

MỘT CÁCH TIẾP CẬN MỚI GIẢI QUYẾT VIỆC CHUYỂN TIẾP CÁC CAMERA TRONG HỆ THỐNG GIÁM SÁT TỰ ĐỘNG

Ngô Đức Vinh^{1,*}, Đỗ Năng Toàn²

¹Trường ĐH Công nghiệp Hà Nội, Nhân, Từ Liêm, Hà Nội

²Viện Công nghệ thông tin, Viện HLKHCNVN, 18 Hoàng Quốc Việt, Cầu Giấy, Hà Nội

*Email: ngoducvinh@gmail.com

Đến Tòa soạn: 11/07/2012; Chấp nhận đăng: 20/5/2013

TÓM TẮT

Bài báo này đề xuất việc tính toán va chạm của đối tượng với một đường ranh giới ảo (virtual line). Đường ranh giới ảo thực chất là một đường thẳng được vẽ lên trên mỗi khung hình trong quá trình thao tác với dữ liệu video. Trên cơ sở việc tính toán này chúng tôi đưa ra việc đánh giá đối tượng là xuất hiện hay biến mất trong một vùng nằm trong thị trường của camera quan sát, nhờ đó giải quyết được việc chuyển tiếp giữa các camera trong một hệ thống giám sát gồm nhiều camera.

Từ khóa: đường ranh giới ảo, chuyển tiếp camera, giám sát tự động, thị trường camera.

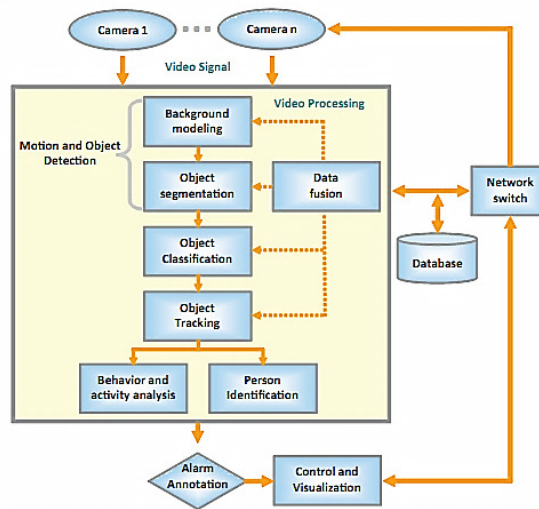
1. GIỚI THIỆU

Trong nhiều hệ thống giám sát tự động, một camera là không đủ để quan sát hết bởi nó bị giới hạn về độ phân giải, thị trường quan sát. Do vậy để giám sát một khu vực rộng lớn cần phải có nhiều camera. Một vấn đề cần được giải quyết trong hệ thống giám sát có nhiều camera đó là sự xuất hiện hoặc biến mất của đối tượng từ camera này sang camera khác, hay còn gọi là sự chuyển tiếp camera.

Nhìn một cách tổng quan, đối với một hệ thống giám sát tự động, đầu vào của hệ thống sẽ là hình ảnh thu được tại các điểm quan sát, đầu ra của hệ thống sẽ là các thông tin về chuyển động, hành vi, lớp... của các đối tượng được giám sát. Mô hình tổng quát của hệ thống giám sát tự động [1] được biểu diễn ở hình 1.

Có thể tổ chức một hệ thống giám sát nhiều camera theo hai hướng: Hệ thống các camera có vùng quan sát chồng chéo (overlapping) hoặc vùng quan sát không chồng chéo (non-overlapping). Với mỗi hệ thống theo dõi bằng nhiều camera đều có những thách thức cần giải quyết, và thông thường các nghiên cứu đều giả định giải quyết trong một điều kiện cụ thể [2].

Khi quan sát từ các camera thì các quan sát của một đối tượng thường cách xa nhau theo thời gian và không gian. Sự xuất hiện của một đối tượng ở camera này lại có thể rất khác nhau so với camera khác (hình 2), có thể là do khác nhau về sự chiếu sáng, tư thế và các thuộc tính của camera [2].



Hình 1. Sơ đồ một hệ thống giám sát tự động.

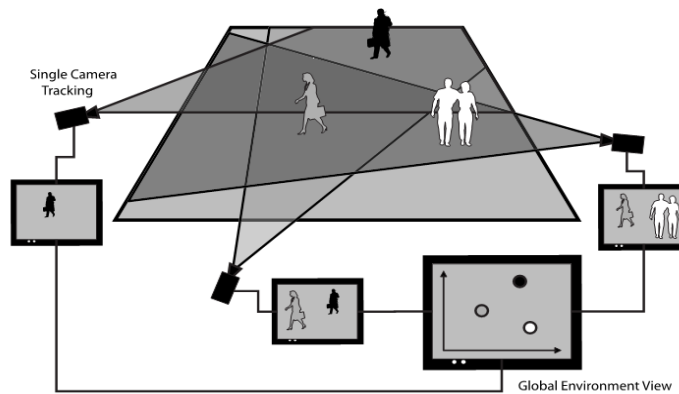


Fig. 1.3 A Distributed surveillance system. Inputs from each camera are integrated at the server level to determine a central view of the environment.

Hình 2. Hệ thống giám sát với nhiều camera có vùng quan sát bị chồng chéo.

Đã nhiều công trình nghiên cứu với giả định trên hệ thống giám sát nhiều camera có vùng quan sát chồng chéo. Một số hướng tiếp cận, như sử dụng các camera có hiệu chỉnh (các camera có thể thay đổi góc mở ống kính, thu, phóng, xoay, điều chỉnh góc nghiêng...). Tuy nhiên hướng tiếp cận sử dụng camera hiệu chỉnh được đánh giá là tốn kém và phụ thuộc nhiều vào thiết bị, do vậy phần lớn các nghiên cứu đều tập trung và hướng giải quyết không sử dụng hiệu chỉnh camera [2] mà thay bằng cách thiết lập sự tương ứng của các đối tượng ở camera này với các camera khác, thực chất là đối sánh các đối tượng nằm trong vùng giao nhau giữa các thị trường của các camera [2]. Có nhiều kỹ thuật đã được áp dụng: như sử dụng lược đồ màu, mô hình xác suất [3], mạng Beysian[4], mô hình Markov, hình học pipolar [4], đường FOV [5, 6, 7]. Tuy nhiên, việc phụ thuộc nhiều vào hình học epipolar làm cho cách tiếp cận không gần với thực tế được đặt ra đối với hệ thống giám sát có nhiều camera là cần phải theo dõi được các đối tượng khi chúng di chuyển từ camera này sang camera khác hay còn gọi là sự chuyển tiếp camera. Trong bài báo này, chúng tôi đề xuất một kỹ thuật nhằm giải quyết việc xuất hiện hoặc biến mất của đối tượng từ camera này sang camera khác thông qua kỹ thuật virtual line, với ý tưởng thực

hiện phân chia vùng thị trường của hệ thống camera thành các vùng con không giao cắt, ranh giới các vùng được xác định bằng kỹ thuật virtual line, thông qua đó việc chuyển giao nhân đối tượng từ camera này sang camera khác sẽ trở nên đơn giản hơn.

