Sensors **2012**, *12*, 8218-8235; doi:10.3390/s120608218

OPEN ACCESS

sensors

ISSN 1424-8220

www.mdpi.com/journal/sensors

Article

A Fast MEANSHIFT Algorithm-Based Target Tracking System

Jian Sun [1,2,3]

[1] State Key Laboratory for Strength & Vibration, School of Aerospace, Xi'an Jiaotong University, Xi'an 710049, China; E-Mail: sunjian10@xjtu.edu.cn; Tel.: +86-029-8258-2547

[2] State Key Laboratory of Precision Measurement Technology and Instruments, Tsinghua University, Beijing 100084, China

[3] College of Astronautics, Northwest Polytechnical University, Xi'an 710072, China

Received: 13 April 2012; in revised form: 9 May 2012 / Accepted: 22 May 2012 / Published: 13 June 2012

Abstract: Tracking moving targets in complex scenes using an active video camera is a challenging task. Tracking accuracy and efficiency are two key yet generally incompatible aspects of a Target Tracking System (TTS). A compromise scheme will be studied in this paper. A fast mean-shift-based Target Tracking scheme is designed and realized, which is robust to partial occlusion and changes in object appearance. The physical simulation shows that the image signal processing speed is >50 frame/s.

Keywords: target tracking system; moving object tracking; fast Mean-Shift algorithm

1. Introduction

Visual tracking plays an important role in various computer vision applications, such as surveillance [1,2], firing systems [3], vehicle navigation [4] and missile guidance [5]. Target tracking using an active video camera is a challenging task mainly due to three reasons [6–8]: (1) the tracking system should have good robustness to the targets' pose variation and occlusion; (2) tracking requires properly dealing with video camera motion through suitable estimation and compensation techniques; (3) most applications would introduce some real-time constraints, which require tracking techniques to reduce the computational time [5].

Target tracking, according to its properties, can be mainly divided into two types: feature- and optical flow-based approaches. Optical flow is the vector field which describes how the image changes with time [9]. The amplitude and direction of the optical flow vector of each pixel is usually computed by the Lucak-Kande algorithm. Shi and Tomasi [10] also proposed the well-known Shi-Tomasi-Kanade

(STK) tracker which iteratively computes the translation of a region centered on an interest point [9]. However, optical flow computation is too complicated to meet real-time requirements, and it is sensitive to illumination changes and noises, which limit its practical application.

Feature-based algorithms were originally developed for tracking a small number of salient features in an image sequence. These features include: color, grain, contour and some detection operators such as invariant feature transform (SIFT) [9] or histogram of oriented gradient (HOG) [11]. Feature-based algorithms involve the extraction of regions of interest in the images and then location of the target in individual images of the sequence. Typical feature-based tracking algorithms are: multiple hypothesis tracking (MHT) [12], Template Matching (TM) [13–16], Mean-Shift (MS) [17–19], Kalman filtering (KF) [20] and particle filter (PF) [21,22].

The TM is a simple and popular technique in target tracking, which is widely used in civilian and military automatic target recognition systems. Given an input and a template image, the matching algorithm finds the partial image that most closely matches the template image in terms of some specific criterion, such as the Euclidean distance or cross correlation. The conventional template matching methods consume a large amount of computational time. A number of techniques have been investigated with the intent of speeding up the template matching, and have given perfect results [14,15]. However, the TM does not achieve robust performance in complex scenes, especially in the case of clutter and occlusion [3].

The Kalman filter and particle filter are used to estimate target location in the next frame, which has also been extensively studied. Comparing to the Kalman filter, the particle filter has a more robust performance in the case of nonlinear and non-Gaussian problems due to the simulated posterior distribution. Many efforts have been carried out to speed up the particle filter. Martinez-del-Rincon *et al.* [21] proposed a new particle filter algorithm based on two sampling techniques, which improves substantially the efficiency of the filter. Sullivan *et al.* [23] proposed layered sampling using multiscale processing of images. It turns out that these solutions significantly reduce the computational costs, but in-depth efforts are desirable for better efficiency.

In image sequences, the target appearances have a strong correlation. Among all appearance based tracking models, there is one popular subset called "subspace model". Black [24] used a set of orthogonal vectors to describe the target image. Principal Component Analysis (PCA) and other classic dimensionality reduction methods provide an effective tool to compute the set of orthogonal vectors. Levy and Linden-Baum [25] presented a novel incremental PCA algorithm (Sequential Kathunen-Loeve, SKL) to update the eigen-basis when new data is available with greatly reduced computation and memory requirements. Lin applied Fisher linear discriminant analysis in subspace tracking to take background into account [26], however, it cannot perform well in case of non-Gaussian distribution.

The MS based tracker has very good robustness to the variation of translation, rotation and scale. The MS algorithm is a nonparametric density gradient estimation approach to local mode seeking and it was originally invented for data clustering. Comaniciu [18] was the first to develop its application in target tracking. The tracker needs a target model to be able to track. The target model is obtained from the color histogram of the moving object. The target candidate is obtained in the same way at a location specified by the MS algorithm. The similarity measure between the target candidate and the target model is computed using the Bhattacharya coefficient.

One of MS's drawbacks is that it often converges slowly. To the best of our knowledge, few attempts have been made to speed up the convergence of MS. The k_d-tree can be used to reduce the large number of nearest-neighbor queries. Although a dramatic decrease in the computational time is achieved for high-dimensional clustering, these techniques are not attractive for relatively low-dimensional problems such as visual tracking. Cheng [27] showed that mean shift is gradient ascent with an adaptive step size, but the theory behind the step sizes remains unclear.

The innovative work in this paper is to propose a novel fast robust tracking algorithm combining the MS with the template match (TM), which is a balanced scheme between robustness and real-time performance. A fast MS-based target tracking scheme is designed and implemented, which has a good robustness to target pose variation and partial occlusion. The hardware-in-loop simulation shows that the image signal processing speed is >50 frame/s.

The paper is organized as follows: the target tracking system description is described in Section 2, the hardware composition is presented in Section 3, the software structure and algorithm are described in details in Section 4, and, finally, Section 5 reports tests and results, and Section 6 describes the future works.

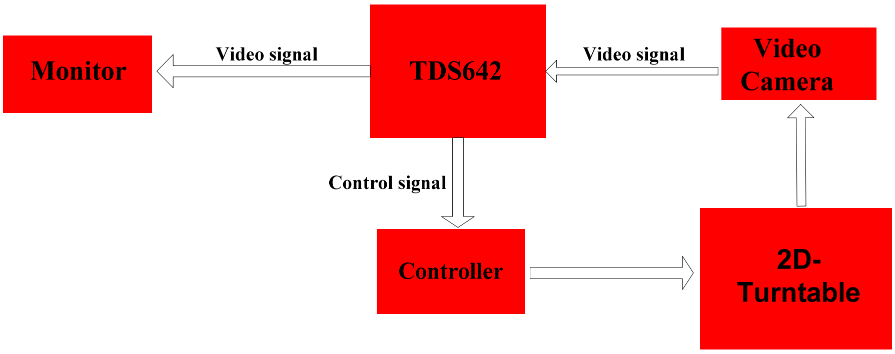
2. System Description

As shown in Figure 1, the target tracking system in this paper mainly has the following parts: video camera, signal processing module, monitor and 2D-turntable. In order to meet some practical application requirements the TTS must to have the following two performance features:

(1) Robustness. In a complex background, most of the applications require the tracker to be robust to partial occlusion, clutter and changes in object appearance.

