

ỨNG DỤNG THỊ GIÁC MÁY TÍNH CHO TAY MÁY LẮP RÁP VẬT DI CHUYỂN TRÊN BĂNG CHUYỀN

COMPUTER VISION FOR ASSEMBLY OBJECT MOVING ON CONVEYOR USING MANIPULATOR

***Lê Đức Hạnh, Lê Đức Đạo**

Khoa Cơ khí, Trường Đại học Bách Khoa, Đại học Quốc gia TP. Hồ Chí Minh

TÓM TẮT

Ứng dụng xử lý ảnh để nhận dạng vị trí của vật và điều khiển tay máy để lắp ráp là một trong những ứng dụng rất phổ biến hiện nay. Tuy nhiên, hiện nay, những ứng dụng lắp ráp phổ biến đều là cho vật tĩnh đứng yên trên băng chuyền, hoặc là vật được gắp ra từ băng chuyền và lắp ráp trên vật tĩnh. Điều này làm cho năng suất của hệ thống bị giảm. Nghiên cứu này trình bày phương pháp có thể tăng năng suất cho một dây chuyền lắp ráp bằng cách lắp ráp cho vật động đang chuyển động trên băng chuyền sử dụng xử lý hình ảnh và tay máy. Hệ thống bao gồm một camera đặt trên cao để phát hiện vật cần lắp ráp đặt bất kỳ hướng và vị trí trên băng chuyền. Sau đó, tọa độ và hướng của vật được tính toán. Các dữ liệu tính toán sẽ được truyền xuống cho tay máy để thực thi tác vụ lắp ráp. Quá trình lắp ráp sẽ được thực hiện dựa trên vị trí và hướng của vật đã tính toán và tốc độ của băng tải. Qua thực nghiệm kiểm chứng, độ chính xác cũng như thời gian nhận dạng, hệ thống chứng minh là có khả năng áp dụng vào môi trường công nghiệp lắp ráp giúp tăng năng suất cho hệ thống.

Từ khóa: Thị giác máy tính; Lắp ráp; Tay máy; Điều khiển; Xử lý ảnh.

ABSTRACT

Image processing application to identify the position of objects and control manipulators for assembly is one of the most popular applications today. However, currently, common assembly applications are for stationary objects on a conveyor belt, or the object is picked up from a conveyor belt and assembled on a stationary object. This causes the system's productivity to decrease. This study presents a method that can increase the productivity of an assembly line by assembling moving objects on a conveyor belt using image processing and manipulators. The system includes an overhead camera to detect objects to be assembled in any direction and position on the conveyor belt. Then the coordinates and direction of the object are calculated. The calculation data will be transmitted to the manipulator to perform the assembly task. The assembly process will be performed based on the calculated position and direction of the object and the speed of the conveyor belt. Through experimental verification, accuracy and recognition time, the system proves to be applicable to the assembly industry environment to help increase system productivity.

Keywords: Computer vision; Assembly; Manipulator; Control; Image processing.



1. GIỚI THIỆU

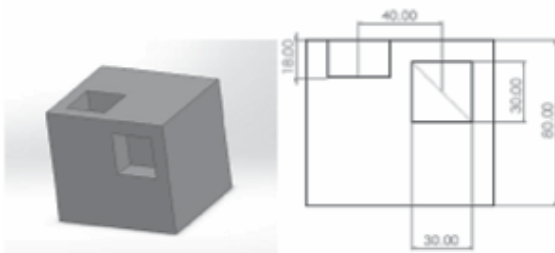
Ngày nay, ở các nhà máy tự động hóa có sử dụng tay máy công nghiệp như ABB, Fanuc, Yakawa, Denso, ở các cụm lắp ráp hay dây chuyền sản xuất, đóng gói các vật thường được di chuyển trên băng chuyền. Tuy nhiên, hiện nay để thực hiện tác vụ lắp ráp hay gấp vật, băng chuyền cần phải dừng lại trong một khoảng thời gian [1-4]. Điều này làm cho năng suất của hệ thống bị ảnh hưởng. Các hãng sản xuất robot hiện nay đang tập trung giải quyết vấn đề trên bằng việc đề xuất mô hình lắp ráp vật di chuyển trên băng tải mà không dừng băng tải như đề xuất của ABB [5]. Việc lắp ráp lúc vật di chuyển là một bài toán thách thức hơn rất nhiều so với vật tĩnh do hệ thống phải đảm bảo sai số nhỏ cũng như khả năng bám đối tượng trong thời gian thực.