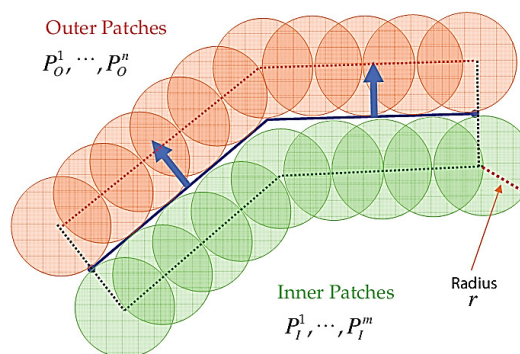
Phần còn lại của bài báo được trình bày như sau: Phần 2: Trình bày về khái niệm đường virtual line, thị trường camera và của một hệ thống camera, khái niệm sự xuất hiện, biến mất của đối tượng. Phần 3: Đề xuất một mô hình thông qua việc tính toán va chạm của đối tượng với đường ranh giới ảo để giải quyết bài toán chuyển tiếp camera trong hệ thống giám sát tự động có sử dụng nhiều camera. Phần tiếp theo là thử nghiệm và cuối cùng là kết luận về kỹ thuật đề xuất.

2. MỘT SỐ KHÁI NIỆM CƠ BẢN

2.1. Đường ranh giới ảo - Virtual line

Năm 1996, trong quá trình tìm giải pháp cho bài toán giám sát, đếm số người qua lại của hệ thống camera giám sát, Segen và Pingali [8] đã đề xuất một cách xác định hướng của người chuyển động khi họ đi qua một đường ranh giới ảo được gọi là virtual line. Virtual line thực chất là một đường thẳng được vẽ lên trên mỗi khung hình trong quá trình thao tác với dữ liệu video.

Bài toán thông thường được đặt ra đối với virtual line là xác định khi nào đối tượng vượt qua nó, mở rộng hơn là khi nào đối tượng đi vào một vùng hoặc đi ra khỏi vùng được giới hạn bởi virtual line. LeFloch [9] đã sử dụng đường ranh giới ảo vào việc đếm số người vào ra trong một tòa nhà. Liu Anan và cộng sự [10] đề xuất sử dụng nhóm các đường virtual line trong việc phát hiện xe tải trong giao thông. Trong một công trình gần đây, Y. Zhai và cộng sự [11] đề xuất kỹ thuật sử dụng đường tripwire cùng với các miếng vá hình tròn trong việc phát hiện các đối tượng đi vào một khu vực được giám sát. Mỗi miếng vá là một cặp 3 giá trị: tâm, bán kính và vector định hướng. Các miếng vá được chia thành hai nhóm, nhóm bên trong và nhóm bên ngoài, theo đó nếu đối tượng khi đi qua tripwire từ nhóm bên trong ra bên ngoài thì cùng với hướng vector định hướng (hình 3).



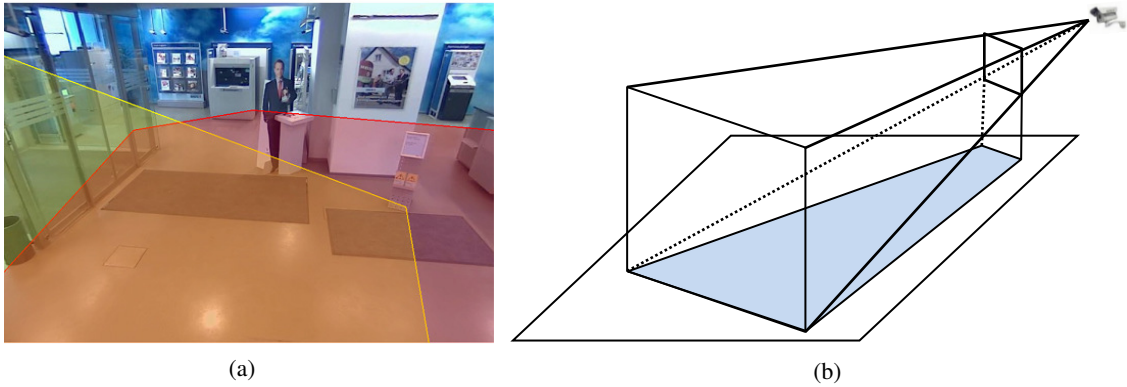
Hình 3. TripWire cùng với các miếng vá.

2.2. Thị trường camera

Là khoảng không gian mà camera có thể nhận biết, thu nhận được hình ảnh, nó được xác định bởi các thông số của camera (kích thước ống kính, góc nghiêng, độ mở ống kính...) (hình 4a).

Trong thực tế, khu vực quan sát của camera bị ảnh hưởng rất nhiều bởi các yếu tố như bị che chắn bởi các vật cản... do vậy với thị trường quan sát mỗi camera đều gắn với một đa giác thể hiện được khu vực quan sát thực sự của nó (hình 4b).

Gọi F_i là thị trường và P_i là vùng quan sát được của một camera, trong đó $P_i = \{V_1, V_2, \dots, V_l\}$, với $V_k (x_k, y_k)$ là đỉnh của đa giác có tọa độ (x_k, y_k) .



Hình 4. Vùng quan sát và mô hình thị trường của camera
 (a). Vùng quan sát thực tế của camera; (b). Mô hình thị trường của camera.

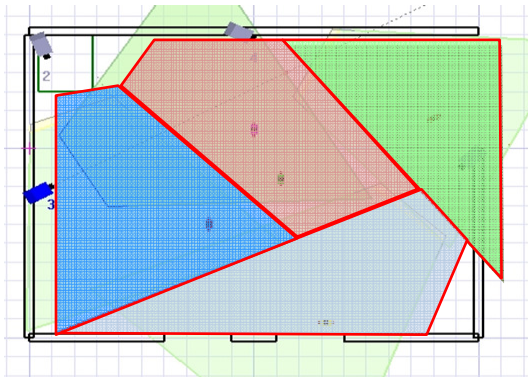
Thị trường hệ thống camera: Một hệ thống giám sát gồm n camera thì thị trường quan sát của hệ thống $F = \{F_1, F_2, \dots, F_n\}$ là hợp tất cả các thị trường của từng camera.

