The 2009 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 11-15, 2009 St. Louis, USA

# BEST: A Real-time Tracking Method for Scout Robot

Diansheng Chen, Feng Bai, Peng Li, and Tianmiao Wang
*Robotics Institute, Beihang University*
*Beijing, China*
chends@163.com

*Abstract* - **We propose a BEST (Background subtraction and Enhanced camShift Tracking) method for a scout robot tracking a moving object in real time. A modified back subtraction method based on time axis is used to segment the moving object in a complicated environment. The centroid and area are chosen as the feature to judge target. We proposed a novel method that combines Camshift, AWS (Adaptive Window Selecting method) and Kalman predicting algorithm together to track the detected object. Experiments based on a DSP image processing system in a scout robot indicate the feasibility and robustness of our method.**

***Index terms*** *– Moving object, Tracking, Background subtraction, Camshift, Pan/Tilt*

## I. INTRODUCTION

The complexity of the situation in which the target lies and its attitude transformation, defection, blurring and occlusion make the scout robot identifying and tracking an object a complex process. The robot also needs to detect when moving in an acute background-changing circumstance, and hence it has to overcome the effect to the vision system caused by the bump.

Visual surveillance has been investigated worldwide under several large research projects. The DARPA supported the Visual Surveillance and Monitoring (VSAM) project in 1997, whose purpose was to develop automatic video understanding technologies that enable a single human operator to monitor behaviors over complex areas such as battlefields and civilian scenes [1]. Real-time visual surveillance system $W^4$ can analyze and track behaviors of groups of people in the presence of occlusion and in outdoor environment [2]. Embedded CMU cam system can track a color characterized object very fast, and can simply identify a human face [3].

Moving object detection, a basis of clustering, identification and tracking, is to find out the interesting region from motion image sequences. It may be classified into two kinds – the static and dynamic background. Background eliminating methods are usually used in a static case, as the temporal difference method [4] and background reconstruction [5], etc. And image registration [6] and optical flow methods are used in dynamic situations [7-9].

Reference [10] studies the compensation relationship between the image positions of objects and the rotation/tilt angles of a PTZ (Pan, Tilt, and Zoom), and the temporal differential method is applicable. In [11] the author uses the registration method with the pre-saved pictures. Limitations are storage volume and registration computation. Optical flow segmenting method in [12] has better noise immunity, but cannot deal with background occlusion, appearance and aperture caused by the movement of target. And the computation loads are great.

Correlation methods are suitable for tracking objects in complex situation as on ground surface. Image matching methods as region matching, feature matching, model matching and frequency domain matching methods [13, 14] are among them. They have high position accuracy but also high computation. Won, et al. proposed Snake model, which is suitable when there is no target overlapping in two successive images [15]. Frequency domain matching method is to detect object motion and can analyze human activities after transforming the video images to the frequency domain [16].

Generally, recent surveillance systems in moving state have features as big volume, low intelligence, heavy computation, poor servo performance, less image stabilizing measures and high cost. And cases of mounting such systems in an underground mobile robot are even fewer.

We propose a novel moving object detecting and tracking method - BEST (Background subtraction and Enhanced camShift Tracking). It uses an improved background subtraction method to precisely segment the moving object. A Camshift algorithm with similarity coefficient is utilized to track the detected target in real time. Kalman predicting method is supplemented to estimate the target position in images. The AWS (Adaptive Window Selecting) method is to reset the tracking process when the target is about to lose. We developed a tracking system using the BEST method, and installed it on a scout robot. The robot can track a moving object when moving and without complex algorithm (real time tracking).

This paper organizes as follows: Part II gives the overall system structure; Part III and IV details in detecting and tracking algorithms we used, respectively; Part V introduces the experiments and results; Part VI concludes this paper.

## II. SYSTEM OVERVIEW

This section introduces the object detecting and tracking system architecture of the scout robot and the platform. The composition chart of the system is as Fig. 1.