Phát hiện đối tượng chuyển động đã được sử dụng rộng rãi trong định vị tự động, giám sát an ninh. Tuy nhiên, ứng dụng trong việc lắp ráp chi tiết trên băng chuyền vẫn chưa được ứng dụng rộng rãi do phải kết hợp giữa chương trình xử lý ảnh và cơ cấu chấp hành. Trong những năm gần đây, có rất nhiều nghiên cứu về việc bắt đối tượng chuyển động như: Li et al. [6] đã nghiên cứu thuật toán phát hiện đối tượng chuyển động bằng cách sử dụng các đặc điểm ảnh và thông số màu để trích xuất đặc điểm để phân đoạn tiền cảnh chuyển động theo thời gian thực và chính xác trong các cảnh phức tạp như thay đổi độ sáng và nhiều nền động. Jin et al. [7] đã đề xuất phương pháp phát hiện đối tượng chuyển động dựa trên mô hình biểu đồ vùng trong không gian thuộc tính màu, nhằm khắc phục những thiếu sót của phương pháp truyền thống dựa trên đặc điểm màu sắc và điểm ảnh. Đầu tiên, không gian màu RGB được ánh xạ tới không gian thuộc tính màu có chiều thấp mạnh mẽ hơn và biểu đồ được thiết lập

trong phạm vi cục bộ của pixel với thuộc tính màu làm đối tượng đặc trưng. Đồng thời, thông tin của pixel trong mỗi phân vùng của biểu đồ được ghi lại để tạo mô hình nền của từng pixel và mỗi biểu đồ sẽ cho các trọng số khác nhau tùy theo mức độ phù hợp của nó. Tuy nhiên, rất khó để thiết lập các thông số trọng số trong quá trình vận hành thực tế của phương pháp này và kết quả bị ảnh hưởng rất nhiều bởi các thông số trọng số này. Yin et al. [8] đề xuất một thuật toán phát hiện vật dựa trên sự bất thường dựa của những thay đổi thường xuyên. Đầu tiên, thuật toán phân đoạn vùng dựa trên phương pháp khác biệt ba khung hình được sử dụng để phân đoạn hình ảnh thành vùng nền tĩnh và vùng nền động bao gồm các đối tượng chuyển động, đồng thời phân chia vùng nền tĩnh và vùng nền động. Thuật toán Surendra dựa trên ba khung hình khác nhau được sử dụng để trích xuất vùng nền tĩnh thông tin và thiết lập mô hình nền. Ding [9] đề xuất mô hình nền với các tính năng màu sắc và cạnh để giải quyết vấn đề về tính năng đơn lẻ trong mô hình nền truyền thống phương pháp. Vấn đề nhiều, điểm đứt đoạn và lỗ hổng bên trong trong việc phát hiện đối tượng chuyển động dẫn đến phương pháp chênh lệch ba khung hình đã được giải quyết và kết quả xử lý hình ảnh được bù đắp dựa trên phương pháp xử lý hình thái. Nhưng nghiên cứu không giải quyết được vấn đề nhiều hình ảnh gây ra bởi chuyển động của mục tiêu trong hình ảnh video. Mặc dù phép trừ nền truyền thống các thuật toán đã được sử dụng rộng rãi trong các môi trường khác nhau nhưng chúng vẫn phải đối mặt với nhiều thách thức khác nhau [10]. Một trong số đó là vật thể ở tiền cảnh thường không hoàn chỉnh và có nhiều nhiễu, nghĩa là nó chứa các lỗ hoặc mảnh vỡ rải rác [11, 12]. Những khiếm khuyết này có thể gây ra việc phát hiện sai và ảnh hưởng đến hiệu quả của các ứng dụng nâng cao như nhận dạng vị trí mục tiêu và ngoại hình [13, 14, 15].

Bài báo này có sự khác biệt với các nghiên cứu trên, đó là sự kết hợp phát hiện vật chuyển động trên băng chuyền thông qua phân ngưỡng và lọc Morphology để xác định vị trí của vật lắp ráp và kết hợp tay máy để thực hiện việc lắp ráp này.

Lỗ lắp có kích thước độ sâu 18 mm, chiều dài 30 mm và chiều rộng 30 mm. Vật lắp ráp là khối vuông có chiều sâu, dài và rộng là 29 mm.