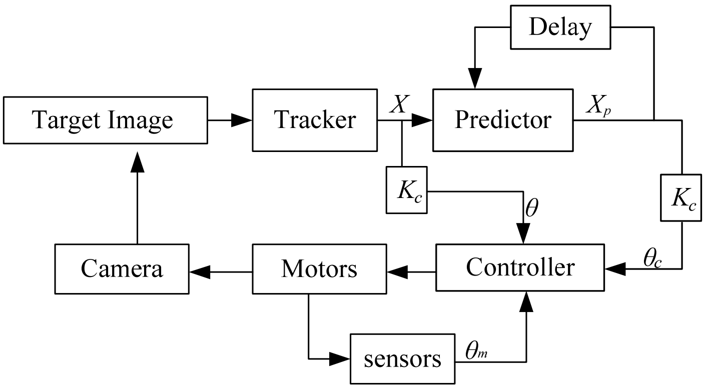
(2) Real-time performance. TTS needs to complete the image signal pre-processing, tracking and predicting target location, control 2D-turntable and other computational tasks which requires that the image processing speed should be >25 frames/s, and for some special applications processing speeds need to be >50 frame/s.

Figure 1. The target tracking system structure chart.



The signal flow diagram of a typical target tracking system is shown in Figure 2. The TTS obtains the target image by a video camera. Through a tracker, the target location X in the current image is obtained and sent to the predictor to predict the target location X_p in the next frame. The predicted result θ_c, the desired angle θ and the feedback angle θ_m are used to control the 2D-turntable.

Figure 2. Signal flow diagram of a typical target tracking system.



3. Hardware Composition

3.1. Signal Processing Module

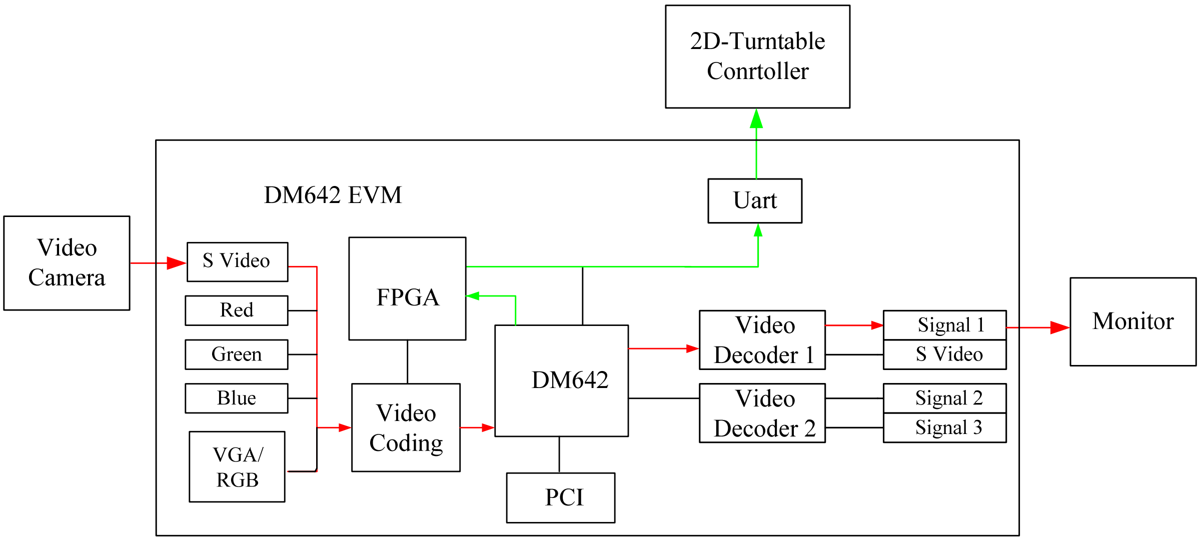
The video signal processer used in this paper is the TDS642EVM multi-channel real-time image processing platform produced by the TI Company. Its main performance features are listed in Table 1.

Table 1. The mainly performance of TDS642EVM.

DSP	DSP Chip	TMXDM642
	Operating voltage	I/O:3.3 V Vcore:1.4 V
	Clock	600 MHz
	External bus clock	100 MHz
Video In/Out	PAL/NTSC/SECAMS	
External Interface	RS232 UART	

The structure of the TDS642EVM is shown in Figure 3. The red line denotes video signal flow; the green line denotes control signal flow.

Figure 3. The structure of TDS642EVM.



3.2. Video Camera and 2D Turntable

The pitch and yaw axis of 2D-turntable (as shown in Figure 4) are linked with the output shaft of the stepping motor, respectively. The control of the 2D-turntable is realized by controlling the two stepper motors. The turntable controller obtains control instructions from TDS642EVM by a UART, and generates the pulse signal to drive the stepping motor. The rotation angle of the turntable measured by a potentiometer is used as the feedback for the closed-loop control system. The performance characterstics of the 2D turntable are given in Table 2.

Figure 4. 2D turntable and video camera.

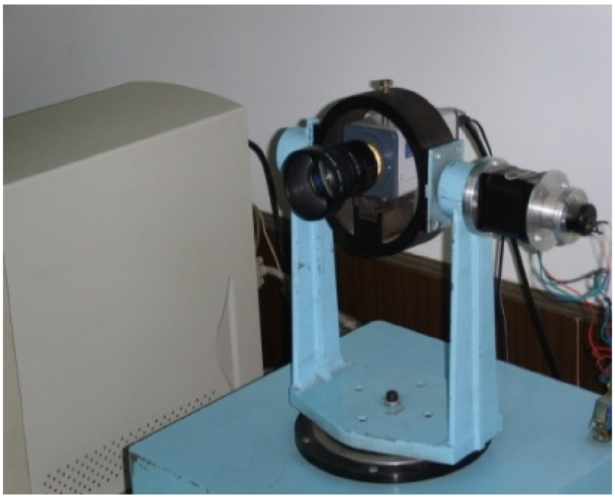


Table 2. Performance of the 2D turntable.

