

## Real-Time Tracking Algorithm for Moving Target Based on Dynamic Monitoring Video

Chaoxiang Chen<sup>1</sup>, Wenshu Li<sup>2</sup>, Shiping Ye<sup>3</sup> and Huafeng Chen<sup>4</sup>

<sup>1</sup>*Department of Educational administration, Zhejiang Shuren University, Hangzhou Zhejiang, 310015, China*

<sup>2</sup>*School of Informatics and Electronics, Zhejiang Sci-Tech University, Hangzhou Zhejiang, 310018, China*

<sup>3</sup>*Zhejiang Shuren University, Hangzhou Zhejiang, 310015, China*

<sup>4</sup>*College of Computer and Information, Zhejiang Shuren University, Hangzhou Zhejiang, 310015, China*

### Abstract

*Target tracking algorithm is a key technical problem in the computer vision field, and the existing algorithms are mainly based on the stationary camera platform. Therefore, a real-time tracking algorithm for moving target based on dynamic monitoring video is proposed in this article. Firstly, the feature points are screened according to the influence of the moving target on the light streams; secondly, the light stream method is adopted to estimate the global motion of the camera; finally, the motion equation of the particle filter is modified according to the global motion estimation result and meanwhile the color histogram is selected as the characteristic model of the target, thus to track the moving target. The experiment result shows that the particle filter algorithm can rapidly and accurately track the moving target during camera motion process to meet the real-time requirement, and has excellent robustness.*

**Keywords:** *Mobile Platform; Feature Point Screening; Global Motion Estimation; Particle Filter Algorithm*

### 1. Introduction

As a key technical problem in the computer vision field, the video based target tracking algorithm is widely applied to pattern recognition, artificial intelligence, video surveillance, robot navigation, etc. [1-3] Common target tracking algorithms mainly include mean-shift algorithm [4], particle filter algorithm [5], etc., wherein the particle filter algorithm can effectively solve the nonlinearity problem and the non-Gaussian problem universally existing in target tracking, thus to be adopted by many scholars. The existing target tracking algorithms are mainly based on the stationary vision platforms, but most monitoring cameras in actual monitoring environments are in motion state, such as ship-borne and vehicle-mounted monitoring devices. Additionally, the electronic image stabilization technology [6-8] is discussed in many literatures, and the tracking algorithm based on electronic image stabilization [9] is discussed in a few literatures, but the image stabilization process and the tracking process are divided in such algorithm, thus causing the following unavoidable disadvantages: the image stabilization process inevitably causes the loss of the video image information; the separated implementation of the image stabilization process and the tracking process can cause time delay, so the real-time requirement cannot be met; the most important disadvantage is that we cannot obtain the tracking result in the original video.

Based on the above analysis, in order to solve the problems in existing algorithms, a real-time tracking algorithm for moving target based on dynamic monitoring video is proposed in this article. Firstly, the feature points are selected; secondly, the light stream

method is adopted to estimate the global motion to obtain the translational motion component and the rotational motion component of the camera; finally, the motion equation of the particle filter is modified according to the global motion estimation result, and meanwhile the color histogram is selected as the feature model of the target in order to track the moving target in a real-time manner during camera motion process. In order to verify the algorithm performance, we select multiple groups of test videos for the experiment. The experiment result shows that the proposed algorithm can rapidly and accurately track the moving target during camera motion process to meet the real-time requirement, and has excellent practical value.

## 2. Global Motion Estimation

### 2.1. Feature Point Selection

Accuracy and real-time performance shall be considered for the feature point selection of the image, and the angular point of each frame of the image is usually selected as the feature point in traditional image stabilization algorithm. The most universally used angular point is defined by Harris [10], and the angular point defined by Harris refers to the location where the autocorrelation matrix of the second derivative of the image has two maximum eigenvalues, namely: at least two textures (or edges) with different directions substantially exist around this point. However, the corresponding problems will be brought during the process of searching the angular point for each frame of the image: the real-time performance will be very poor when more than 300 angular points are needed and combined with the light stream.

Relevant experiment proves that it is more important to disperse the feature points into the whole image during the image stabilization process than to judge whether the feature point is an angular point. Enlightened by the image stabilization algorithm based on the block matching, we select one point every 20 pixels in each frame of the image as a feature point. The advantages of uniformly selecting the feature points are as follows: these feature points can realize excellent real-time performance and can well represent the global motion; the global motion can be accurately estimated through screening the local motion vector.