This system constitutes of one 2 DOF (degree of freedom) pan and tilt, one camera, a DSP based image processing system, and a motion controller. The dashed line in Fig. 1 means the camera is mounted on the pan and tilt,

and it rotates and tilts. The DSP image processing system sample video signals from the camera, convert them from analogue form to digital form, decode the signals, and finally perform the algorithms on the DSP chip. Tracking results are sent to the motion controller in which the two servo motors are modulated so that the tracked target will always appear in the FOV (Field of View). The motion controller ensures the pan and tilt rotate to the designated angle at an ordered speed. A laser indicator is fixed above the camera to represent the tracking result (Fig. 2). We realize the motion controller using AVR MCU from Atmel with integrated motor driver chip. It is mainly responsible for servo control of DC motors (position loop and speed loop) and communication between DSP and itself.

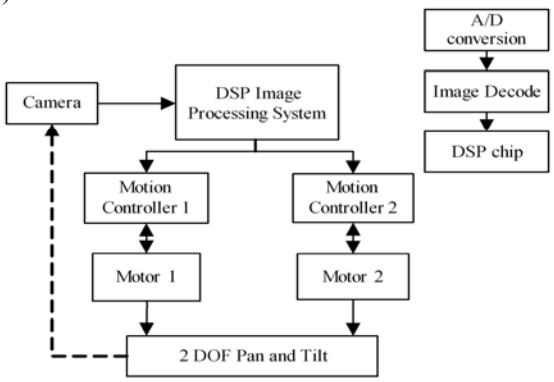



Fig. 1. Architecture of moving object tracking system

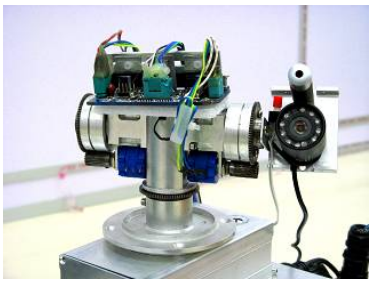


Fig. 2. Tracking system (Pan/tilt, motion controller, and camera)

Detecting and tracking algorithm needs a processor with great computation ability. TMS320DM643, a high-performance 32 bits fixed-point chip specialized in digital multimedia application, is chosen. It has a working frequency up to 600 MHz, with a theoretical processing performance up to 4800 MIPS due to an 8-level pipeline. The image acquisition device is an analogue camera; its signals (in PAL format) are sent to DSP chip after decoding to BT.656 format (YUV422). A 3.5 inch TFT LCD module driven by FPGA and a video encoder chip are used to display the processed image when debugging (Fig. 3). Communication between DSP and motion controller are based on RS232/485. The asynchronous serial chip, a voltage level transform chip and CPLD are also designed.

The algorithm flow chart which the pan and tilt use to track the target after initialization is as Fig.4. We will describe that in the Parts III and IV.

## III. Detecting Algorithm

In the detecting algorithm, the system will firstly preprocess the inputted images, transform the color space, and store the compressed images into RAM. Secondly, it will segment the moving target from the video images, and extract the features of centroid, area, and etc. Finally, the system will determine whether the appointed target is detected based on judgment conditions, and capture it. If it is not the one to track, the system will repeat the above procedures.

### *A. Image Pre-processing*

After sampled from the camera, the images are stored in SRAM. We use a 2 by 2 template based mean filtering to save the image in the size of 320 by 240. In the space of YUV, we extract Y component (luminance) for the following background subtraction method in gray space.

### *B. Modified Background Subtraction Algorithm*

Background subtraction, temporal difference, and optical flow are means to segment moving object in a clutter environment where the threshold method fails.

The background subtraction method assumes the camera is fixed in a position. It subtracts the gray background from the sampled gray images to obtain the moving object. This method has a fast computation and a precise detecting result and the key is to obtain the background image, whereas it is not easy in some circumstances. In addition, effects from noise may bring failure using only one frame of video. Reference [5] rebuilds the background through inter-frame information estimation and restoration.

