của các video cũng rất khác nhau và đa dạng về thuộc tính như: loại vật cần theo vết, video màu hoặc trắng đen, camera tĩnh hoặc động,...

Các thử thách trong bộ dữ liệu bao gồm:

IV – Độ sáng của đối tượng thay đổi đáng kể.

SV – Tỷ lệ của hình chữ nhật chứa vật ảnh thứ nhất với ảnh hiện tại vượt ra khỏi khoảng  $\left[1/t_s, t_s\right], t_s > 1 (t_s = 2)$ .

OCC – Đối tượng bị che khuất một phần hoặc toàn phần.

DEF – Đối tượng không đặc biến đổi hình dạng.

MB – Đối tượng bị nhòe do chuyển động của camera.

FM – Chuyển động của groundtruth lớn hơn  $tm$  pixels ( $tm = 20$ ).

IPR – Đối tượng xoay trong miền ảnh.

OPR – Đối tượng ra khỏi miền ảnh.

OV – Một phần của đối tượng ra khỏi miền ảnh.

BC – Nền gần đối tượng có màu sắc hoặc đường nét giống đối tượng.

LR – Số lượng pixels trong hình chữ nhật chứa vật (xét groundtruth) nhỏ hơn  $tr$  ( $tr = 400$ ).

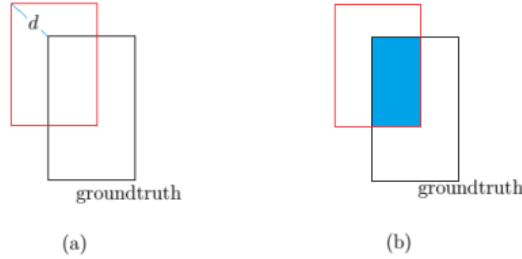
##### 4.2. Phương thức đánh giá

Nghiên cứu của Yi Wu *et al.* cung cấp các tiêu chuẩn để đánh giá thuật toán theo vết [15].

Phương thức 1 ( $R_1$ ): Đánh giá dựa trên khoảng cách Euclid (precision plot): Đo khoảng cách Euclid  $d$  từ tâm ước lượng của thuật toán đến tâm thực sự của vật (groundtruth), nếu  $d$  nhỏ hơn hoặc bằng một ngưỡng  $t_0$  ( $t_0 = 20$ ) thì được xem là thành công (Hình 7a).

Phương thức 2 ( $R_2$ ): Đánh giá dựa trên mức độ trùng nhau. Số điểm trùng lặp được định nghĩa  $S = \frac{|r_t \cap r_a|}{|r_t \cup r_a|}$ , trong đó:  $r_t$  là hình chữ nhật bao vật do thuật toán ước lượng và  $r_a$  là hình chữ nhật groundtruth (Hình 7b).

$$\text{Tính tỷ lệ } R_1, R_2 = \frac{\text{số ảnh thành công}}{\text{tổng số ảnh}}$$



Hình 7. (a) Đo khoảng cách Euclid; (b) đo mức độ trùng nhau

### 4.3. Kết quả đánh giá

Kết quả theo vết người với camera không dao động nhiều và bảo toàn tương đối các tỷ lệ trên cơ thể người và người không quá nhỏ. Tác giả sử dụng thuật toán InfoPart so sánh với thuật toán GradNet [16].

Bảng 1. Kết quả thuật toán GradNet và InfoPart

Video	Thuộc tính	GradNet %		InfoPart %	
		R1	R2	R1	R2
Crossing	SV, DEF, FM, OPR, BC	100	100	100	98,33
Dancer	SV, DEF, IPR, OPR	92,44	98,22	95,11	88,89
Dancer2	DEF	100	100	77,33	59,01
David3	OCC, DEF, OPR, BC	100	100	73,02	81,35
Human8	IV, SV, DEF	8,59	7,03	100	89,06
Walking	SV, OCC, DEF	100	99,27	100	98,54
Walking2	SV, OCC, LR	100	100	100	100
		85,86	86,36	92,21	87,88

## 5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

Thuật toán InfoPart lấy phần có lợi thông tin 5/8 đối tượng người (gồm đầu và thân) cần theo vết cho kết quả tốt hơn thuật toán GradNet. Cụ thể, cho thấy độ chính xác trung bình của thuật toán InfoPart ( $R_1 = 92,21\%$ ,  $R_2 = 87,88\%$ ) lớn hơn so với thuật toán GradNet ( $R_1 = 85,86\%$ ,  $R_2 = 86,36\%$ ).

Ngoài ra, theo vết phần có nhiều thông tin rồi nội suy phần còn lại sẽ giảm thời gian tính toán so với thuật toán cơ sở particle filter ban đầu theo vết trên toàn bộ đối tượng.

Đối với lớp video theo vết người, camera tương đối ổn định và phép quay góc bảo toàn tương đối tỷ lệ cơ thể người, tác giả theo vết phần có nhiều thông tin nhất rồi suy ra cả cơ thể. Kết quả thực nghiệm của các thuật toán nêu trên cho từng video là chênh lệch nhau rất lớn, cao nhất là 100% và thấp nhất là 59%. Kết quả này cũng dễ hiểu vì bộ dữ liệu TB-100 là tổng

hợp của rất nhiều bộ dữ liệu khác, rất đa dạng và nhiều thử thách. Nhìn chung, kết quả đạt ở mức tốt.