Maximum Speed	10°/s
Rotation Range	Pitch: ±20°; Yaw: ±80°
Motor Type	Stepper Motor
Maximum Torque	2 Nm

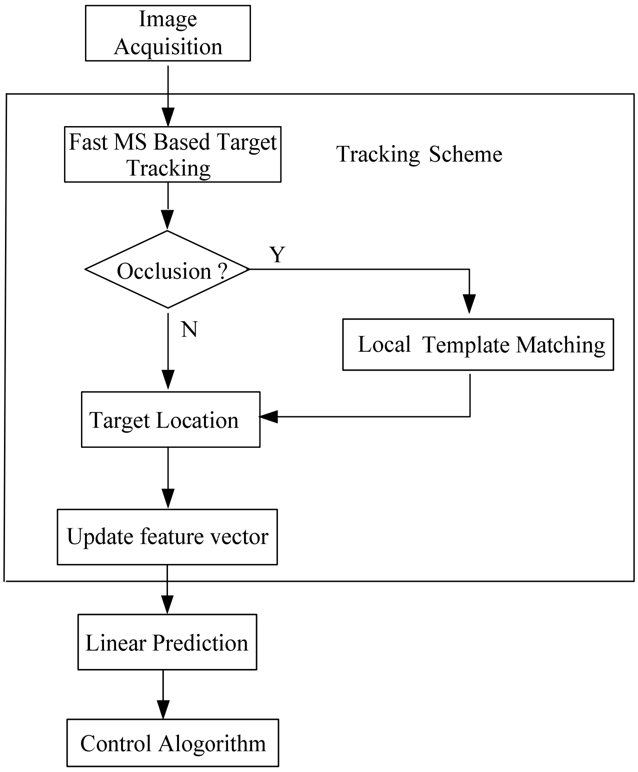
4. Software Structure and Algorithm

The structure of the TTS software is shown in Figure 5. The TTS software mainly includes the following two parts: image tracking algorithm, the target prediction algorithm.

(1) The tracking algorithm is to identify the location of the target in the current image. A fast robust MS-based target tracking algorithm is presented.

(2) The target prediction algorithm is to predict the location of the target in the next image though the sequence image. There are many algorithms that can achieve the prediction goal such as Kalman filter, particle filter and linear prediction method. Although the Kalman filter and particle filter [20,21] have obtained good results, these two algorithms are both inefficient. In this paper we use a linear prediction method to implement the target location prediction.

Figure 5. The structure of the TTS software structure.



4.1. Fast MS Tracking Scheme

4.1.1. Mean-Shift Basis [19]

Kernel density estimation is a nonparametric method that extracts information about the underlying structure of a data set when no appropriate parametric model is available. Given *n* data points x_i, $i = 1\ n$, in the *d* dimensional space $\mathbf{R}^{d\times d}$, the kernel density estimation at the location *x* can be computed by:

$$\hat{f}_K(x) = \frac{c_k}{nh^d}\sum_{i=1}^{n} k(\|\frac{x-x_i}{h}\|^2) \tag{1}$$

where $k(\cdot)$ is the profile the kernel function $K(\cdot)$ and c_k is a normalization constant. The optimization procedure of seeking the local modes is solved by setting the gradient equal to zero. Thus, we can derive the following equation:

$$m_G(x) = \frac{\sum_{i=1}^{n} x_i g(\|\frac{x-x_i}{h}\|^2)}{\sum_{i=1}^{n} g(\|\frac{x-x_i}{h}\|^2)} - x \tag{2}$$

where $g(x) = -k'(x)$, $m_G(x)$ is the MS vector.

4.1.2. Target Description and Distance Metric

According to the classical MS tracking algorithm [19], we can compute the target and candidate target feature vectors as follows:

Target feature vectors:

$$\hat{q}_u = C\sum_{i=1}^{n} k\ (\|\ x_i\ \|^2)\delta[b(x_i)-u],\ u=1\cdots m \tag{3}$$

Candidate target feature vectors:

$$\hat{p}_u(y)=C_h\sum_{i=1}^{n_h} k\ (\|\frac{y-x_i}{h}\|^2)\delta[b(x_i)-u],\ u=1\cdots m \tag{4}$$

where δ is the Kronecker delta function, $b(x_i)$ is the quantified number of the pixels value in the quantitative feature space, C, C_h are the normalization constants.

The similarity function defines a distance between target model and candidates. To accommodate comparisons among various targets, this distance should have a metric structure. We define the distance between two discrete distributions as:

$$d(y)=\sqrt{1-\rho[\hat{p}(y),\hat{q}]} \tag{5}$$

$$\hat{\rho}(y)=\rho[\hat{p}(y),\hat{q}]=\sum_{u=1}^{m}\sqrt{\hat{p}_u(y)\hat{q}_u} \tag{6}$$

where $\hat{\rho}(y)$ named Bhattacharyya coefficient.

4.1.3. Tracking Algorithm

To find the location of the target in the current frame, the distance (5) of a function of y should be minimized. The tracking starts from the location of the target in the previous frame and searches in the neighborhood. Minimizing the distance (5) is equivalent to maximizing the Bhattacharyya coefficient $\hat{\rho}(y)$.

Thus, the probabilities $\{\hat{p}_u(y_0)\}_{u=1,\ldots m}$ of the target candidate at location y_0 in the current frame must be computed first. Using Taylor expansion around the values $\hat{p}_u(y_0)$, the linear approximation of the Bhattacharyya coefficient (6) is obtained after some manipulations as:

$$\rho[\hat{p}(y\ ),\hat{q}]\approx\frac{1}{2}\sum_{u=1}^{m}\sqrt{\hat{p}_u(\hat{y}_0)\hat{q}_u}+\frac{1}{2}\sum_{u=1}^{m}\hat{p}_u(y\ )\sqrt{\frac{\hat{q}_u}{\hat{p}_u(\hat{y}_0)}} \tag{7}$$

This approximation is satisfactory when the candidate $\{\hat{p}_u(y)\}_{u=1,\ldots m}$ and the initial $\{\hat{p}_u(\hat{y}_1)\}_{u=1,\ldots m}$ are little difference. In general, for adjacent two frames this assumption is reasonable. Thus we have:

$$\rho[\hat{p}(y\ ),\hat{q}]\approx\frac{1}{2}\sum_{u=1}^{m}\sqrt{\hat{p}_u(\hat{y}_1)\hat{q}_u}+\frac{C_h}{2}\sum_{i=1}^{n_h}w_i k'(\|\frac{y-x_i}{h}\|^2) \tag{8}$$

In which:

$$w_i=\sum_{u=1}^{m}\delta[b(x_i)-u]\sqrt{\frac{\hat{q}_u}{\hat{p}_u(\hat{y}_1)}} \tag{9}$$

In this way, minimizing $d(y)$ becomes to maximize the second of Equation (8), which denotes the kernel density estimation computed by using $k(x)$ at the y in current frame. In this process, the kernel shifts from the current location y to the new location y_1. Thus we can use the MS procedure to find the great density estimation value in the neighborhood:

$$y_1 = \frac{\sum_{i=1}^{n_h} x_i w_i g(\| \frac{y - x_i}{h} \|^2)}{\sum_{i=1}^{n_h} w_i g(\| \frac{y - x_i}{h} \|^2)} \tag{10}$$

The general MS algorithm steps are as follows [19]:

Given: the target model $\{q_u\}_{u=1,\ldots,m}$ at $y_0$in the previous frame, y_1 is the new location of spot. Then the flow of MS algorithm is:

Set the spot with a feature vector $\{q_u\}_{u=1,\ldots,m}$, at y_0 in the previous frame.