In our system, it is also hard to detect as the camera is not still. The temporal mean filtering smoothes background and disturbances which may also influence the extracted images. We then use a median filtering on the time axis. i.e. we put several images together according to the time sequence, and make the median filtering on each pixel through every frame (The noises can be recognized as pepper noise, or instant pulse along the time axis). We therefore get a clean background without disturbs during initialization, with which we may further extract the moving target using background subtraction method. During this time, we request the robot stop and the camera stand towards a certain direction. We also use the opening operation in morphology to further remove the spots. Comparison results between temporal mean filtering and median filtering method are shown in Fig. 5, from which the actual background can be seen. In Fig. 5(b), the spots are caused by a shaking object during rebuilding background.

### *C. Feature Extraction*

Feature extraction is the key technique to identify or to track an object in identification or tracking occasions. Common features are centroid, area, length-width ratio, compact ratio, moment [17, 18], texture [19], etc. We select the coordinators of the centroid and the area of target image as the feature and extract the target from the image. A square that has the same area as the target image is drawn to represent the target region.

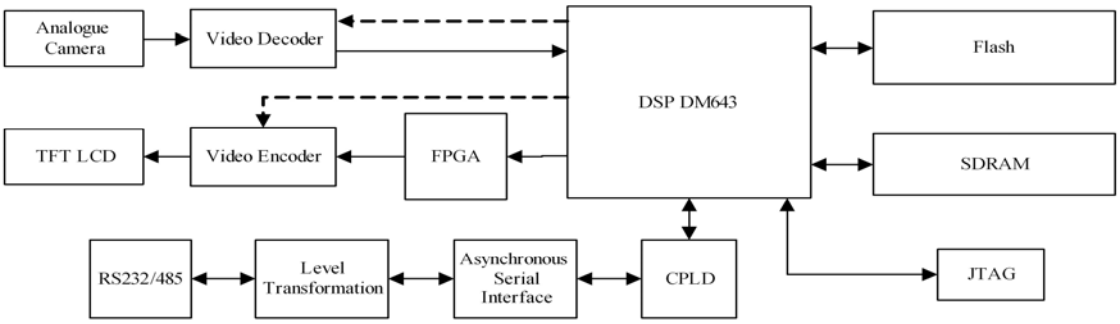



Fig. 3. Block diagram of DSP based image processing system

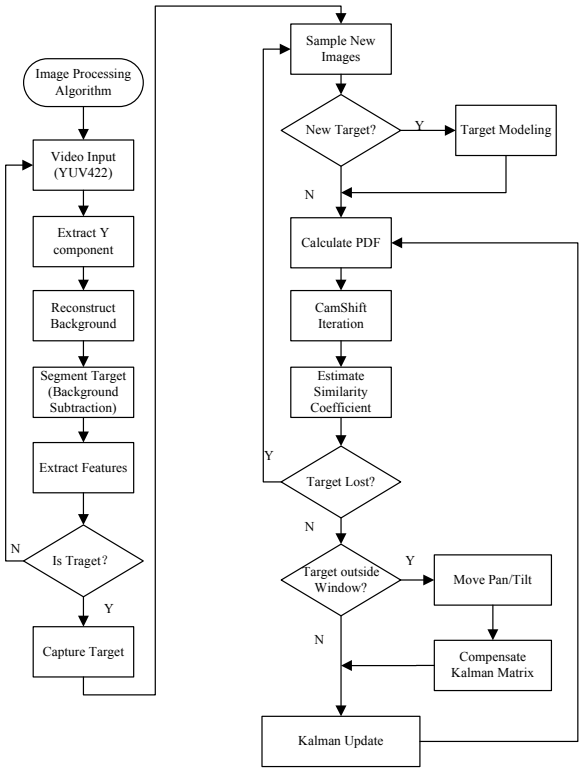



Fig.4. Flow chart of tracking system

The mass of center and the centroid of a binary image coincide at the same point. The coordinates of a digital image *f(x,y)* that has an area of S can be calculated using formula (1).

$$\begin{cases} \overline{x} = \dfrac{1}{S}\sum_{x=1}^{M}\sum_{y=1}^{N} x\delta(x,y) \\ \overline{y} = \dfrac{1}{S}\sum_{x=1}^{M}\sum_{y=1}^{N} y\delta(x,y) \end{cases} \tag{1}$$