Vùng quan sát của toàn bộ hệ thống nhiều camera có thể được chia thành tập các vùng quan sát của từng camera, vùng quan sát của mỗi một camera là một hình đa giác không tự cắt gồm n đỉnh được giới hạn bởi các đường ranh giới ảo, tập các đa giác này không giao nhau (hình 5).

Như vậy, với một hệ thống giám sát có n camera:

$$F = \bigcup_{i=1}^n F_i; P = \bigcup_{i=1}^n P_i; \bigcup_{i=1}^n F_i = \bigcup_{i=1}^n P_i$$

trong đó: $P_i \cap P_j = \emptyset (\forall i, j \in 1..n)$ và $F_i \cap F_j \neq \emptyset (\forall i, j \in 1..n)$.



Hình 5. Mô hình thị trường của hệ thống camera được chiếu xuống mặt phẳng 2D.

2.3. Sự xuất hiện hoặc biến mất đối tượng

Sau khi áp dụng kĩ thuật phân đoạn đối tượng chuyển động, mỗi đối tượng sẽ được gán một nhãn k (O_k là đối tượng có nhãn k).

Gọi $O_i^T = \{O_1, O_2, \dots, O_n\}$ và $O_i^{T+1} = \{O_1, O_2, \dots, O_m\}$ là tập các đối tượng (n, m: số đối tượng) đang nằm trong vùng quan sát bởi một camera i ở khung hình T và T+1.

Hiện tượng biến mất đối tượng là hiện tượng đối tượng O_k , xuất hiện trong khung hình T nhưng không xuất hiện trong khung hình T+1.

$$\text{Tập các đối tượng biến mất khỏi camera i: } O_i^D = \{O_i^T \setminus O_i^T \cap O_i^{T+1}\}$$

Hiện tượng xuất hiện đối tượng là hiện tượng đối tượng O_k , xuất hiện trong khung hình T + 1 nhưng không xuất hiện trong khung hình T.

Tập các đối tượng xuất hiện trong vùng quan sát của camera i : $O_i^A = \{O_i^{T+1} \setminus O_i^T \cap O_i^{T+1}\}$

Xét các trường hợp sau:

- Nếu $m < n$: khi đó có $n-m$ đối tượng bị biến mất khỏi vùng quan sát của camera i .
- Nếu $m > n$: khi đó có $m-n$ đối tượng mới xuất hiện trong vùng quan sát của camera i .
- Nếu $m = n$:
 - + Nếu $O_i^T = \{O_i^T \cap O_i^{T+1}\}$ khi đó không có đối tượng nào xuất hiện hoặc biến mất.
 - + Nếu $O_i^T \neq \{O_i^T \cap O_i^{T+1}\}$ khi đó số đối tượng xuất hiện bằng số đối tượng biến mất và $O_i^D = \{O_i^T \setminus O_i^T \cap O_i^{T+1}\}$, $O_i^A = \{O_i^{T+1} \setminus O_i^T \cap O_i^{T+1}\}$. Nói cách khác, danh sách đối tượng biến mất là các đối tượng có nhãn thuộc O_i^{T+1} nhưng không thuộc O_i^T và ngược lại.

3. TÍNH TOÁN VA CHẠM CỦA ĐỐI TƯỢNG VỚI ĐƯỜNG RANH GIỚI ẢO

Để tính toán được va chạm của các đối tượng, ta tìm một hình bao “thích hợp” quanh nó, việc phát hiện va chạm bây giờ được đưa về bài toán phát hiện va chạm giữa các hình bao. Việc sử dụng hình bao gì cho đối tượng là tùy thuộc vào từng hệ thống, các hình bao thường dùng: hình elipsoid, hình hộp, hình cầu...

Đối với kỹ thuật sử dụng hình bao là hình hộp, có hai kỹ thuật khác nhau đó là sử dụng hộp bao có các cạnh song song với các trục tọa độ (Axis Aligned Bounding Boxes - AABBs) hoặc là hộp bao theo hướng của đối tượng (Oriented Bounding Boxes- OBBs) [12].

Để thuận lợi trong hệ thống giám sát theo dõi người chuyển động, chúng tôi chọn hướng tiếp cận sử dụng hình hộp bao theo hướng đối tượng (OBB).

3.1. Hình hộp bao theo hướng

Trong không gian \mathbb{R}^3 , hình hộp bao theo hướng OBB [13] gồm một tâm C_0 , ba vector $\vec{A}_0, \vec{A}_1, \vec{A}_2$ chỉ hướng của hình hộp và 3 hệ số độ dài tương ứng với kích thước của hình hộp là $a_0 > 0, a_1 > 0, a_2 > 0$. Khi đó, 8 đỉnh của hình hộp sẽ được xác định như sau:

$$C_0 + \sum_{i=0}^2 s_i a_i * \vec{A}_i \quad |s_i| = 1, i = 0, 1, 2.$$

3.2. Tính toán va chạm

Đường ranh giới ảo thực chất là một đường thẳng được vẽ lên trên mỗi khung hình trong quá trình thao tác với dữ liệu video. Trong không gian 3D, thì nó chính là một mặt phẳng đi qua đường ranh giới ảo.

Trong thực tế, tốc độ di chuyển của đối tượng và tốc độ thu nhận của camera không phù hợp, do vậy để tránh trường hợp khi đối tượng di chuyển nhanh qua đường ranh giới ảo mà camera chưa bắt kịp, chúng tôi đề xuất thay vì sử dụng một đường ranh giới ảo, ta sẽ dùng một nhóm 3 đường ranh giới ảo (hai đường mở rộng về hai phía của đường ban đầu với khoảng

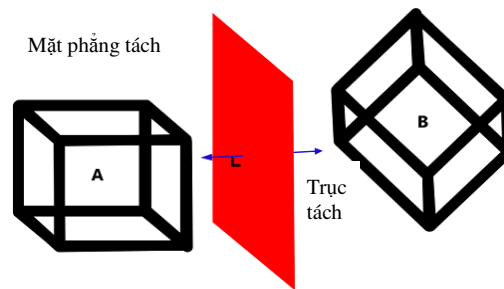
cách dx) như vậy trong không gian 3D thì nó trở thành một hình hộp. Chính vì vậy, việc tính toán va chạm sẽ thực hiện trên hai khối hình hộp.

Định lý 3.1[13]. Hai khối đa diện lồi không giao nhau nếu có thể tách rời được chúng bằng một mặt phẳng P thoả mãn một trong hai điều kiện sau:

- P song song với một mặt nào đó của một trong hai khối đa diện.
- Hoặc là P chứa một cạnh thuộc đa diện thứ nhất và một đỉnh thuộc đa diện thứ hai.