(1) Compute the feature vector of candidate spot $\{\hat{p}_u(\hat{y}_0)\}_{u=1\ldots m}$, and evaluate Bhattacharyya coefficient $\rho[\hat{p}(\hat{y}_0),\hat{q}] = \sum_{u=1}^{m} \sqrt{\hat{p}_u(\hat{y}_0)\hat{q}_u}$

(2) Derive $\{w_i\}_{i=1\ldots m}$ with Equation (9).

(3) Find the new location of spot with Equation (10)

(4) Compute $\{\hat{p}_u(\hat{y}_1)\}_{u=1\ldots m}$ and evaluate $\rho[\hat{p}(\hat{y}_1),\hat{q}] = \sum_{u=1}^{m} \sqrt{\hat{p}_u(y_1)\hat{q}_u}$.

(5) While $\rho[\hat{p}(\hat{y}_1),\hat{q}] < \rho[\hat{p}(\hat{y}_0),\hat{q}]$

Do $\hat{y}_1 \leftarrow \frac{1}{2}(\hat{y}_0 + \hat{y}_1)$

Evaluate $\rho[\hat{p}(\hat{y}_1),\hat{q}]$

(6) If $\| \hat{y}_1 - \hat{y}_0 \| < \varepsilon$ stop iteration.

Else $\hat{y}_0 \leftarrow \hat{y}_1$ jump to 2

4.1.4. Fast Tracking Algorithm

References [27,28] show that MS is actually a bound maximization. One step of the MS iteration finds the exact maximum of the lower bound of the objective function. The existing literatures [21,29–33] also show that MS is a gradient ascent algorithm with adaptive step size. Hence, its convergence rate is better than conventional fixed-step gradient algorithms and no step-size parameters need to be tuned [17]. From the viewpoint of bound optimization, the learning rate can be over-relaxed to make its convergence faster.

From another point of view, bound optimization methods always adopt conservative bounds in order to guarantee increasing the cost function value at each iteration [17]. A lot of work has been done to speed up bound optimization methods. In [17,29], it was shown that by over-relaxing the step size, acceleration can be achieved. Supposing $\mathrm{M_G}$ is the MS shift vector, and then the over-relaxed bound optimization iteration is given by:

$$y^{(k+1)} = y^{(k)} + \alpha \cdot \mathrm{M_G} \tag{11}$$

Apparently when the $\alpha = 1$, over-relaxed optimization reduces to the standard MS algorithm. It is easily found that when $\alpha > 1$ acceleration is realized, but for a fixed α, no convergence is guaranteed and it is hard to get the optimal α [17]. References [17,31] prove that in the case of general bound optimization model, convergence can be secured using the over-relaxed bound optimization iteration when the candidate are close to a local maximum and $0 < \alpha < 2$. Based on this proposition, an adaptive over-relaxed bound optimization is readily available: α can be adjusted by evaluating the cost function.

When the cost function becomes worse for some $\alpha > 1$, then α has been set too large and needs to be reduced. By setting $\alpha = 1$ immediately, convergence can be achieved. In this paper, we presented the accelerated MS algorithm as follows:

1. Initialization:

Set the iteration index $k = 1$, and the step parameter $\beta > 1$, $\alpha = 1$.

2. Iterate until convergence condition is met:

(1) Compute $\hat{y}_{i+1}$ with Equation (13). And the MS vector $\mathrm{m}_G(\hat{y}_{i+1}) = \hat{y}_{i+1} - \hat{y}_i$.
(2) $y_{i+1} = y_i + \alpha\ \mathrm{m}_G(\hat{y}_{i+1})$
(3) If $\rho(y_{i+1}) > \rho(y_i)$
Accept y_{i+1} and $\alpha = 1$, $\alpha = \beta \cdot \alpha$.
Else reject y_{i+1}, and $y_{i+1} = \hat{y}_{i+1}$, $\alpha = 1$.
(4) Set $k = k + 1$, start a new iteration
(5) If $\mathrm{m}_G(\mathrm{y}_{i+1}) < \varepsilon$ stop iteration.

4.1.5. Case Study

We compare the performance of the accelerated MS algorithm to the standard MS algorithm on real images (as shown in Figure 6). In the experiments, all codes run on the EVM642 mentioned in Section 3. We repeat all the tests 10 times and the average CPU time is reported in Table 3. From the test results we can conclude that the Fast MS is at least three times faster than the standard MS.

Figure 6. Two images for fast MS *versus* the standard MS.