### 2.2. Feature Point Screening and Translational Component Estimation of Camera

In order to analyze the influence of the moving target on the light stream, we firstly suppose that the camera is stationary. If the coordinate of the  $i^{\text{th}}$  feature point  $P_i(t)$  in the camera coordinate system at time  $t$  is  $[x_i(t), y_i(t), z_i(t)]$ , then the coordinate of this feature point in the image plane is as follows:

$$[u(t), v_i(t)] = \left[ \lambda \frac{x_i(t)}{z_i(t)}, \lambda \frac{y_i(t)}{z_i(t)} \right] \quad (1)$$

In the above formula,  $\lambda$  is an coefficient related to camera resolution and focal length. If the displacement of this feature point in the camera coordinate system is  $dx_i(t), dy_i(t), dz_i(t)$ , then the light in the image plane is as follows:

$$\begin{aligned} \text{OF}_i(t) &= \left[ \frac{du_i(t)}{dt}, \frac{dv_i(t)}{dt} \right] \\ &= \left[ \lambda \frac{z_i(t) dx_i(t)/dt - x_i(t) dz_i(t)/dt}{z_i(t)^2}, \left[ \lambda \frac{z_i(t) dy_i(t)/dt - y_i(t) dz_i(t)/dt}{z_i(t)^2} \right] \right] \quad (2) \end{aligned}$$

According to Formula (2), the influence of the moving target on the light stream is related to the speed and DOF (Depth of Field) of the moving target. When the time interval is very short, we can regard the speed is fixed; the smaller DOF of the moving target has larger influence on the light stream. As shown in Figure 1, target A with large DOF is basically consistent with the background light stream, but target B with small DOF is inconsistent with the background light stream, so we need to completely remove the feature points inconsistent with the background light stream and then adopt the statistical method to estimate the global motion.

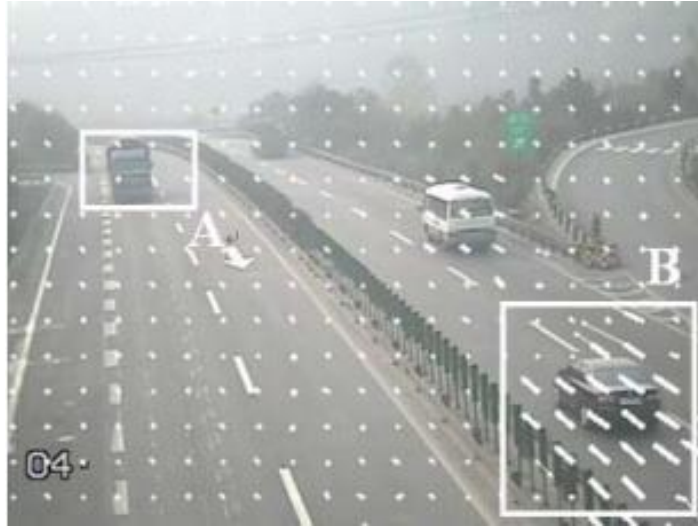


Figure 1. Light Stream Calculation Result

### 2.3. Estimation of Rotational Component of Camera

The coordinate of the  $i^{\text{th}}$  feature point  $P_i(t)$  in the camera coordinate system is  $[x_i(t), y_i(t), z_i(t)]$ , and the coordinate of this feature point in the world coordinate system is assumed to be unchanged, and after rotational and translational shifts of the camera, the coordinate of this feature point  $P_i(t+1)$  in the camera coordinate system at time  $t+1$  is changed as  $[x_i(t+1), y_i(t+1), z_i(t+1)]$ . Then, the following formula can be obtained:

$$\begin{aligned} & [x_i(t+1), y_i(t+1), z_i(t+1)]^T \\ & = R_{3 \times 3}(t) * [x_i(t), y_i(t), z_i(t)]^T + T_{3 \times 1}(t) \end{aligned} \quad (3)$$