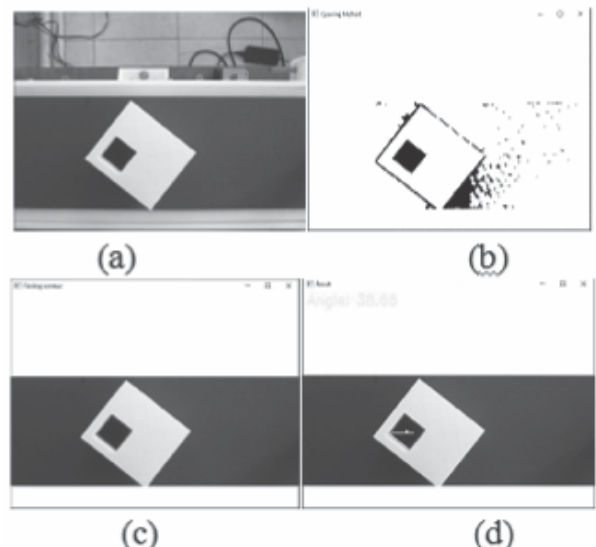


Hình 1. Hình ảnh vật và kích thước các lỗ cần lắp ráp trên vật khi di chuyển

2. XỬ LÝ ẢNH

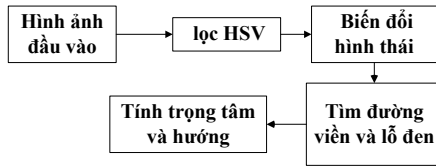
Trong hệ thống này, camera được sử dụng để phát hiện vật thể chuyển động trên băng tải. Sau đó, camera chụp ảnh và PC xử lý ảnh để trích xuất các đặc điểm của đối tượng như hình 2a, sau đó là phát hiện màu đen của lỗ phía trước. Thông thường, hình ảnh được đọc trong không gian màu RGB và phương pháp phân đoạn đơn giản nhất là phân ngưỡng. Từ một hình ảnh trong không gian màu RGB được chuyển sang màu xám và sau đó sử dụng thao tác ngưỡng để phát hiện. Tuy nhiên, phương pháp này vẫn có quá nhiều nhiễu cũng như phát hiện màu sắc không tốt vì chỉ sử dụng kênh màu xám. Để trích xuất một đối tượng màu, sử dụng không gian màu HSV và đặt ngưỡng hình ảnh HSV cho một phạm vi màu này là một phương pháp hiệu quả vì HSV dễ thể hiện màu hơn không gian màu RGB như hình 2b.

Sau khi trích xuất đối tượng màu đen, phương pháp biến đổi hình thái được sử dụng để có kết quả tốt hơn (loại bỏ nhiễu) trước khi phát hiện đường viền của đối tượng. Các phép biến đổi hình thái được chọn vì nó thường được thực hiện trên ảnh nhị phân. Hai toán tử hình thái cơ bản là Erosion và Dilation được áp dụng để lọc bỏ bớt nhiễu. Để tính trọng tâm và hướng của vật thể cần phải thực hiện nhiệm vụ tìm đường viền. Nhiệm vụ này có thể được giải thích đơn giản là một đường cong nối tất cả các điểm liên tục (đọc theo đường biên), có cùng màu sắc hoặc cường độ. Đường viền là một công cụ hữu ích để phân tích hình dạng, phát hiện và nhận dạng đối tượng. Từ kết quả của bước tìm đường viền, điều cần thiết là chỉ lấy được một đường viền từ ảnh. Vì vậy, ý tưởng là sử dụng diện tích đường viền để chọn như hình 2c. Sau khi tìm được đường viền, ta có được hình chữ nhật bao quanh nó. Áp dụng phương pháp Hu moment, ta tính trọng tâm và hướng của đường viền duy nhất đó và đó cũng chính là vật ta cần tìm như hình 2d. Quá trình xử lý ảnh được tóm tắt như hình 3.



Hình 2. Hình RGB (a), Hình sau khi lọc ngưỡng HSV (b), Hình sau khi tìm đường viền (c), Hình tính tọa độ trọng tâm và góc (d)





Hình 3. Giải thuật nhận dạng vật trên băng chuyển

3. TÍNH TOÁN TỌA ĐỘ THẬT CỦA VẬT

3.1. Tính chỉnh camera

Các tác vụ xử lý hình ảnh trả về tọa độ pixel và hướng của đối tượng. Tuy nhiên, robot không thể hiểu giá trị của tọa độ pixel và cần phải hiệu chỉnh camera. Hiệu chỉnh camera là một quá trình để chuyển tọa độ pixel của u, v đến vị trí tương ứng trong tọa độ bàn cờ, ở dạng X, Y và Z. Các giá trị tọa độ được tính toán sau đó sẽ được so sánh với các giá trị đã biết trước đó trên bảng bàn cờ để xác minh tính chính xác của chúng.