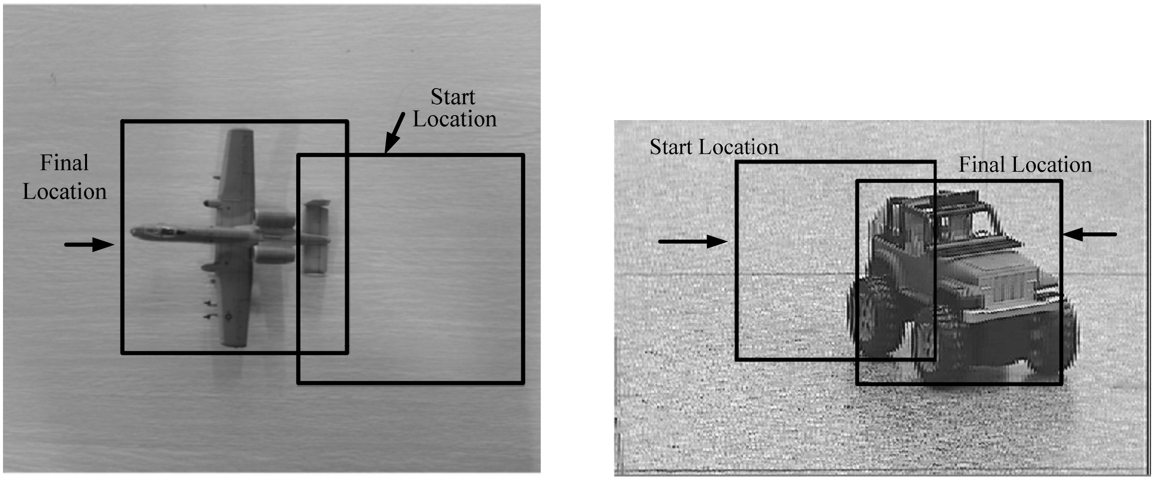


Image 1 Image 2

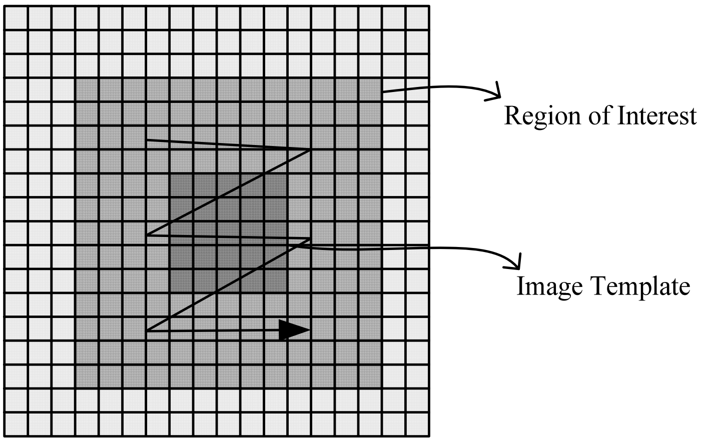
Table 3. Comparison of CPU times for two cases.

	Image 1		Image 2	
	Number of iterations	CPU time	Number of iterations	CPU time
Fast MS	8	18.8 ms	7	14.2 ms
Standard MS	26	61.1 ms	28	60.8 ms

4.1.6. Occlusion Issue

The occlusion issue is a technical challenge in the image tracking field. Many methods have been proposed to solve this problem. In this paper, the Bhattacharyya coefficient is used to determine whether the target is in occlusion or lost. Setting thresholds T1, T2, if T1 < Bhattacharyya coefficients < T2, the target is considered to be occluded, if Bhattacharyya coefficients < T1, the target is considered to be lost. In addition, by the effects of the environment illumination and the target appearance changes, the Bhattacharyya coefficient of the target candidate is, in general, the local maximum rather than the global maximum. When the target is in occlusion, the distance between the local maximum and the global maximum would increase, so some special method needs to be implemented to improve the tracking robustness. The Local Template Matching (LTM) method is used in this article to solve this problem. Template Matching (LM) is an existing algorithm, and, usually, it is a global template matching technique. In this paper template matching is implemented in the region of the candidate target, so here it is called Local Template Matching.

Figure 7. The template matching algorithm.



The final location (x, y) of the target is computed over a region of interest (ROI) surrounding the candidate location derived from the fast MS as shown in Figure 7. The LTM algorithm is as follows:

$$D(x,y)=\sum_{u=1}^{M}\sum_{v=1}^{M}|R(u+x,v+y)-S(x,y)| \tag{12}$$

where $S(x, y)$ is the pixel value at (x, y) in template image, $R(u+x, v+y)$ is the pixel value at $(u+x, v+y)$ in the search area. (u, v) is the candidate location derived from the fast MS. $D(x, y)$ is the distance in the feature space, and a smaller value shows a higher correlation. Then the minimum distance $D_{\mathrm{Min}}(x, y)$ and the corresponding location (x, y) are determined.

4.2. Target Prediction Algorithm

In order to improve the TTS response speed it is necessary to use the prediction method in the tracking scheme. Compared to the Kalman filter and particle filter, the linear prediction algorithm is less complex and offers moderate performance. In this paper we use the linear prediction method to get the predicted angular position of the target.

A simple method to estimate the location of the target in the image can be formulated by the following equation:

$$(\hat{x}_{t+1}, \hat{y}_{t+1}) = (x_t + \Delta\hat{x}_{t+1}, y_t + \Delta\hat{y}_{t+1}) \approx (x_t + \Delta x_t, y_t + \Delta y_t) \tag{13}$$

where $(\hat{x}_{t+1}, \hat{y}_{t+1})$ represents the estimated location of the target, $(\Delta\hat{x}_{t+1}, \Delta\hat{y}_{t+1})$ represents the estimated shift vector from t to t + 1, $(\Delta x_t, \Delta y_t)$ represents the real shift vector from t − 1 to t. This method assumes that the shift vector of velocity of the target is unchanged in a short time.

Another advanced algorithm which formulates the shift vector $(\Delta\hat{x}_{t+1}, \Delta\hat{y}_{t+1})$ as a linear combination of the shift vectors: $\{(\Delta x_{t-k}, \Delta y_{t-k}), (\Delta x_{t-k+1}, \Delta y_{t-k+1}), \ldots, (\Delta x_t, \Delta y_t)\}$. Then the following equation is:

$$\Delta\hat{x}_{t+1} = a_k \cdot \Delta x_{t-k} + a_{k-1} \cdot \Delta x_{t-k+1} + \cdots + a_0 \cdot \Delta x_t \tag{14}$$

$$\Delta\hat{y}_{t+1} = b_k \cdot \Delta y_{t-k} + b_{k-1} \cdot \Delta y_{t-k+1} + \cdots + b_0 \cdot \Delta y_t \tag{15}$$

where $a_k, \ldots, a_0, b_k, \ldots, b_0$ is a group of fix coefficients which are set offline.

The 2D-Turntable's pitch and yaw angular deviation can be obtained by the following formula:

$$\mathbf{\Delta\theta} = \begin{pmatrix} \Delta\theta_x \\ \Delta\theta_y \end{pmatrix} = \begin{pmatrix} k_1 \\ k_2 \end{pmatrix} \begin{pmatrix} \Delta\hat{x}_{t+1} \\ \Delta\hat{y}_{t+1} \end{pmatrix} \tag{16}$$

where $\Delta\theta_x$ and $\Delta\theta_y$ is respectively the pitch and yaw angular deviation.

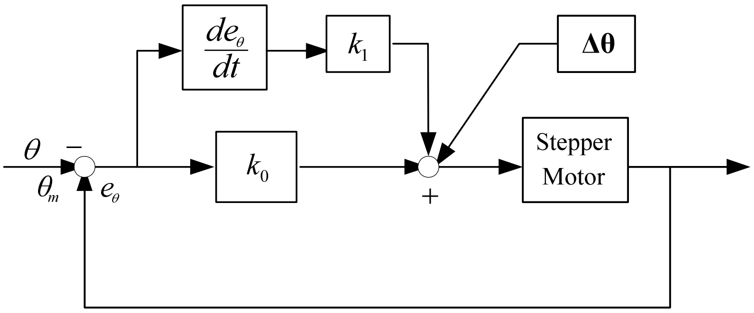
A reliable PD controller is used for the tracking system, and the angular deviation $\mathbf{\Delta\theta}$ obtained from linear prediction is used as feed forward compensation, then the final control algorithm is:

$$\mathbf{u} = \mathbf{k}_0 * (\mathbf{\theta} - \mathbf{\theta}_m) + \mathbf{k}_1 * \frac{d\mathbf{e}_\theta}{dt} + \mathbf{\Delta\theta} \tag{17}$$

where $\mathbf{\theta}$ represents the angles of the instruction, $\mathbf{\theta}_m$ represents the angle of the feedback. The scheme of the feed-forward compensation based PD controller is shown in Figure 8.

$$\mathbf{e}_\theta = \mathbf{\theta} - \mathbf{\theta}_m \tag{18}$$

Figure 8. The scheme of the feed forward compensation-based PD controller.