In the above formula,  $R_{3 \times 3}(t)$  and  $T_{3 \times 1}(t)$  respectively represent the rotation matrix and the translation matrix of the camera at time  $t$ . Formula (1) is put into Formula (3) to obtain 3-D affine model of the camera as follows:

$$\begin{bmatrix} u_i(t+1) \\ v_i(t+1) \\ \lambda \end{bmatrix} = \frac{z_i(t)}{z_i(t+1)} \begin{bmatrix} R_{11}(t) & R_{12}(t) & R_{13}(t) \\ R_{21}(t) & R_{22}(t) & R_{23}(t) \\ R_{31}(t) & R_{32}(t) & R_{33}(t) \end{bmatrix} \begin{bmatrix} u_i(t) \\ v_i(t) \\ \lambda \end{bmatrix} + \begin{bmatrix} T^x(t) \\ T^y(t) \\ T^z(t) \end{bmatrix} \quad (4)$$

The camera rotation out of the image plane is ignored, and the simplified 2-D affine model can be obtained as follows:

$$\begin{bmatrix} u_i(t+1) \\ v_i(t+1) \end{bmatrix} = \frac{z_i(t)}{z_i(t+1)} \begin{bmatrix} R_{11}(t) & R_{12}(t) \\ R_{21}(t) & R_{22}(t) \end{bmatrix} \begin{bmatrix} u_i(t) \\ v_i(t) \end{bmatrix} + \begin{bmatrix} T^x(t) \\ T^y(t) \end{bmatrix} \quad (5)$$

Without considering the sudden change of DOF, the following condition can be met:

$$\frac{z_i(t)}{z_i(t+1)} = 1.$$

If the rotation angle of the camera at time  $t$  is  $\theta_i(t)$ , then the rotation matrix can be written as follows:

$$\begin{bmatrix} R_{11}(t) & R_{12}(t) \\ R_{21}(t) & R_{22}(t) \end{bmatrix} = \begin{bmatrix} \cos(\theta_i(t)) & -\sin(\theta_i(t)) \\ \sin(\theta_i(t)) & \cos(\theta_i(t)) \end{bmatrix} \quad (6)$$

The above formula is put into Formula (5) to obtain the following formula:

$$\begin{bmatrix} u_i(t+1) \\ v_i(t+1) \end{bmatrix} = \begin{bmatrix} \cos(\theta_i(t)) & -\sin(\theta_i(t)) \\ \sin(\theta_i(t)) & \cos(\theta_i(t)) \end{bmatrix} \begin{bmatrix} u_i(t) \\ v_i(t) \end{bmatrix} + \begin{bmatrix} T^x(t) \\ T^y(t) \end{bmatrix} \quad (7)$$

In Formula (6), the translation matrix  $[T^x(t), T^y(t)]^T$  of the camera has been solved in previous section; if the coordinates of the feature point in the image plane at time  $t$  and time  $t+1$  are known, then the rotation matrix at time  $t$  can be estimated.

### 3. Tracking Algorithm for Moving Target Based on Particle Filter

#### 3.1. Problem Modeling

The subject idea of the particle filter is to adopt one group of weighted particles to describe the posterior probability distribution. According to Monte Carlo theory, when there are enough particles, this group of weighted particles can accurately describe the posterior probability distribution and the particle filter at this moment is the optimal Bayesian estimation. In the target tracking task, the positions of the moving target can be regarded as the discrete-time Markov process in the state space. Actually, the purpose of the target tracking is to estimate the target state  $x_n$  in the observed sequence  $y_n$  with noises.

$y_{1:n} = \{y_i, i = 1, \dots, n\}$  is assumed as all known observed values, and  $p(x_n | y_{1:n})$  shall be finally solved. If  $p(x_0 | y_0) = p(x_0)$  is set and the distribution  $p(x_{n-1} | y_{1:n-1})$  at time  $n-1$  is known, then the prior probability distribution at time  $n$  can be obtained according to the system model:

$$p(x_n | y_{1:n}) = \int p(x_n | y_{n-1}) p(x_{n-1} | y_{1:n-1}) dx_{n-1} \quad (8)$$

After the new observed value  $y_n$  at time  $n$  is obtained, the posterior probability distribution can be obtained through Bayesian formula distraction distribution:

$$P(x_v | y_{1:n}) = \frac{p(y_n | x_n) p(x_n | y_{1:n-1})}{p(y_n | y_{1:n-1})} \quad (9)$$