Where, $S = \sum_{x=1}^{M}\sum_{y=1}^{N}\delta(x,y)$ is the area of target, and

$$\delta(x,y) = \begin{cases} 1, 0 < f(x,y) \le 255 \\ 0, f(x,y) = 0 \end{cases} \tag{2}$$

Then, the points of the border of the square are determined by (3).

$$\begin{cases} x_{l,r} = \overline{x} \pm \sqrt{S} \\ y_{l,r} = \overline{y} \pm \sqrt{S} \end{cases} \tag{3}$$

*D. Determining of the target*

The right target must meet these two constraints:

1) The centroid is within a designated window (wave gate), size of which should be smaller than the image window the camera can get, and position of which is near the center of FOV in case of that part of the target is chosen as the whole target when it starts to move into the FOV.

2) Area of the target image should be greater than a threshold, which can be determined through experiments. If it is too small, the result may be interfered with other tiny objects.

## IV. TRACKING ALGORITHM

We use an improved Camshift algorithm to track the object after the moving object is detected. Firstly, we build a model for the object. Secondly we locate the candidate target with Camshift iterative method. According to the estimation of the position and velocity of the candidate target, Kalman filtering is used to predict the next starting iterative point. We will rotate the pan/tilt so that the camera always orient to the target. When the object is missing, an Adaptive Window Selecting method (AWS) is utilized to continue the search.

*A. Target Modeling*

We define the regional center as $x_0$ , $n$ image pixel locations as $\{x_i\}, i = 1,...,n$ , and the number of eigenvalues $m$. Then, the probability density estimation of the target based on a kernel function with a probability of the feature $u = 1...m$ can be computed as:

$$\hat{q}_u = c\sum_{i=1}^{n} k\left(\left\|\frac{x_i - x_0}{h}\right\|^2\right)\delta\left[b(x_i) - u\right] \tag{4}$$

Where *k(x)* is the profile function of the kernel *K(x)*.The function *b*: $R^2 \to \{1...m\}$ associates to the pixel at $x_i$ the index $b(x_i)$ of its bin in the quantized feature space, and $\delta$ is the Kronecker delta function defined by:

$$\delta_{i,j} = \begin{cases} 1, i = j \\ 0, i \ne j \end{cases} \tag{5}$$

Due to the impact of occlusion, the pixels close to the center of the object are more reliable than remote pixels. So

the pixels are weighted by kernel function according to their distances from the center, and a near one gets a bigger weight.

a) Original b) Temporal Mean filtering c) Median filtering

Fig. 5. Comparison results using different methods to subtract background

The region of the target is set to the searching window, while the weight outside the window is set to zero. So in Camshift algorithm the profile function $k\left(\|x\|^2\right)$ is the corresponding profile for Uniform kernel, which is:

$$k_{\mathrm{U}}(x)=\begin{cases}1 & \|x\|\le 1\\ 0 & \text{other}\end{cases} \tag{6}$$

To eliminate the impact of the scale change of the object, function $k\left(\left\|\frac{x_i-x_0}{h}\right\|^2\right)$ is used to normalize the target area, and a constant C is also used for normalization so that $\sum_{u=1}^{m}\hat{q}_u=1$.

In our system, we first map the image from YUV color space to HSV space, and use the one-dimensional histogram into 16 bins consisting of the hue component from HSV color space as the feature.

*B. Candidate Modeling*

Camshift uses the histogram back projection [20] (also known as the Probability Distribution Function, PDF) for object modeling. The Probability Distribution Image (PDI) is plotted according to PDF, to show the probability density distribution of the object with corresponding characteristics. For a point in the candidate, we search the index in the hue histogram computed by the target model. So the index is also named as the probability distribution density of this position. The formula below is used for normalization to make the probability density distribution be independent from the size of the object.

$$\left\{\hat{p}_u=\min\left(\frac{255}{\max(\hat{q})}\hat{q}_u,255\right)\right\}_{u=1\ldots m} \tag{7}$$

. Similarity function is used to measure the similarity between object and candidate object as:

$$d(y)=\sqrt{1-\rho\left[\hat{p}(y),\hat{q}\right]} \tag{8}$$

Where candidate object's position is defined as y.

$$\hat{\rho}(y)\equiv\rho\left[\hat{p}(y),\hat{q}\right]=\sum_{u=1}^{m}\sqrt{\hat{p}_u(y)\hat{q}_u} \tag{9}$$

Formula (9) stands for Bhattacharyya coefficient of the sampled-data estimation.