5. Experiments Section

5.1. Parameter Setting

The kernel function has an important influence on the experimental results. In this paper the Epanechnikove kernel profile is used as:

$$K_E(x)=\begin{cases} C_k(1-\|\frac{x}{z}\|^2) & \|\frac{x}{z}\|^2\le 1 \\ 0 \quad \|\frac{x}{z}\|^2>1 \end{cases} \tag{19}$$

where z = 128 is the bandwidth of MS tracking algorithm which is decided by the size of the target. x actually represents the distance between the effective pixels and the center of the tracking region.

The quantization function b is:

$$b(x)=\begin{cases} 1 & 0<x<60 \\ 2 & 110\le x\le 160 \\ 3 & 160<x<210 \\ 4 & 210\le x<255 \end{cases} \tag{20}$$

Region of interest (ROI) is 20 × 20.

5.2. Experiments Results

Four experiments have been implemented to test the above target tracking scheme. A wireless remote control car (as shown in Figure 9) has been used to simulate a moving target. The experiments include four cases: in case of tracking with the traditional MS (as shown in Figure 10), tracking in case of poses variation with the proposed method (as shown in Figure 11), tracking in case of partial occlusion with the proposed method (as shown in Figure 12), tracking in the case of poses variation in a complex scene with the proposed method (as shown in Figure 13).

From the following tracking image sequence, we can find two rectangular boxes. One represents the center of the optical system; the other represents the target location in the current image. The distance between the two rectangular boxes are used as errors to control the 2D-turntable. When the target is in stop condition, the two rectangular boxes should overlap.

From the following experiments results, we can conclude that the TTS designed in this paper has good robustness to the target pose variation and occlusion. The system totally processes an image in 18.21 ms, in which the fast MS consuming 14.6 ms, TM consumes 1.83 ms, other algorithms consume 1.78 ms. The Target Tracking Scheme time-consuming statistical table is as shown in Table 4. The final image processing speed is >50 frame/s. The experiment results indicate our approach to tracking a moving target is fast and robust. However, this proposed algorithm needs to be comprehensively evaluated in a wider database. Although the tracking results are promising in certain situations, further development and more evaluation is anticipated in severe image clutter and occlusion situations.

Table 4. TTS time-consuming statistical.

Algorithm	Time
Fast MS (10 iteration)	14.6 ms
TM	1.83 ms
Other	1.78 ms
Total	18.21 ms

Figure 9. The wireless remote car.

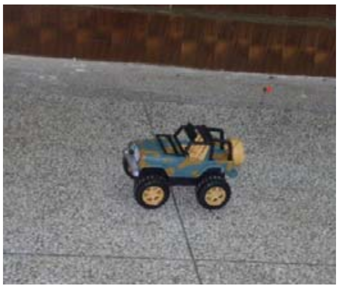
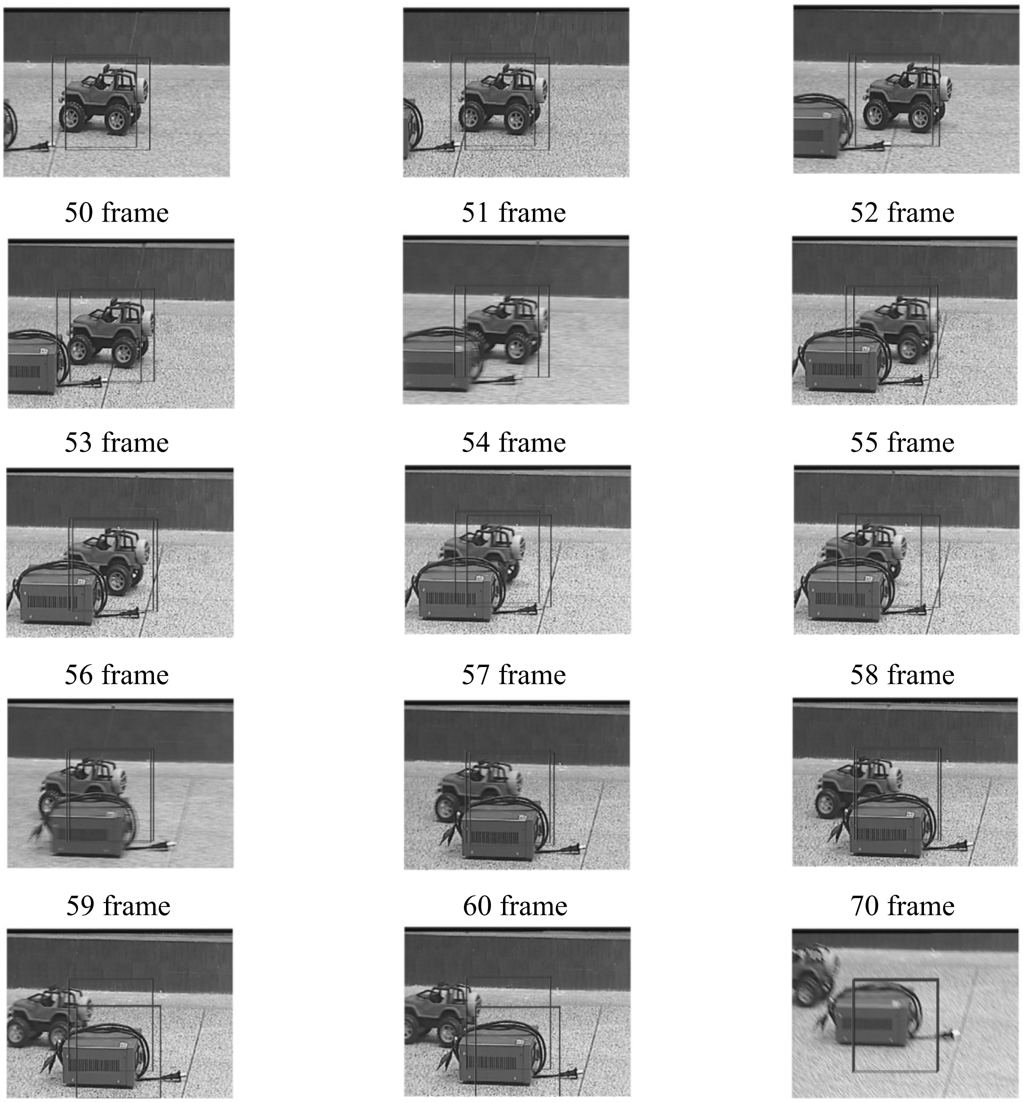


Figure 10. Tracking with the traditional MS.