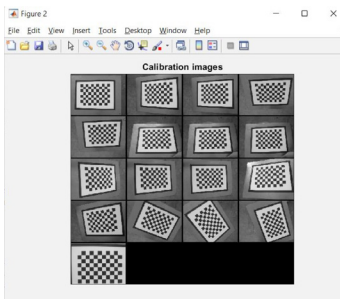
Công thức tổng quát:

$$s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K \begin{bmatrix} R & T \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \quad (1)$$

Trong đó:

- K: Ma trận thông số nội camera;
- [R T]: Ma trận thông số ngoại camera.

Quá trình calib camera được thực hiện trên Matlab, dùng hàm calib để lấy tất cả các ảnh chụp được ở 16 vị trí khác nhau như hình 4.



Hình 4. Các hình ảnh bàn cờ cho quá trình calib

Sau đó, lấy 04 góc của bàn cờ nhằm tìm ma trận thông số nội, làm liên tục với 15 hình còn lại. Sau khi xong quá trình trên, ta thu được ma trận thông số nội:

$$K = \begin{bmatrix} 667.4 & 0 & 324.4 \\ 0 & 666.5 & 241.05 \\ 0 & 0 & 1 \end{bmatrix} \quad (2)$$

Sau khi tìm được ma trận thông số nội, ta sẽ đặt bàn cờ lên mặt trên cùng của vật thể lớn và chụp ảnh đó dùng để tìm ma trận thông số ngoại:

$$T = \begin{bmatrix} -91.23 \\ -64.53 \\ 281.94 \end{bmatrix} \quad R = \begin{bmatrix} 0.0034 & 0 & 0.0197 \\ 1 & -0.0032 & -0.01 \\ -0.01 & 0.0197 & -1 \end{bmatrix} \quad (3)$$

3.2. Tính toán vận tốc di chuyển của vật

Quá trình tính toán vận tốc được kích hoạt qua điều kiện “Check speed”. Chương trình sẽ liên tục lấy dữ liệu vị trí và thời gian của đối tượng trên băng tải. Khi kết thúc quá trình này sẽ tiến hành tính ra vận tốc của băng tải qua hồi quy tuyến tính. Giá trị vận tốc được cập nhật lên hệ thống để dùng cho các mục đích tính toán khác.

Ta giả định việc setup vận tốc băng tải được thực hiện qua núm xoay biến trở và trong quá trình hoạt động, vận tốc băng tải không đổi bởi các điều kiện khác nhau. Ở phần trước, việc tiền xử lý ảnh đưa ra được các đối tượng trên băng tải, ta có thể tìm được tâm của đối tượng đó qua việc tính các moment ảnh ở mỗi thời điểm khác nhau.

Khi một đối tượng di chuyển trên băng tải, với vận tốc v là hằng số, ta có phương trình:

$$X = v.t + x_0 \quad (4)$$

Từ đó, ta thu thập được bảng số liệu sau:

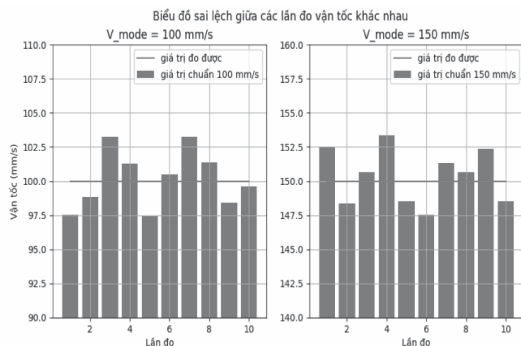
Bảng 1. Bảng số liệu thu thập

Vị trí	Thời gian
X1	T1
X2	T2
.....	
X4	T4

Các giá trị thu được là kết quả quan sát của một đối tượng.

Do hệ thống tách biệt vùng làm việc của camera và vùng làm việc của robot nên việc xác định vận tốc của của băng tải hết sức quan trọng. Nó là một thông số quan trọng để xác định vị trí của vật theo thời gian và phải có sai số nhỏ để đảm bảo robot có thể bắt chính xác được vị trí của vật.

Để xác định chỉ tiêu này, ta sử dụng thêm một encoder ngoài hệ thống và cho băng tải chạy ở các mức vận tốc khác nhau để tiến hành so sánh kết quả giữa hai phép đo như hình 5.