It is difficult to adopt Bayesian filter to find the posterior probability, and Monte Carlo method is adopted in common engineering application. Specifically, one group of weighted particles  $\{x_n^m, \omega_n^m\}_{m=1}^N$  are used to represent the posterior probability  $p(x_n | y_{1:n})$ , wherein  $\{\omega_n^m, m = 1, \dots, N\}$  represents the weight values of the particles.

$$P(x_v | y_{1:n}) \approx \sum_{m=1}^N \omega_n^m \delta(x_{0:n} - x_{0:n}^m) \quad (10)$$

The recurrence equation of the normalized weight values  $\omega_n^m$  is as follows:

$$\omega_n^m \propto \omega_{n-1}^m \frac{p(y_n|x_n^m)p(x_n^m|x_{n-1}^m)}{\pi(x_n^m|x_{0:n-1}^m, y_{1:n})} \quad (11)$$

In the above formula,  $\pi(x_n^m|x_{0:n-1}^m, y_{1:n})$  is the proposal distribution. The main idea of Monte Carlo method is to adopt one group of samples following a certain probability distribution to replace the above probability distribution. In fact, the probability distribution calculation is only an intermediate step for target tracking, and since the samples following the probability distribution are obtained, the target position can be directly estimated according to the samples, without the need to calculate the specific probability distribution. In this way, the expected state density can be regarded as the estimation for the actual state:

$$\hat{x}_k^\Delta = E[x_n] \approx \sum_{i=1}^N \omega_n^m x_n^m \quad (12)$$

It can be proven that when  $N \rightarrow \infty$  is true,  $\hat{x}_k$  will converge in mean value to the real posterior probability distribution of  $p(x_n|y_{1:n})$ . Meanwhile, the re-sampling method shall be adopted to avoid particle weight degeneracy problem. Specifically, we copy the particles with large weight values and remove the particles with small weight values to solve such problem.

### 3.2. Quadratic Regression Motion Model

In actual tracking task, we adopt the video global motion information obtained in previous section to improve the particle tracking algorithm. Specifically, the quadratic auto-regression model is adopted as the motion model, and the motion model is modified according to the global motion estimation. If we select the position of the moving target in the image plane as the state variable, then the state variable at time  $n$  is  $x(n) = [u_i(k), v_i(k)]^T$ . Firstly, the global motion estimation obtained thereby is converted into the correction value  $\Delta_k$  of the target in the image plane:

$$\Delta_k = \begin{bmatrix} \cos(\theta_i(k)) & -\sin(\theta_i(k)) \\ \sin(\theta_i(k)) & \cos(\theta_i(k)) \end{bmatrix} x_k + \begin{bmatrix} T^x(k) \\ T^y(k) \end{bmatrix} \quad (13)$$

The modified motion model is as follows:

$$x_{k+1} - \hat{x}_k = A_1(x_k + \Delta_k - \hat{x}_k) + A_2(x_{k-1} + \Delta_k + \Delta_{k+1} - \hat{x}_k) + Bw \quad (14)$$

In the above formula,  $\bar{x}$  is the mean value of all particles, and  $Bw$  is the random noise of the system. In the modified motion model, the influence caused by camera motion is removed through particle position adjustment, thus to avoid the tracking failure caused by camera motion.

### 3.3. Color Histogram Distribution Model

The color histogram of RGB color space is adopted to describe the moving target. Specifically, R, G and B channels are quantized into 16 levels, and the histogram is divided into 16X16X16 bins. If  $u = \{1, 2, \dots, m\}$  is set and the particles including the target is given in the initial frame, then the statistical histogram distribution model of the target can be obtained as  $\hat{q} = \hat{q}_{u=1, \dots, m}$ , wherein the color probability density of the position is as follows:

$$\hat{q}_u = C \sum_{i=1}^n k \left( \left\| \frac{x_i^* - y_0}{h} \right\|^2 \right) \delta[b(x_i^*) - u] \quad (15)$$

In the above formula,  $y_0$  is the central position of the initial target, and  $k(\cdot)$  is the kernel function, wherein this kernel function can ensure that the pixel farther away from the particle center has smaller weight value, thus to improve the robustness of the model;  $C$  is a normalizing parameter for  $\sum_{u=1}^m \hat{q}_u = 1$ .