50 frame

51 frame

52 frame

53 frame

54 frame

55 frame

56 frame

57 frame

58 frame

59 frame

60 frame

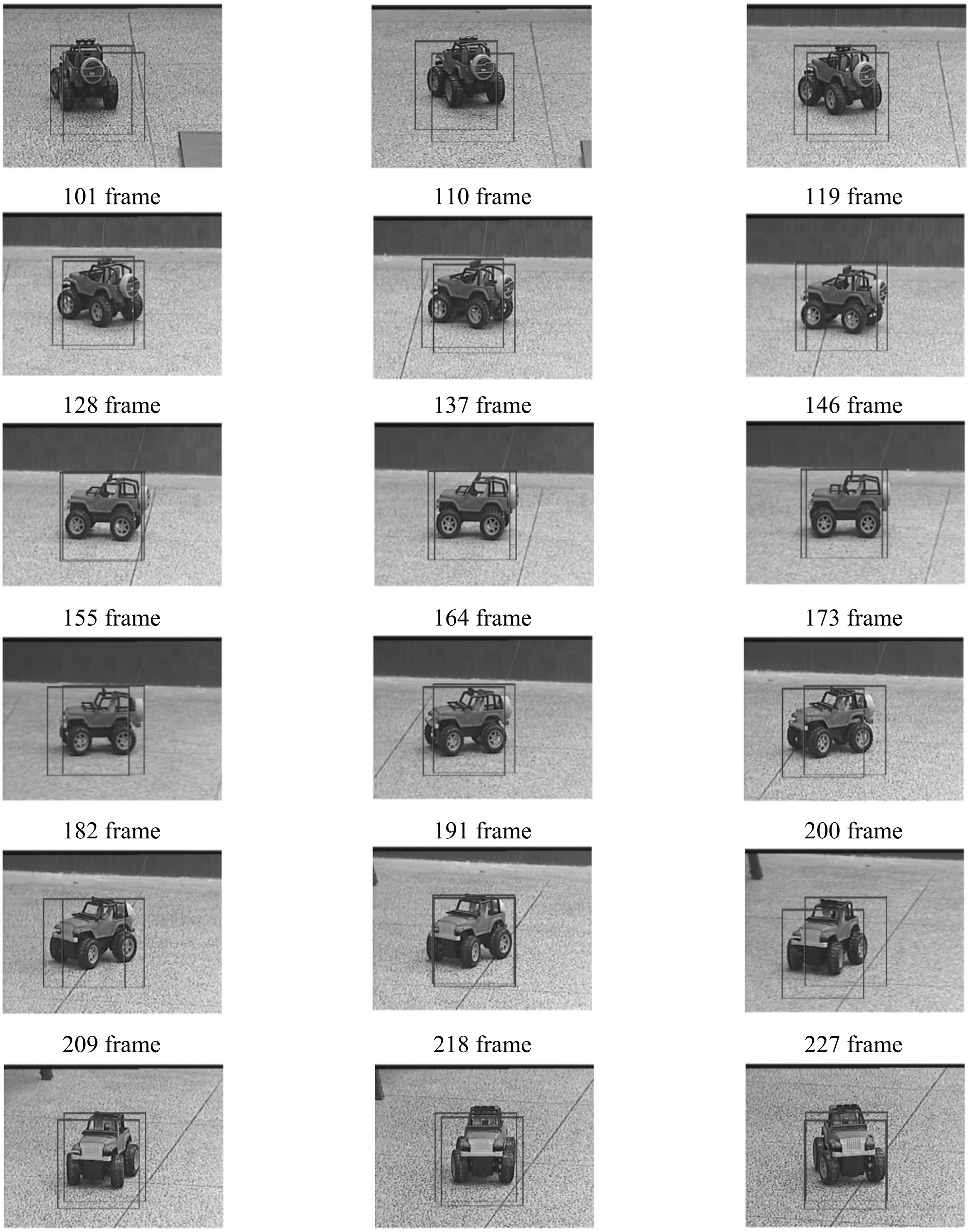
70 frame

71 frame

72 frame

73 frame

Figure 11. Tracking with proposed method in case of poses variation.