Từ định lý trên, ta rút ra nhận xét sau cho phép kiểm tra nhanh sự giao nhau của hai khối đa diện lồi: Điều kiện cần và đủ để kiểm tra hai khối đa diện lồi có giao nhau hay không là kiểm tra giao nhau giữa các hình chiếu của chúng lên đường thẳng vuông góc với mặt phẳng P ở trên, đường thẳng này được gọi là *trục tách*.

Ta thấy rằng các hộp bao OBBs là những khối đa diện lồi, bởi vậy ta hoàn toàn có thể áp dụng định lý trên để kiểm tra va chạm giữa chúng.



Hình 7. Mặt phẳng tách và trục tách.

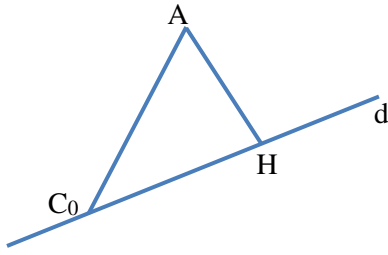
Cho hai hình bao OBBs xác định bởi các thông số $[C_0, \vec{A}_0, \vec{A}_1, \vec{A}_2, a_0, a_1, a_2]$ và $[C_1, \vec{B}_0, \vec{B}_1, \vec{B}_2, b_0, b_1, b_2]$. Tập ứng cử viên các trục tác chỉ tối đa là 15 trục sau:

- 3 trục chỉ hướng của hộp bao thứ nhất (\vec{A}_i).
- 3 trục chỉ hướng của hộp bao thứ hai (\vec{B}_j).
- 9 trục tạo bởi tích có hướng của một trục thuộc hộp bao thứ nhất và một trục thuộc hộp bao thứ hai ($\vec{A}_i \otimes \vec{B}_j$).

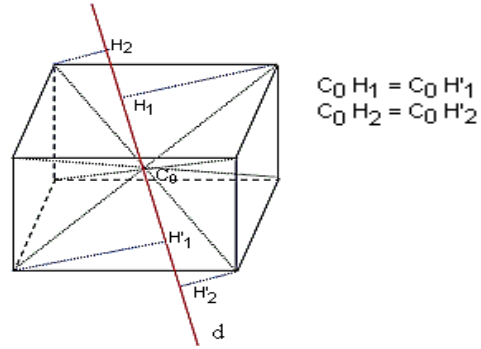
Mặt khác, ta biết rằng nếu một trục là trục tách thì khi tịnh tiến đến vị trí nào, nó vẫn là trục tách. Bởi vậy, không mất tính tổng quát ta sẽ gọi trục tách có vector chỉ phương là \vec{V} và đi qua tâm C_0 của hộp bao thứ nhất, do vậy nó có phương trình như sau: $d = C_0 + t * \vec{V}$. Trong đó, t là tham số. \vec{V} có thể là \vec{A}_i, \vec{B}_j hoặc $\vec{A}_i \otimes \vec{B}_j$ với $i, j = 0, 1, 2$.

Gọi A là một điểm bất kì, H là hình chiếu của A lên đường thẳng d (hình 8) khi đó độ dài đoạn thẳng C_0H xác định như sau:

$$L(A, d, C_0) = \frac{\overrightarrow{C_0A} * \vec{V}}{|\vec{V}|}$$



Hình 8. Hình chiếu A lên đường thẳng d với tâm chiếu C_0 .



Hình 9. Chiếu 8 đỉnh của hình hộp lên trục tách d.

Như vậy, khi chiếu 8 đỉnh của hộp bao thứ nhất lên trục tách d thì sẽ thu được 4 cặp đoạn thẳng có độ dài bằng nhau nằm về hai phía so với C_0 (hình 9), độ dài của mỗi đoạn thẳng được xác định như sau:

$$L(C_0 + \sum_{i=0}^2 s_i * a_i * \vec{A}_i, d, C_0) = \frac{\left| \left(\sum_{i=0}^2 s_i * a_i * \vec{A}_i \right) * \vec{V} \right|}{|\vec{V}|} \quad (1)$$

Như vậy, khoảng cách nhỏ nhất chứa 8 đoạn thẳng (1) sẽ có tâm có tâm là C_0 và bán kính r_0 được xác định như sau:

$$r_0 = \max \left\{ \frac{\left| \left(\sum_{i=0}^2 s_i * a_i * \vec{A}_i \right) * \vec{V} \right|}{|\vec{V}|} \right\} \text{ với mọi } |s_i| = 1$$

Đặt $R_0 = r_0 * |\vec{V}|$, ta có:

$$R_0 = \max \left\{ \begin{aligned} &|a_0 * \vec{A}_0 * \vec{V} + a_1 * \vec{A}_1 * \vec{V} + a_2 * \vec{A}_2 * \vec{V}|, |a_0 * \vec{A}_0 * \vec{V} + a_1 * \vec{A}_1 * \vec{V} - a_2 * \vec{A}_2 * \vec{V}|, \\ &|a_0 * \vec{A}_0 * \vec{V} - a_1 * \vec{A}_1 * \vec{V} + a_2 * \vec{A}_2 * \vec{V}|, |a_0 * \vec{A}_0 * \vec{V} - a_1 * \vec{A}_1 * \vec{V} - a_2 * \vec{A}_2 * \vec{V}|, \\ &|-a_0 * \vec{A}_0 * \vec{V} + a_1 * \vec{A}_1 * \vec{V} + a_2 * \vec{A}_2 * \vec{V}|, |-a_0 * \vec{A}_0 * \vec{V} + a_1 * \vec{A}_1 * \vec{V} - a_2 * \vec{A}_2 * \vec{V}|, \\ &|-a_0 * \vec{A}_0 * \vec{V} - a_1 * \vec{A}_1 * \vec{V} + a_2 * \vec{A}_2 * \vec{V}|, |-a_0 * \vec{A}_0 * \vec{V} - a_1 * \vec{A}_1 * \vec{V} - a_2 * \vec{A}_2 * \vec{V}| \end{aligned} \right\}$$

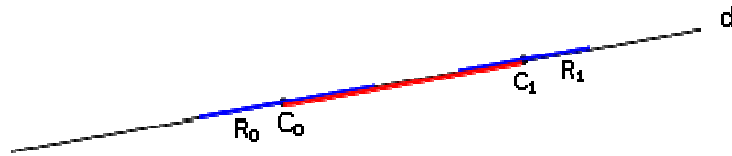
$$\Rightarrow R_0 = a_0 * |\vec{A}_0 * \vec{V}| + a_1 * |\vec{A}_1 * \vec{V}| + a_2 * |\vec{A}_2 * \vec{V}|$$