In each frame after the initial frame, the area that may include the target is regarded as a candidate area and the central coordinate thereof is  $y$ . Then, the candidate target area, namely the statistical histogram distribution model of the particles, is:  $\hat{p}(y) = \{\hat{p}_u(y)\}_{u=1, \dots, m}$ , wherein the color probability density of the position is as follows:

$$\hat{p}_u = C \sum_{i=1}^n k \left( \left\| \frac{x_i - y}{h} \right\|^2 \right) \delta[b(x_i) - u] \quad (16)$$

Bhattacharyya distance between the candidate target area  $\hat{P}(y)$  ( $y$ ) and the target area  $\hat{q}$  is as follows:

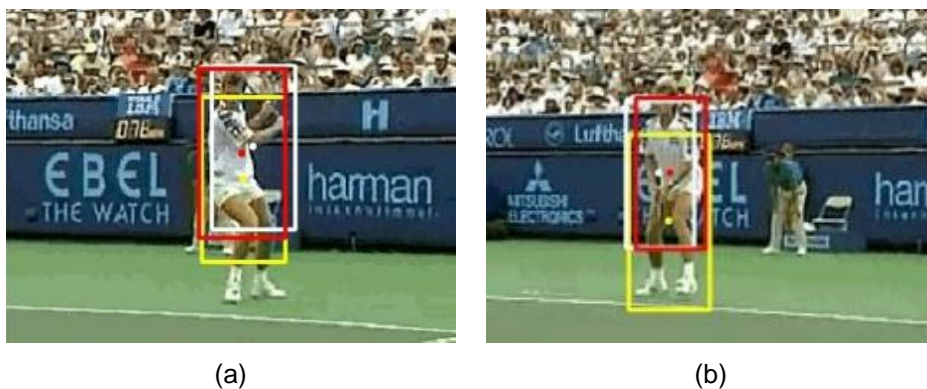
$$d(y) = \sqrt{1 - \rho[\hat{P}(y), \hat{q}]} \quad (17)$$

Then, we establish the observation likelihood model as follows:

$$p(y_n | x_n) = \frac{1}{\sqrt{2\pi}\sigma} \exp(-d(y)/2\sigma^2) \quad (18)$$

#### 4. Experiment Result and Analysis

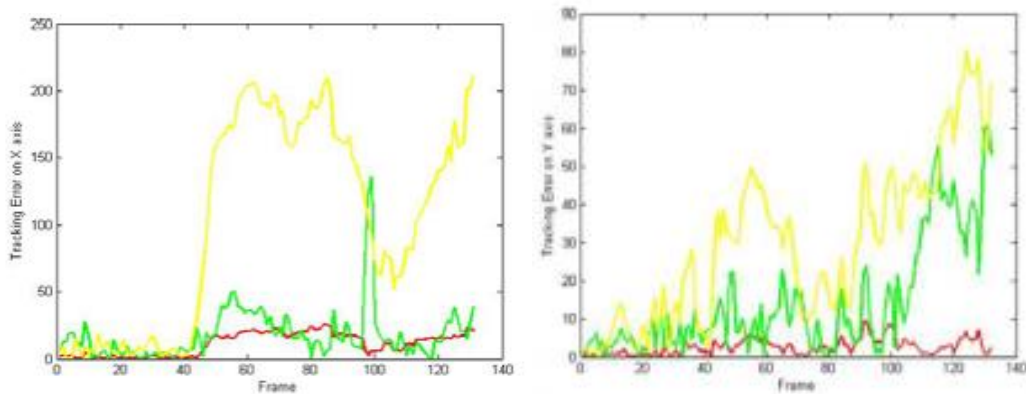
In order to verify the performance of the real-time tracking algorithm for moving target based on dynamic monitoring video, we adopt the experiment video for the experiment. The experiment environment for the implementation of the proposed algorithm is as follows: windows 7 operating system, Visual Studio 2010 and opencv2.3 as the software platform, Intel(R) Core(TM)i3 CPU M350 as the computer configuration. The target tracking result of the experiment video is as shown in Figure 2, wherein the red box in the figure represents the tracking result of the proposed algorithm, the white box represents the tracking result of the traditional particle filter algorithm and the yellow box represents the tracking result of the mean-shift algorithm. According to the experiment results, during camera motion process, the proposed algorithm is obviously superior to other algorithms in the aspects of accuracy and robustness.