Hướng phát triển là thay đổi mô hình quan sát. Nhóm tác giả nhận thấy quá trình huấn luyện gentle adaboost rất tốn thời gian và khiến cho hệ thống chưa thể hoạt động theo thời gian thực được. Các thuật toán sử dụng lớp bộ lọc tương quan lại có ưu điểm là tốc độ xử lý rất nhanh và chính xác, nếu tích hợp được lớp bộ lọc tương quan này vào mô hình quan sát thì sẽ rút ngắn đáng kể thời gian thực thi.

**Lời cảm ơn:** Nghiên cứu này do Trường Đại học Công nghiệp Thực phẩm TP.HCM bảo trợ và cấp kinh phí theo Hợp đồng số 50/HĐ-DCT ngày 03 tháng 9 năm 2019.

## TÀI LIỆU THAM KHẢO

1. David S. Bolme, Bruce A. Draper, J. Ross Beveridge - Average of synthetic exact filters, IEEE Conference on Computer Vision and Pattern Recognition (2009).
2. Joao F. Henriques, Rui Caseiro, Pedro Martins, and Jorge Batista - Exploiting the circulant structure of tracking-by-detection with kernels, European Conference on Computer Vision (2012) 702-715.
3. Joao F. Henriques, Rui Caseiro, Pedro Martins, and Jorge Batista - High-speed tracking with kernelized correlation filters, IEEE Transactions on Pattern Analysis and Machine Intelligence (2015) 583-596.
4. David S. Bolme, J. Ross Beveridge, Bruce A. Draper, Yui Man Lui - Visual object tracking using adaptive correlation filters, IEEE Computer Society Conference on Computer Vision and Pattern Recognition (2010) 2544-2550.
5. Shai Avidan - Ensemble tracking, IEEE Transactions on Pattern Analysis and Machine Intelligence (2007).
6. Freund Y. - An adaptive version of the boost by majority algorithm, Machine Learning **43** (2001) 293-318.
7. Zhe Chen - Bayesian filtering: from kalman filters to particle filters, and beyond, Statistics: A Journal of Theoretical and Applied Statistics (2003).
8. Arnaud Doucet, Adam M. Johansen - A tutorial on particle filtering and smoothing: Fifteen years later (2008).
9. Daniel Jurafsky, James H. Martin - Chapter 9: Hidden Markov models, Speech and Language Processing (2017).
10. Dominik A. Klein, Dirk Schulz, Simone Frintrop, and Armin B. Cremers - Adaptive real-time video-tracking for arbitrary objects, IEEE/RSJ International Conference on Intelligent Robots and Systems (2010) 772-777.
11. Perez P., Hue C., Varmaak J., Gangnet M. - Color-based probabilistic tracking, European Conference on Computer Vision (2002) 661-675.
12. Changjiang Yang, Ramani Duraiswami, Larry Davis - Fast multiple object tracking via a hierarchical particle filter, 10<sup>th</sup> IEEE International Conference on Computer Vision (2005) 212-219.
13. Olov Samuelsson - Video tracking algorithm for unmanned aerial vehicle Surveillance, Stockholm - Sweden (2012).
14. Sanjeev Arulampalam, Simson Maskell, Neil Gordon, Tim Clapp - A tutorial on particle filters for online no-linear/non-gaussian bayesian tracking, IEEE Transactions on Signal Processing **50** (2) (2002) 174-188.

15. Yi Wu, Jongwoo Lim, Ming-Hsuan Yang - Online object tracking: A benchmark, *Computer Vision and Pattern Recognition* (2013).
16. Peixia Li, Boyu Chen, Wanli Ouyang, Dong Wang, Xiaoyun Yang, Huchuan Lu - GradNet: Gradient-guided network for visual object tracking, *IEEE International Conference on Computer Vision (ICCV)* (2019).

## **ABSTRACT**

### **TRACKING MOVING OBJECTS IN VIDEO BASED ON BENEFIT PART**

Ngo Duong Ha, Tran Nhu Y, Le Huu Ha,  
Nguyen Phuong Hac, Nguyen Van Tung\*  
*Ho Chi Minh City University of Food Industry*  
\*Email: *tungnv@hufi.edu.vn*

Tracking object in video is important problem of computer vision. Tracking the movement of objects that can be used in security systems, automatic observation, integrated into robots unmanned aerial vehicles of the military, etc. This paper presents an approach that describes the problem in terms of filtering a hidden Markov model using particle filters to filter the random process and combining the computation of confidence scores from gentle adaboost. We suggest tracking the more informative parts (head, body) and omitting the foot part which is the more oscillating part. An InfoPart algorithm is proposes to track the benefit part (head, body) without leg that is more oscillatory. Experimental results show that the average accuracy of InfoPart algorithm is greater than GradNet algorithm.

*Keywords:* Hidden Markov model, gentle adaboost, particle filter, Haar like.