Tương tự, ta chiếu 8 đỉnh của hộp bao thứ hai lên d với góc C_1 , sẽ thu được 4 cặp đoạn thẳng có độ dài bằng nhau nằm về hai phía so với C_1 , như sau:

$$L(C_1 + \sum_{i=0}^2 s_i * b_i * \vec{B}_i, d, C_1) = \left| \frac{\left(\sum_{i=0}^2 s_i * b_i * \vec{B}_i \right) * \vec{V}}{|\vec{V}|} \right| \quad (2)$$

Chú ý rằng, 8 đoạn thẳng này được nhóm thành 4 cặp đối xứng nhau qua C_1 . Do vậy, khoảng cách nhỏ nhất chứa 8 đoạn thẳng (2) sẽ có tâm là C_1 và bán kính r_1 được xác định như sau:

$$r_1 = \max \left\{ \left| \frac{\left(\sum_{i=0}^2 s_i * b_i * \vec{B}_i \right) * \vec{V}}{|\vec{V}|} \right| \right\} \text{ với mọi } |s_i| = 1$$



Hình 10. Kết quả chiếu 2 hình hộp lên trục cô tách d.

Đặt $R_1 = r_1 * |\vec{V}|$, tương tự như trên ta có: $R_1 = b_0 * |\vec{B}_0 * \vec{V}| + b_1 * |\vec{B}_1 * \vec{V}| + b_2 * |\vec{B}_2 * \vec{V}|$

Hai đoạn thẳng trên sẽ không giao nhau nếu:

$$C_0 C_1 > r_0 + r_1 \Leftrightarrow |\vec{V}| * C_0 C_1 > |\vec{V}| * r_0 + |\vec{V}| * r_1 \Leftrightarrow R > R_0 + R_1 \quad (3)$$

trong đó: $R = C_0 C_1 * |\vec{V}|$.

Bây giờ ta sẽ giải cụ thể các phương trình trên.

Với mỗi vector \vec{B}_i của hình bao thứ hai ta có thể biểu diễn thông qua các trục của hình bao thứ nhất, tức là: $\vec{B}_i = c_{0i} \vec{A}_0 + c_{1i} \vec{A}_1 + c_{2i} \vec{A}_2$ với $i=0, 1, 2$.

$$\text{Đặt: } A = (\vec{A}_0, \vec{A}_1, \vec{A}_2) \text{ và } B = (\vec{B}_0, \vec{B}_1, \vec{B}_2); C = \begin{bmatrix} c_{00} & c_{01} & c_{02} \\ c_{10} & c_{11} & c_{12} \\ c_{20} & c_{21} & c_{22} \end{bmatrix} \quad (4)$$

$\Rightarrow B = A * C$, do A được cấu thành từ $(\vec{A}_0, \vec{A}_1, \vec{A}_2)$ là các vector đơn vị và vuông góc với nhau

$$\Rightarrow A^T * B = C \Leftrightarrow C = \begin{bmatrix} |\vec{A}_0| \\ |\vec{A}_1| \\ |\vec{A}_2| \end{bmatrix} * \begin{bmatrix} B_0 & B_1 & B_2 \end{bmatrix} = \begin{bmatrix} \vec{A}_0 * \vec{B}_0 & \vec{A}_0 * \vec{B}_1 & \vec{A}_0 * \vec{B}_2 \\ \vec{A}_1 * \vec{B}_0 & \vec{A}_1 * \vec{B}_1 & \vec{A}_1 * \vec{B}_2 \\ \vec{A}_2 * \vec{B}_0 & \vec{A}_2 * \vec{B}_1 & \vec{A}_2 * \vec{B}_2 \end{bmatrix} \quad (5)$$

Từ (4) và (5) suy ra: $c_{ij} = \vec{A}_i * \vec{B}_j$, hay c_{ij} chính là tích vô hướng của hai vector \vec{A}_i, \vec{B}_j . Tương tự ta có $\vec{A}_i = c_{i0}\vec{B}_0 + c_{i1}\vec{B}_1 + c_{i2}\vec{B}_2$.

Như đã trình bày ở phần trên, tập các trục tách ứng cử viên $\vec{V} = \{ \vec{A}_i, \vec{B}_j, \vec{A}_i \otimes \vec{B}_j \}$ với $i, j = 0, 1, 2$.

Để tính toán các hệ số R_0, R_1, R ở trên thì ta sẽ làm minh họa cho hai trường hợp $\vec{V} = \vec{A}_i$ và $\vec{V} = \vec{A}_i \otimes \vec{B}_j$, các trường hợp còn lại được tính toán tương tự.

Xét trường hợp $\vec{V} = \vec{A}_0$:

- $R_0 = a_0 * |\vec{A}_0 * \vec{V}| + a_1 * |\vec{A}_1 * \vec{V}| + a_2 * |\vec{A}_2 * \vec{V}| = a_0$
- $R_1 = b_0 * |\vec{B}_0 * \vec{V}| + b_1 * |\vec{B}_1 * \vec{V}| + b_2 * |\vec{B}_2 * \vec{V}| = b_0 * |c_{00}| + b_1 * |c_{01}| + b_2 * |c_{02}|$
- $R = \vec{A}_0 * \vec{D}$, với $\vec{D} = \vec{C}_1 - \vec{C}_0$

Xét trường hợp $\vec{V} = \vec{A}_0 \otimes \vec{B}_0$:

$$R_0 = a_0 * |\vec{A}_0 * \vec{A}_0 \otimes \vec{B}_0| + a_1 * |\vec{A}_1 * \vec{A}_0 \otimes \vec{B}_0| + a_2 * |\vec{A}_2 * \vec{A}_0 \otimes \vec{B}_0|$$

$$\text{Mặt khác: } \vec{V} = \vec{A}_0 \otimes \vec{B}_0 = \vec{A}_0 \otimes (c_{00}\vec{A}_0 + c_{10}\vec{A}_1 + c_{20}\vec{A}_2) = c_{10}\vec{A}_2 - c_{20}\vec{A}_1$$

$$\Rightarrow R_0 = a_1 * |c_{20}| + a_2 * |c_{10}|$$

$$\Rightarrow R = \vec{V} * \vec{D} = (c_{10}\vec{A}_2 - c_{20}\vec{A}_1) * \vec{D}$$

$$\Rightarrow R_1 = b_0 * |\vec{B}_0 * \vec{V}| + b_1 * |\vec{B}_1 * \vec{V}| + b_2 * |\vec{B}_2 * \vec{V}|$$