**Figure 2. Comparison of Tracking Results of Different Algorithms**

The target tracking position error curve of the experiment video is as shown in Figure 3, wherein the red line represents the proposed algorithm, the green line represents the traditional particle filter tracking algorithm and the yellow line represents the mean-shift algorithm. According to the tracking position error curve, during the camera motion process, the tracking error of the proposed algorithm is obviously smaller than those of other two algorithms.



(a) X-axis Tracking Position Error (b) X-axis Tracking Position Error

**Figure 3. Tracking Error Results of Different Algorithms**

## 5. Conclusion

The target location algorithm based on dynamic monitoring video proposed in this article can not only conquer the disadvantages of separating the image stabilization process and the tracking process, but also capture the target tracking result in original motion video, without any loss of the video image information. Meanwhile, the proposed algorithm has low algorithm complexity and satisfactory robustness. The experiment result shows that the proposed algorithm has excellent real-time performance.

## References

- [1] J. Hu and Z. Gao, "Distinction immune genes of hepatitis-induced hepatocellular carcinoma", *Bioinformatics*, vol. 28, no. 24, (2012), pp. 3191-3194.
- [2] J. Yang, B. Chen and J. Zhou, "A Low-Power and Portable Biomedical Device for Respiratory Monitoring with a Stable Power Source", *Sensors*, vol. 15, no. 8, (2015), pp. 19618-19632.
- [3] G. Bao, L. Mi, Y. Geng and K. Pahlavan, "A computer vision based speed estimation technique for localizing the wireless capsule endoscope inside small intestine", 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), (2014).

- [4] X. Song and Y. Geng, "Distributed community detection optimization algorithm for complex networks", *Journal of Networks*, vol. 9, no. 10, (2014), pp. 2758-2765.
- [5] D. Jiang, X. Ying and Y. Han, "Collaborative multi-hop routing in cognitive wireless networks", *Wireless Personal Communications*, (2015), pp. 1-23.
- [6] Z. Lv, A. Halawani and S. Feng, "Multimodal hand and foot gesture interaction for handheld devices", *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 11, (2014), pp. 10.
- [7] G. Liu, Y. Geng and K. Pahlavan, "Effects of calibration RFID tags on performance of inertial navigation in indoor environment", 2015 International Conference on Computing, Networking and Communications (ICNC), (2015).
- [8] J. He, Y. Geng, Y. Wan, S. Li and K. Pahlavan, "A cyber physical test-bed for virtualization of RF access environment for body sensor network", *IEEE Sensor Journal*, vol. 13, no.10, (2013), pp. 3826-3836.
- [9] W. Huang and Y. Geng, "Identification Method of Attack Path Based on Immune Intrusion Detection", *Journal of Networks*, vo. 9, no.4, (2014), pp. 964-971.
- [10] X. Li, Z. Lv and J. Hu, "XEarth: A 3D GIS Platform for managing massive city information", *Computational Intelligence and Virtual Environments for Measurement Systems and Applications (CIVEMSA)*, 2015 IEEE International Conference on. IEEE, (2015), pp. 1-6.
- [11] J. He, Y. Geng, F. Liu and C. Xu, "CC-KF: Enhanced TOA Performance in Multipath and NLOS Indoor Extreme Environment", *IEEE Sensor Journal*, vol. 14, no. 11, (2014), pp. 3766-3774.
- [12] N. Lu, C. Lu, Z. Yang and Y. Geng, "Modeling Framework for Mining Lifecycle Management", *Journal of Networks*, vol. 9, no. 3, (2014), pp. 719-725.
- [13] Y. Geng and K. Pahlavan, "On the accuracy of rf and image processing based hybrid localization for wireless capsule endoscopy", *IEEE Wireless Communications and Networking Conference (WCNC)*, (2015).
- [14] X. Li, Z. Lv and J. Hu, "Traffic management and forecasting system based on 3d gis", *Cluster, Cloud and Grid Computing (CCGrid)*, 2015 15th IEEE/ACM International Symposium on, (2015), pp.991-998.

## Authors



**Chen Chaoxiang**, received her M.S. degree in software engineering from Hangzhou Dianzi University in Zhejiang, China. She is currently an associate professor in Zhejiang Shuren University. Her research interest is mainly in the area of Graphics image and Intelligent mining. She has completed several research and published several academic papers in the above research areas.