In traditional Camshift algorithm, there is no similarity calculation, so we cannot measure the similarity between target and candidate, and that may cause great difference. In our enhanced Camshift modeling process, the normalized target and candidate histograms are used to compute Bhattacharyya coefficient as the criteria of object lose.

*C. Target Positioning and Adaptive Window Selecting*

In [21] zeroth moment and first moment according to PDF are used to determine the new search window. The iterative Camshift procedure leads to the optimal position of the candidate in one frame.

In a sequence of experiments it has been observed that, Mean Shift and traditional Camshift can both adapt to the situation of the change of the searched window, but once the window becomes narrow it's hard to become wider, and that decrease the reliability. Reference [22] puts forward a method that separately uses 0.9*h*, *h*, and 1.1*h* as the size of window, in the waste of time.

We use an adaptive window selecting (AWS) method to overcome the problem that the window shrinks. The window size of the next frame is determined by that in the detecting process with an assumption that the object changes less, and the size is not changed in iteration unless it is about to lose (when the Bhattacharyya coefficient is less than certain threshold). So at this time, we will reset the tracking window and set its position in the center of FOV (the tracked object is still in the image since the camera refreshes in a fast speed).

*D. Motion Prediction*

In traditional Camshift, the iteration starts from a specified location (such as the upper left corner of the image) in every new frame. But in the continuous video, displacement between successive frames is in fact very small. Therefore, we may set the starting position of the candidate be the result point of the last frame. We emerge Kalman filtering in the process of predicting the new position.

*E. Control Strategy of Pan and Tilt*

To improve the effect of our enhanced Camshift, and to avoid the pan/tilt moving in a very fast speed, we set an outside wave gate as in the detecting algorithm. Once the target is going to leave this wave gate, we will then rotate the pan/tilt, so the target returns in the FOV.

In addition, when the camera is not still, the observation matrix in Kalman filtering is no longer based

solely on the image coordinates unless an offset value is compensated.

## V. Experiments

We installed the moving object tracking system on the scout robot to validate the proposed algorithm above. The dimension of the robot is 575mm by 430mm by 260mm, and its weight is 26kg.without any loads. It is geared by crawler and driven by two high-power 24V DC servo motors. The maximum speed is about 1.1 meter per second on the flat land. It can also climb a step of 15cm high or a slope of 40 degree which is suitable for fieldwork. The scout robot and its remote controller are as in Fig. 6.

Fig. 6. Scout robot and remote controller

### *A. Two DOF Pan and Tilt Experiment*

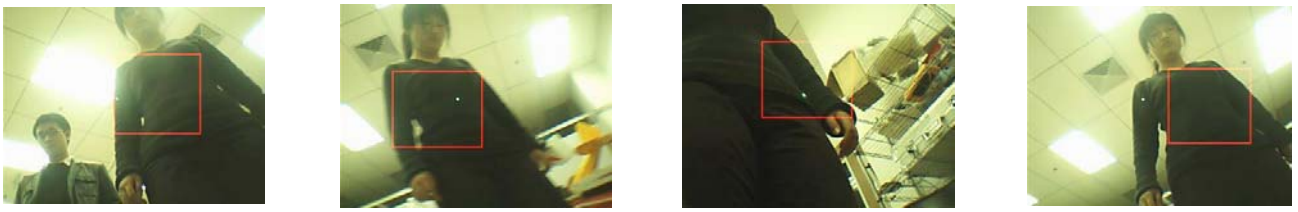
The 2 DOF pan and tilt is set to an extreme position at initialization. There is corresponding relationship between the pixels and the motor angles, or coordinates transformation. In order to obtain the camera's internal and external parameters, we should calibrate the camera. We assume it to be a linear relation to simplify the pan and tilt model. Through experiments, we make it out that the motor will rotate 30 degree when the centroid of target image moves about 235 pixels. The conversion between the motor rotating angles and the instruction from the motion controller is performed in DSP.