Ta có thể viết lại \vec{V} dưới dạng:

$$\vec{V} = \vec{A}_0 \otimes \vec{B}_0 = (c_{00}\vec{B}_0 + c_{01}\vec{B}_1 + c_{02}\vec{B}_2) \otimes \vec{B}_0 = -c_{01}\vec{B}_2 + c_{02}\vec{B}_1$$

$$\Rightarrow \vec{B}_0 * \vec{V} = \vec{B}_0 * (-c_{01}\vec{B}_2 + c_{02}\vec{B}_1) = 0 \quad \Rightarrow \vec{B}_1 * \vec{V} = \vec{B}_1 * (-c_{01}\vec{B}_2 + c_{02}\vec{B}_1) = c_{02}$$

$$\Rightarrow \vec{B}_2 * \vec{V} = \vec{B}_2 * (-c_{01}\vec{B}_2 + c_{02}\vec{B}_1) = -c_{01} \quad \Rightarrow R_1 = b_1 * |c_{02}| + b_2 * |c_{01}|$$

Tiếp tục làm cho các trường hợp còn lại, ta xây dựng được bảng các giá trị cho R, R_0, R_1 (bảng 1).

4. THỰC NGHIỆM

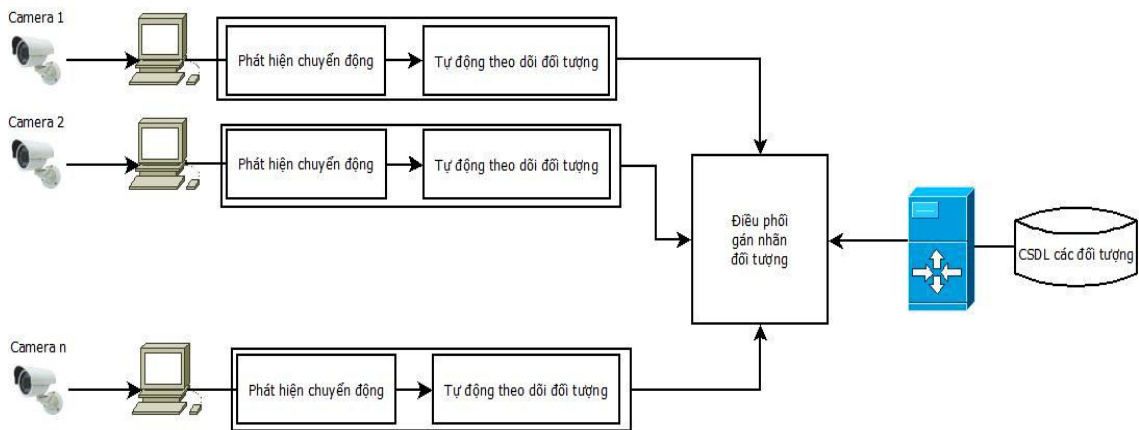
Nhằm kiểm chứng kĩ thuật đề xuất, chúng tôi đã tiến hành cài đặt chương trình demo VirtualLine. Chương trình được phát triển trên môi trường Visual C++ 2008 với bộ thư viện mã nguồn mở OpenCV. Đầu vào của chương trình là các file video hoặc kết nối trực tiếp với camera. Chương trình cho phép dựng đường ranh giới ảo, thực hiện tính toán, phát hiện sự va chạm của các đối tượng chuyển động với đường ranh giới ảo, các đối tượng sau khi va chạm với đường ranh giới ảo sẽ được đánh dấu, chương trình được thực hiện trên thời gian thực.

Mô hình của cấu trúc của hệ thống được thể hiện trong hình 11.

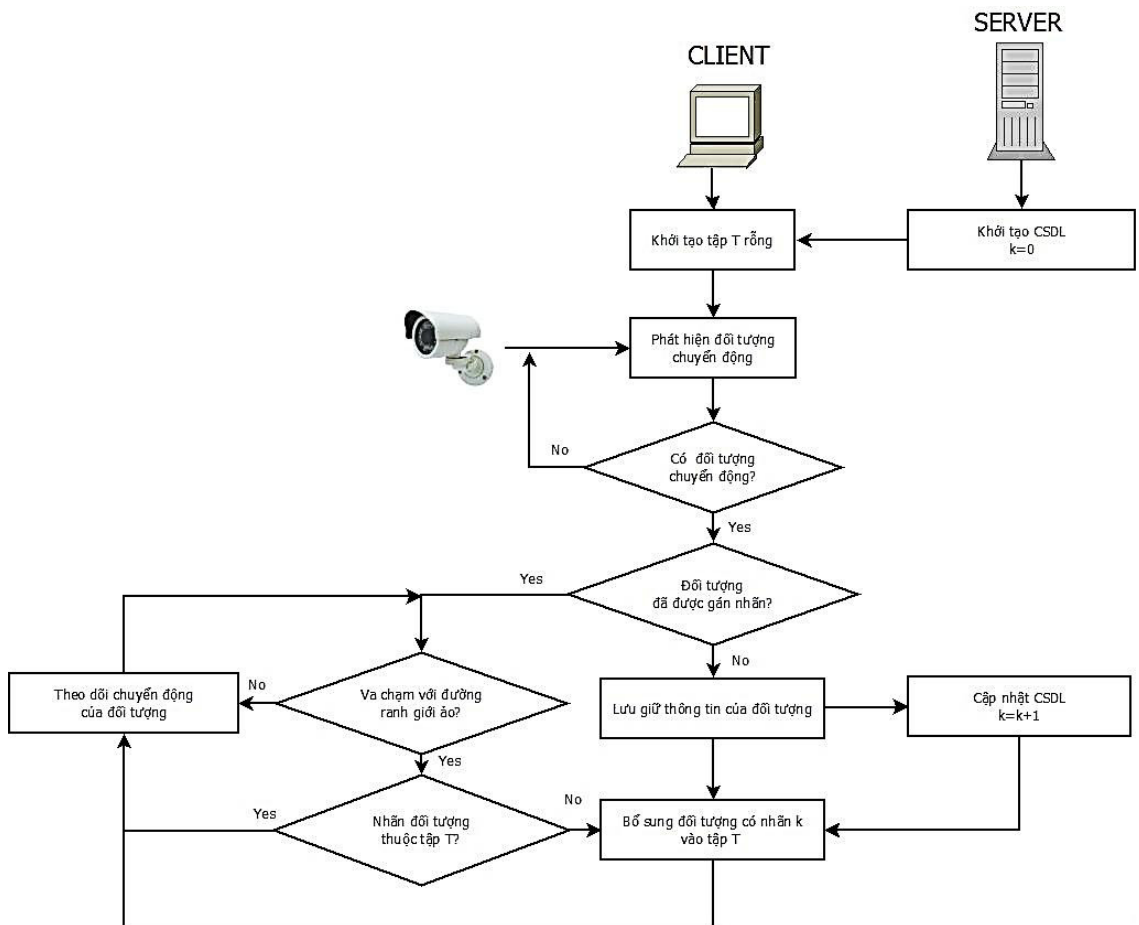
Bảng 1. Các giá trị R, R_0, R_1 .