Hình 5. Sai số của vận tốc đo bằng camera

3.3. Chuyển đổi tọa độ từ bàn cờ về robot

Sau khi hiệu chuẩn đầu tool, ta di chuyển đầu tool ngay trên bàn cờ để xác định tọa độ bàn cờ trong tọa độ gắn với robot. Điểm

đầu tiên là gốc của bàn cờ tuân theo quy ước của quy tắc hiệu chuẩn máy ảnh. Điểm thứ hai và thứ ba lần lượt là các điểm nằm trên trục x và y của bàn cờ.

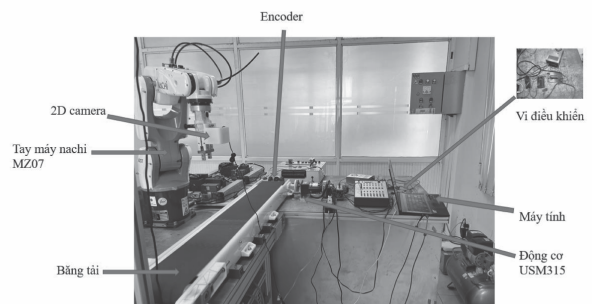
Tọa độ bàn cờ có thể tính toán bởi các điểm bên trên như sau:

$$\begin{aligned} {}^{robot}_{banco} \vec{X} &= \frac{P_2 - P_1}{|P_2 - P_1|} \\ {}^{robot}_{banco} \vec{Y} &= \frac{P_3 - P_1}{|P_3 - P_1|} \\ {}^{robot}_{banco} \vec{Z} &= {}^{robot}_{banco} \vec{X} \times {}^{robot}_{banco} \vec{Y} \end{aligned} \quad (5)$$

Trong đó: P1 là gốc, P2 là điểm nằm trên trục X, P3 là điểm nằm trên trục Y.

4. KẾT QUẢ VÀ THẢO LUẬN

Quá trình thực nghiệm được thực hiện trên robot Nachi 6 bậc tự do, camera 2D được đặt trên khâu cuối của robot như hình 6. Vật cần lắp nằm trên băng tải. Tốc độ di chuyển của băng tải là 50mm/s. Băng tải không dừng trong suốt quá trình lắp ráp.



Hình 6. Môi trường thực nghiệm

Quá trình hoạt động của thực nghiệm được tóm tắt như sau:

- Bước 1: Tay máy gấp chi tiết nhỏ về vị trí gốc. Camera luôn chụp ảnh ở bước này. PC sử dụng xử lý hình ảnh để phát hiện mẫu vật 🖱️

lớn đang di chuyển trên băng tải như hình 7a. Ngoài ra, các tác vụ xử lý hình ảnh trả về tâm pixel của lỗ phía trước, hướng của nó và tâm pixel của đối tượng lớn.

- Bước 2: Thread 1 của PC được khởi động, kết nối với bộ mã hóa để lấy giá trị vị trí cập nhật. Bộ vi điều khiển nhận lệnh “Bắt đầu” từ bộ mã hóa, sau đó nó đọc và gửi liên tục giá trị mới của bộ mã hóa đến PC. Dựa trên tọa độ pixel và giá trị của bộ mã hóa, PC tính toán vị trí hiện tại của lỗ phía trước trong khung robot. Luồng 2 được khởi động, PC gửi vị trí hiện tại và lệnh di chuyển tới robot. Robot theo dõi chuyển động của vật thể cũng như giảm dần trục Z để thực hiện nhiệm vụ lắp ráp như hình 7b. Nếu robot đạt đến vị trí lỗ phía trước, hãy lắp phôi nhỏ vào như hình 7c.

- Bước 3: Sau khi đưa phôi nhỏ vào, robot sẽ kẹp một phôi nhỏ khác như hình 7d. Vi điều khiển cũng tiếp tục gửi giá trị của bộ mã hóa tới PC. Vị trí lỗ bên hiện tại của lỗ bên được tính toán bằng PC nhờ nội suy liên quan giữa lỗ bên và lỗ bên trước. PC sẽ dự đoán vị trí lỗ bên cách vị trí hiện tại 200 mm, sau đó robot di chuyển đến vị trí này và giảm khoảng cách để hoàn thành công việc chèn như hình 7e và hình 7f.