### *B. Tracking Experiment*

At the beginning, we let the robot in a static state to rebuild the background model. After a few seconds, the initialization completes. When a moving target appears in the FOV, the system rotates the pan and tilt towards the target. In order to simulate the robot in the field in reality, we use the remote controller to let the robot move forward or backward or rotate. The background of experimental environment was rather complicated, and strong lights affected the use of the traditional algorithm. But our tracking system can track the target in this situation.

We saved the sampled images and intermediate results for later analysis. And the algorithm we described in this article works. This video is all together about 300 frames. Some key frames are as in Fig. 7. The green dot, (almost white because of the worse picture quality), is lighted by the laser indicator of the system, which has nothing to do with the algorithm.

(a) background (b) the 41th frame (c) the 45th frame (d) the 59th frame



(e) the 95th frame (f) the 126th frame (g) the 169th frame (h) the 256th frame

Fig. 7 Video image sequences of moving object tracking experiments

Fig. 7(a) is the background after reconstruction. At the 41st frame, the target was selected by the system with a blue box. Because the goal is still in the FOV, the system did not immediately enter into the tracking state; at the 45th frame, the system started to track, in which the target was marked with a red box; at the 95th frame, there is a non-target disturb, but it did not disperse the target; the 169th frame is the tracking results where the robot was very close to the target. In the video sequences, because the camera and robot all moved rapidly, some images were obscure (as the 95th and 126th frame, with a linear speed of 0.3m/s and angular speed of 30 degree/s).

As in Fig.8, the centroid position scatter of the moving target tracking window was plotted in which most points fell in the neighbor of FOV and they were coincident with the trails of target. The coordinate units are all pixel of image.

In Fig.9, x axis represents frame, and y axis represents the similarity coefficients, which means the extent the target resembled the candidate. We may also note that the introduction of similarity coefficients to the traditional Camshift algorithm reduced the chance that the target lost.

Camshift algorithm iteration times (y axis) during the tracking state are shown as in Fig.10. The x axis is the frames. The mean iteration time was 3.1323 and the highest one was 15. Only few of them exceeded 5 times.

## VI. Conclusion

This paper studies the moving object tracking system of a scout robot. The system platform is introduced, and detecting and tracking algorithms are detailed explained. We proposed a novel method – BEST (Background subtraction and Enhanced camShift Tracking) to track the detected target in real time. A time-axis based median filtering is used to reconstruct the background. Bhattacharyya coefficient is added to traditional Camshift method to estimate when the target loses. Once the target loses, we use an adaptive window selecting method (AWS) to reset the tracking process. Kalman filtering is adopted to predict motion. Experiments show our method can deal the tracking problem when the robot moves.

Fig. 8. Centroid scatter of tracking experiment

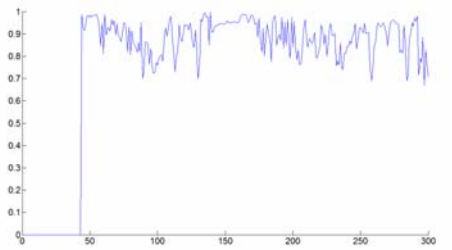


Fig. 9. Similarity coefficient curve

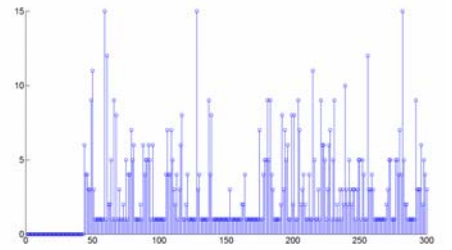


Fig. 10. Iteration times of enhanced Camshift

## ACKNOWLEDGMENT

This work is supported by National Science Fund for Distinguished Young Scholars of P.R.China Grant # 60525314.