| \vec{V} | R_0 | R_1 | R |
|-------------------------------|---|---|---|
| \vec{A}_0 | a_0 | $b_0 c_{00} + b_1 c_{01} + b_2 c_{02} $ | $ \vec{A}_0 \cdot \vec{D} $ |
| \vec{A}_1 | a_1 | $b_0 c_{10} + b_1 c_{11} + b_2 c_{12} $ | $ \vec{A}_1 \cdot \vec{D} $ |
| \vec{A}_2 | a_2 | $b_0 c_{20} + b_1 c_{21} + b_2 c_{22} $ | $ \vec{A}_2 \cdot \vec{D} $ |
| \vec{B}_0 | $a_0 c_{00} + a_1 c_{10} + a_2 c_{20} $ | b_0 | $ \vec{B}_0 \cdot \vec{D} $ |
| \vec{B}_1 | $a_0 c_{01} + a_1 c_{11} + a_2 c_{21} $ | b_1 | $ \vec{B}_1 \cdot \vec{D} $ |
| \vec{B}_2 | $a_0 c_{02} + a_1 c_{12} + a_2 c_{22} $ | b_2 | $ \vec{B}_2 \cdot \vec{D} $ |
| $\vec{A}_0 \otimes \vec{B}_0$ | $a_1 c_{20} + a_2 c_{10} $ | $b_1 c_{02} + b_2 c_{01} $ | $ c_{10}\vec{A}_2 \cdot \vec{D} - c_{20}\vec{A}_1 \cdot \vec{D} $ |
| $\vec{A}_0 \otimes \vec{B}_1$ | $a_1 c_{21} + a_2 c_{11} $ | $b_0 c_{02} + b_2 c_{00} $ | $ c_{11}\vec{A}_2 \cdot \vec{D} - c_{21}\vec{A}_1 \cdot \vec{D} $ |
| $\vec{A}_0 \otimes \vec{B}_2$ | $a_1 c_{22} + a_2 c_{12} $ | $b_0 c_{01} + b_1 c_{00} $ | $ c_{12}\vec{A}_2 \cdot \vec{D} - c_{22}\vec{A}_1 \cdot \vec{D} $ |
| $\vec{A}_1 \otimes \vec{B}_0$ | $a_0 c_{20} + a_2 c_{00} $ | $b_1 c_{12} + b_2 c_{11} $ | $ c_{20}\vec{A}_0 \cdot \vec{D} - c_{00}\vec{A}_2 \cdot \vec{D} $ |
| $\vec{A}_1 \otimes \vec{B}_1$ | $a_0 c_{21} + a_2 c_{01} $ | $b_0 c_{12} + b_2 c_{10} $ | $ c_{21}\vec{A}_0 \cdot \vec{D} - c_{01}\vec{A}_2 \cdot \vec{D} $ |
| $\vec{A}_1 \otimes \vec{B}_2$ | $a_0 c_{22} + a_2 c_{02} $ | $b_0 c_{11} + b_1 c_{10} $ | $ c_{22}\vec{A}_0 \cdot \vec{D} - c_{02}\vec{A}_2 \cdot \vec{D} $ |
| $\vec{A}_2 \otimes \vec{B}_0$ | $a_0 c_{10} + a_1 c_{00} $ | $b_1 c_{22} + b_2 c_{21} $ | $ c_{00}\vec{A}_1 \cdot \vec{D} - c_{10}\vec{A}_0 \cdot \vec{D} $ |
| $\vec{A}_2 \otimes \vec{B}_1$ | $a_0 c_{11} + a_1 c_{01} $ | $b_0 c_{22} + b_2 c_{20} $ | $ c_{01}\vec{A}_1 \cdot \vec{D} - c_{11}\vec{A}_0 \cdot \vec{D} $ |
| $\vec{A}_2 \otimes \vec{B}_2$ | $a_0 c_{12} + a_1 c_{02} $ | $b_0 c_{21} + b_1 c_{20} $ | $ c_{02}\vec{A}_1 \cdot \vec{D} - c_{12}\vec{A}_0 \cdot \vec{D} $ |

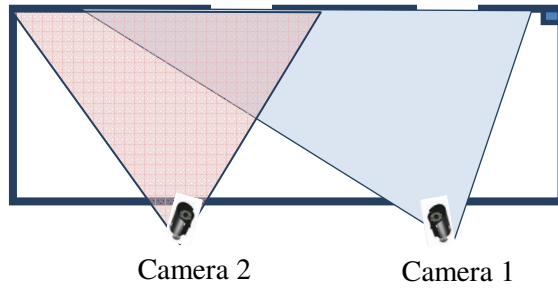
Một cách tiếp cận mới giải quyết việc chuyển tiếp các camera trong hệ thống giám sát tự động



Hình 11. Mô hình cấu trúc của hệ thống.



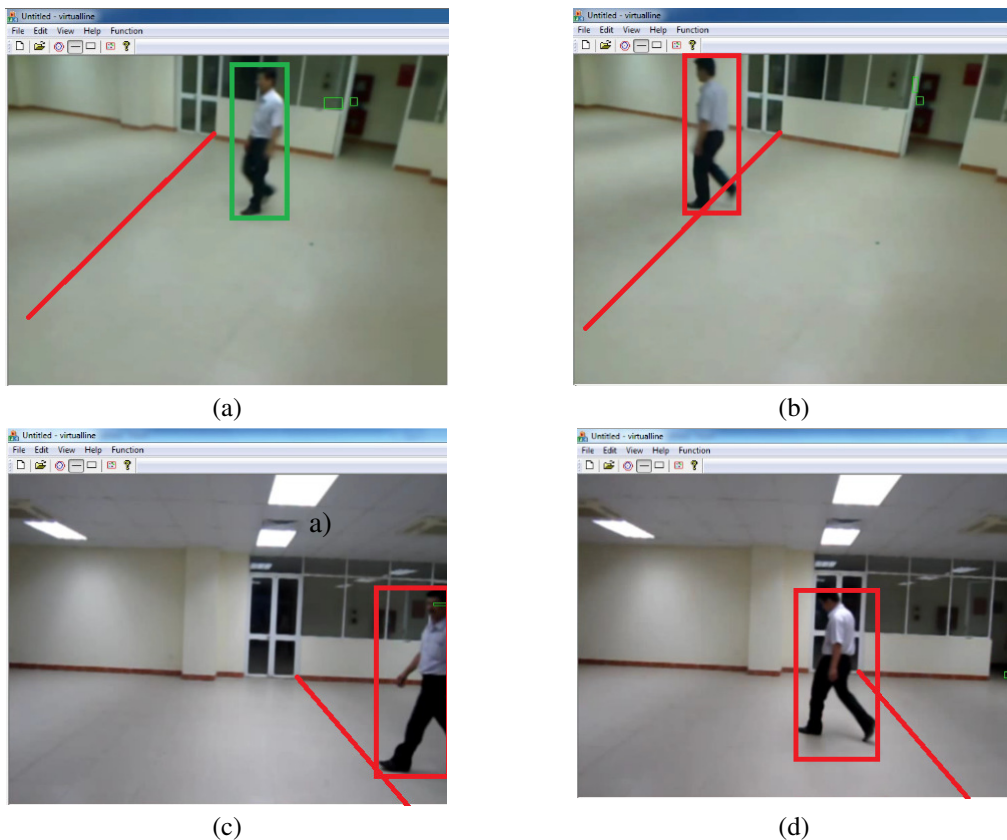
Hình 12. Sơ đồ thực hiện chương trình của hệ thống.