(a)



(b)



(c)



(d)



(e)



(f)

Hình 7. Quá trình nhận dạng và lắp chai nhựa

Bảng 2. Thời gian xử lý của hệ thống

	Thời gian nhận dạng và tính toán vị trí (ms)	Thời gian robot thực thi (s)
Lần 1	164.67	1.88
Lần 2	180.23	1.91
Lần 3	201.45	2.12
Lần 4	180.1	1.93
Lần 5	188.39	1.88
Lần 6	160.97	2.15
Lần 7	188.31	2.04
Lần 8	178.86	2.13
Lần 9	212.38	1.95
Lần 10	218.75	1.99

Nhận xét:

- Hệ thống hoạt động tốt đối với các vật cần lắp ở vị trí khác nhau.
- Thời gian xử lý ảnh tổng cộng không vượt quá 0,25s, ứng dụng có thể chạy thời gian thực rất tốt.
- Thời gian thực thi của robot trung bình khoảng 2s phù hợp với tác vụ lắp ráp nhanh trên băng chuyền.

5. KẾT LUẬN

Nghiên cứu đã trình bày một phương pháp hiệu quả để tự động hóa quá trình lắp ráp vật trong dây chuyền sản xuất bằng cách sử dụng

dụng công nghệ xử lý hình ảnh và tay máy công nghiệp. Từ đây, vật trên băng chuyền không cần phải dừng lại cho quá trình lắp ráp mà vẫn có thể di chuyển liên tục. Qua thực nghiệm có thể thấy phương pháp có thể thay thế việc lắp ráp thủ công hiện nay của người lao động, tăng năng suất cho quá trình lắp ráp cũng như bảo vệ sức khỏe người lao động tốt hơn. Cách làm cũng đơn giản và dễ thực hiện cũng như dễ bảo trì. Để tăng độ chính xác, hệ thống camera có thể dễ dàng thay đổi.

Xung đột lợi ích:

Nhóm tác giả xin cam đoan rằng không có bất kỳ xung đột lợi ích nào trong công bố bài báo.

Đóng góp của tác giả:

- Lê Đức Hạnh: Đưa ra ý tưởng, phân tích và viết bài.
- Lê Đức Đạo: Thực nghiệm, thu thập dữ liệu và phân tích.❖

Ngày nhận bài: 12/12/2023

Ngày phản biện: 02/01/2024

Tài liệu tham khảo:

- [1]. *Efficient programming and teaching of DENSO robots*, <https://www.artiminds.com/robot-applications/robot-programming/denso-robot-programming/>
- [2]. *Robotic Solutions for Packaging*, <https://www.motoman.com/en-us/applications/picking-packing>
- [3]. *Solutions for Robotic Pick and Place and Packaging Automation*, <https://www.fanucamerica.com/solutions/applications/picking-and-packing-robots>
- [4]. *ABB confectionery palletizing solution enhances collaboration and improves productivity by 53 percent*, <https://new.abb.com/news/detail/73220/>
- [5]. *Why robots are focusing on final assembly processes*, <https://new.abb.com/news/detail/74750/why-robots-are-focusing-on-final-assembly-processes>
- [6]. Li Y, Xu X and Chen J (2017); *Application of background update of dynamic feature block matching in motion detection*, Journal of Instrumentation, 38 (2) 445-53.
- [7]. Jin J, Dang J and Wang Y (2019); *Spatial histogram background modeling of regional color attributes*, Chinese Journal of Image Graphics, pp. 714-23.
- [8]. Yin R, Yang J and Lu W (2019); *Research on anomaly detection algorithm based on the background of regular change*, Control Engineering, 26 (8).
- [9]. Ding Z and Lu W (2019), *Moving object detection algorithm based on vibe background modeling*, Computer System Application, 28 (04) 187-91.
- [10]. Di C (2017), *Research on video detection and tracking methods of moving objects based on features*, University of Chinese Academy of Sciences.
- [11]. Yin H, Chen B, Chai Y et al. (2016), *Overview of vision based object detection and tracking*, Journal of Automation, 42 (10) 1466-89.
- [12]. Wang D, Fan Y and Wang X (2017), *A motion detection algorithm based on gradient and vibe algorithm*, Television Technology, 41 (3) 100-04.
- [13]. Zhan Z, An Y and Cui W (2017), *Image threshold segmentation algorithm and comparative study*, Information Communication, (4) 86-89.
- [14]. Chen J (2017), *Research on threshold segmentation of non-uniform illumination image*, Hunan University of Technology.
- [15]. Hu Z, Fan W and Yu G (2017); *Adaptive threshold image foreground extraction based on OpenCV*, Photoelectric Technology Application, 32 (1) 56-59.