## REFERENCES

[1] Weiming, H., Tieniu, T., Liang, W., and S, M., "A survey on visual surveillance of object motion and behaviors", IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews, 2004, 34, (3), pp. 334-352

[2] Haritaoglu, I., Harwood, D., and Davis, L.S., "W4: real-time surveillance of people and their activities", Ieee T Pattern Anal, 2000, 22, (8), pp. 809-830

[3] Rowe, A., Goode, A., Goel, D., and Nourbakhsh, I, "CMUcam3: An Open Programmable Embedded Vision Sensor", in Editor (Ed.) CMUcam3: An Open Programmable Embedded Vision Sensor (Robotics Institute at Carnegie Mellon University, 2007,edn.).

[4] Xuechao. Y, and Wenping L, "Moving objects detection technology in video sequences", Computer Applications and Software, 2008, 25, (01), pp. 215-217

[5] Xiaoyan M, Shuxing Y. and Bo M."The Analysis of Sequence Images Background Rebuild", Laser & Infrared, 2004, 34, (02), pp. 144-146

[6] Xing J., Yanhua Ma, and Rong S, "Image registration method of remote sensing image", Infrared, 2004, (09), pp. 23-30

[7] Talukder, A., Goldberg, S., Matthies, L., and Ansar, A., "Real-time detection of moving objects in a dynamic scene from moving robotic vehicles", in Editor (Ed.), pp. 1308-1313

[8] Braillon, C., Pradalier, C., Crowley, J.L., and Laugier, C., "Real-time moving obstacle detection using optical flow models", in Editor (Ed.), pp. 466-471

[9] Naian L, Ning O.Y, and Ming D, "Design and Implement Real-time Object Detection and Tracking System", Laser & Infrared, 2008, 38, (01), pp. 88-91

[10] Murray, D., and Basu, A., "Motion tracking with an active camera", Ieee T Pattern Anal, 1994, 16, (5), pp. 449-459

[11] Dellaert, F., and Collins, R., "Fast Image-Based Tracking by Selective Pixel Integration", Proceedings of the ICCV Workshop on Frame-Rate Vision, 1999, pp. 1-22

[12] Wenkuan S, Aimin H, and Qing W, "Survey on Object Tracking", Image Technology, 2006, (1), pp. 17-20

[13] Ling X, "A Survey of Image Matching in Computer Vision", Journal of Hubei University of Technology, 2006, 21, (03), pp. 171-173

[14] Hongmei W, Ke Z, and Yanjun Li, "Research Progress on Image Matching", Computer Engineering and Applications, 2004, (19), pp. 42-44

[15] W, K., Y, L.C., and J, L.J, "Tracking moving object using Snake's jump based on image flow", Mechatronics, 2001, 11, (2), pp. 199-226

[16] Guilin Z, and Jie X, "Frequency Technique in Image Matching Application", Mode Recognition and Artificial Intelligence, 1997, 10, (1), pp. 87-92

[17] Sheng X, and Qizong P, "Three Dimensional object recognition based combined moment invariants and neural network", Computer Engineering and Applications, 2008, (31), pp. 78-80

[18] Zhiqin Z, "Image Registration Methods", Fujian Compute, 2008, (11), pp. 55-56

[19] Wen W, Guosheng R, Xiaodong W, and Fucheng X, " A Review of Multiscale Statistical Image Models, Journal of Image and Graphics, 2007, (06), pp. 961-969

[20] Allen, J.G., Xu, R.Y., and Jin, J.S., "Object tracking using CamShift algorithm and multiple quantized feature spaces", 2004 pp. 3-7

[21] Bradski, G.R., "Computer Vision Face Tracking For Use in a Perceptual User Interface", Intel Technology Journal Q2, 1998, (2)

[22] Comaniciu, D., Ramesh, V., and Meer, P., "Kernel-Based Object Tracking", Ieee T Pattern Anal, 2003, 25, (5), pp. 564-577