Hình 13. Sơ đồ mặt bằng bố trí camera.

Chúng tôi sử dụng 2 camera USB Logitech để giám sát một căn phòng với thị trường quan sát bị chùng chéo như hình 13. Các camera này được gắn cố định trên tường của căn phòng. Dữ liệu video được ghi với kích thước khung hình 640×480, tốc độ 15 hình/giây.

Thuật toán: Dựa vào việc tính toán va chạm ở trên (bảng 1), tại mỗi thời điểm liên tục kiểm tra xem có đối tượng nào va chạm với đường ranh giới ảo. Việc xác định đối tượng xuất hiện hay biến mất chỉ thực hiện khi phát hiện có sự va chạm, khi đó đối tượng sẽ được đánh dấu và thực hiện kiểm tra theo quy tắc đã trình bày trong mục 2.3 (hình 12).



Hình 14. Chuyển tiếp giữa 2 camera
(a), (b) Góc quay của camera 1; (c), (d) Góc quay của camera 2.

Hình 14 thể hiện việc chuyển tiếp theo dõi giữa 2 camera. Người di chuyển va chạm với đường ranh giới ảo giữa camera 1 và camera 2 (đường màu đỏ) đã được đánh dấu (màu đỏ) và xuất hiện tại vùng quan sát của camera 2.

Nhận xét: Quá trình phát hiện va chạm được thực hiện nhanh chóng, do ta đã lập được bảng chứa các thông tin cần tính toán, điều này rất có ý nghĩa đối với hệ thống thời gian thực.

5. KẾT LUẬN

Một trong những vấn đề quan trọng trong việc xây dựng hệ thống giám sát gồm nhiều camera là giải quyết việc chuyển tiếp giữa các camera trong hệ thống.

Trong bài báo, chúng tôi đã đưa ra một hướng tiếp cận giải quyết việc chuyển tiếp camera trong một hệ thống giám sát tự động có nhiều camera với vùng quan sát bị chồng lấn, bằng việc đưa ra các khái niệm về sự xuất hiện, biến mất của đối tượng trong vùng quan sát của camera cùng với các tính toán sự va chạm của đối tượng đến đường ranh giới ảo.

TÀI LIỆU THAM KHẢO

1. Teddy Ko - A survey on behavior analysis in video surveillance applications, Video Surveillance, InTec., 2011, pp. 279-294.
2. Omar Javed, Mubarak Shah - Automated multi-camera surveillance algorithms and Practice, Springer ISBN: 978-0-387-78880-7, 2008.
3. Faouzi Alaya Cheikh, Sajib Kumar Saha, Victoria Rudakova, Peng Wang - Multi-people tracking across multiple cameras. International Journal on New Computer Architectures and Their Applications (IJNCAA) **2** (1) (2012) 23-33.
4. Ting-Hsun, Chang, and Shaogang Gong - Tracking multiple people with a multi-camera system, IEEE workshop on multi object tracking, 2001, pp. 19-26.
5. Khan S. and Shah M. - Tracking people in presence of occlusion, Asian Conference on Computer Vision, 2001, pp. 1132-1137.
6. Khan S. and Shah M. - Consistent labeling of tracked objects in multiple cameras with overlapping fields of view, IEEE Transactions on Pattern Analysis and Machine Intelligence **25** (2003) 1355-1360.
7. Khan S. and Shah M. - Tracking multiple occluding people by localizing on multiple scene planes, IEEE Transactions On Pattern Analysis And Machine Intelligence **31** (3) (2009) 505-519.
8. Segen J. and Pingali S. G. - A camera-based system for tracking people in real time, Proceedings of the 13th International Conference on Pattern Recognition, Vol. 3, 1996, pp. 63-67.
9. LeFloch D. - Real-time people counting system using video camera. Master's thesis, Gjoevik University College, Universite de Bourgogne, 2007.
10. Liu Anan, Yang Zhaoxuan, Li Jintao - Video vehicle detection algorithm based on virtual line group. IEEE Asia Pacific Conference on Dec. 2006, pp. 1148-1151.
11. Zhai Y., Feris R., Hampapur A., Russo S., Pankanti S.- Parsing object events in heavy urban traffic. Object Tracking, Intech. 2011, pp. 201-208.

12. Mukundan R. - Advanced methods in computer graphics: With examples in OpenGL, Springer-Verlag London Limited, Chapter 9, 2012.
13. David Eberly - Dynamic collision detection using oriented bounding boxes, Geometric Tools, LLC, March 2008.

ABSTRACT

A NEW APPROACH FOR SOLVING THE TRANSITION FROM CAMERA TO CAMERA IN AUTOMATIC MONITORING SYSTEM

Ngo Duc Vinh^{1,*}, Do Nang Toan²

¹*Hanoi University of Industry, Nhon, Tu Liem, Hanoi*

²*Institute of Information Technology, VAST, 18 Hoang Quoc Viet, Cau Giay, Hanoi*

*Email: *ngoducvinh@gmail.com*

This paper proposed a calculation of the collision of the object with a virtual line. Virtual line essentially is a line drawn on each frame in the video data manipulation. Based on this calculation we could suggest objects to appear or disappear in an area, thereby to solve the transition between camera in a surveillance system including multiple cameras.

Keywords: virtual line, transition camera, automatic monitoring system, field of camera view.