101 frame

110 frame

119 frame

128 frame

137 frame

146 frame

155 frame

164 frame

173 frame

182 frame

191 frame

200 frame

209 frame

218 frame

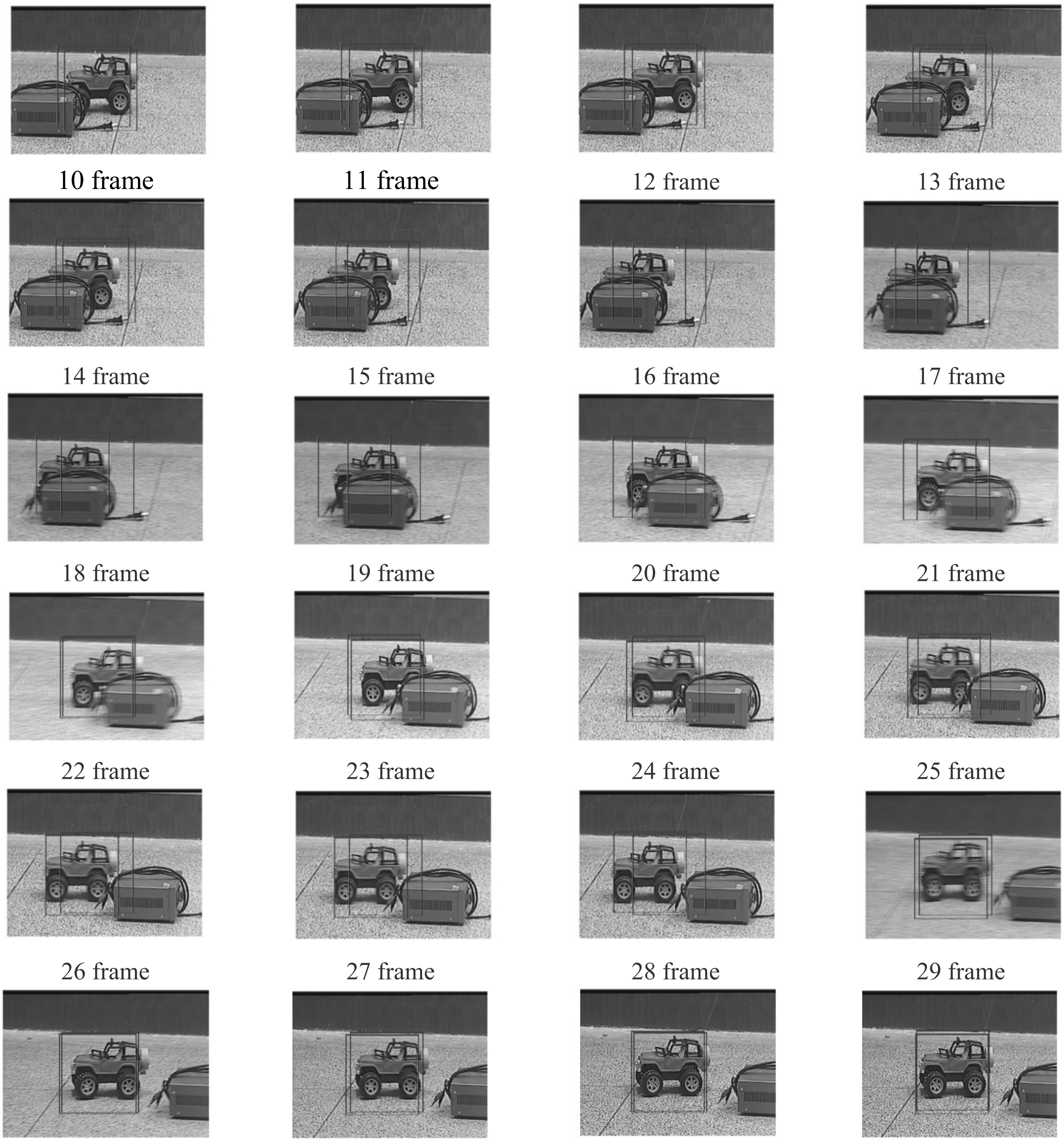
227 frame

236 frame

245 frame

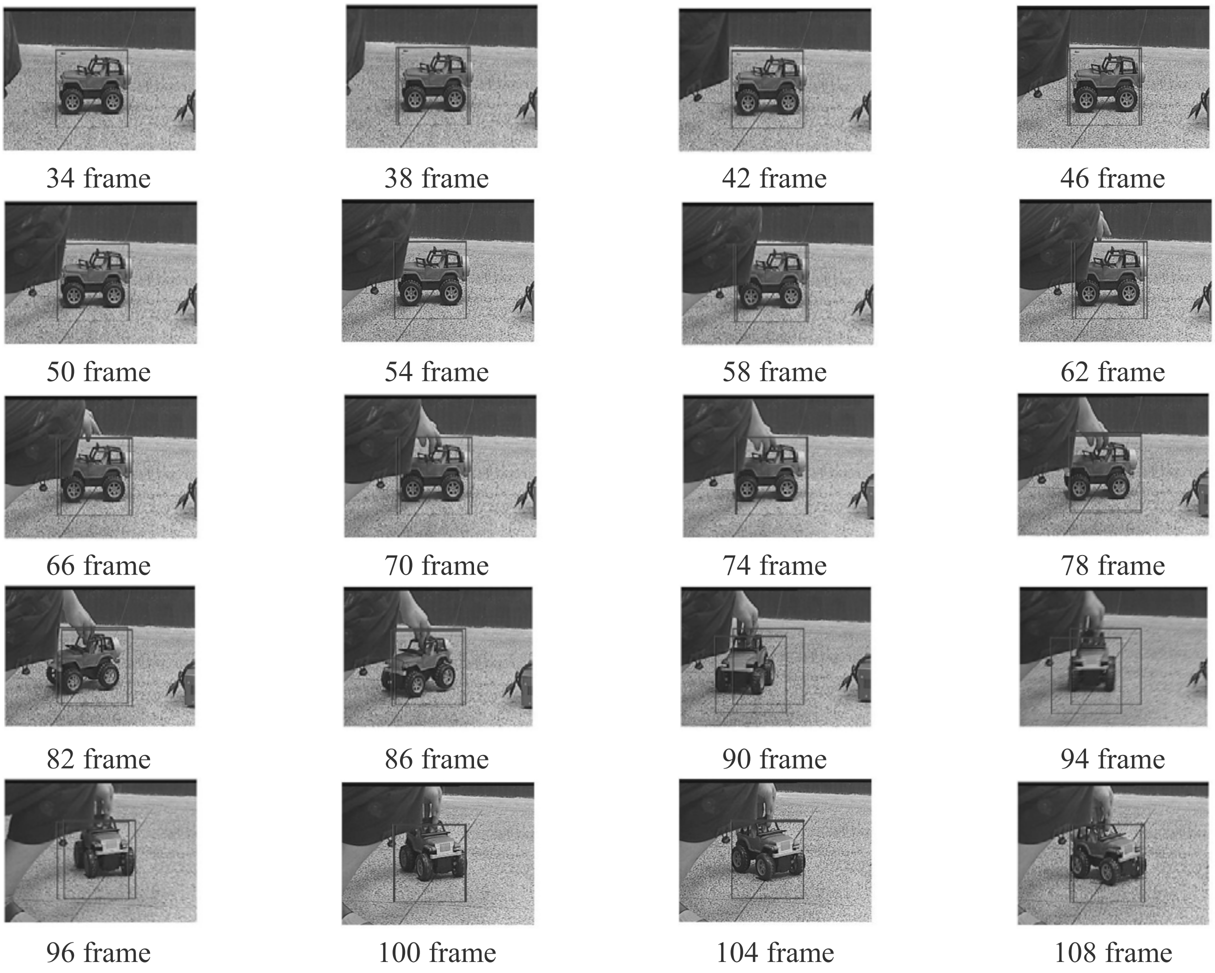
254 frame

Figure 12. Tracking with proposed method in case of occlusion.



10 frame
11 frame
12 frame
13 frame
14 frame
15 frame
16 frame
17 frame
18 frame
19 frame
20 frame
21 frame
22 frame
23 frame
24 frame
25 frame
26 frame
27 frame
28 frame
29 frame
30 frame
31 frame
32 frame
33 frame

Figure 13. Tracking with proposed method in case of poses variation under complex scene.



34 frame 38 frame 42 frame 46 frame

50 frame 54 frame 58 frame 62 frame

66 frame 70 frame 74 frame 78 frame

82 frame 86 frame 90 frame 94 frame

96 frame 100 frame 104 frame 108 frame

6. Conclusions and Future Work

In this paper, a balanced scheme between the robustness and real-time performance of a TTS is presented. A novel robust tracking algorithm combining the MS with template match (TM) has been proposed, which has a good robustness to target pose variation, partial occlusion, and a fast MS-based target tracking scheme is designed and implemented. The hardware-in-loop simulation shows that the image signal processing speed is >50 frame/s. The TTS presented in this paper utilized s common CCD camera to realize acquisition of images, but for some special applications infrared CCD sensors or heterogeneous sensors are used, so IR CCD or heterogeneous sensor-based fast target tracking techniques would be a future research direction.

Acknowledgements

The authors thank the anonymous reviewers for their professional questions and constructive comments that help improve the quality of this manuscript. This paper was funded by the National Natural Science Foundation of China (No. 60904089) and the Fundamental Research Funds for the Central Universities of China. This research work was mainly carried out in Northwest Polytechnical University. The authors wish to acknowledge the important contributions made by Fengqi Zhou, Jun Zhou and Yanning Zhang.

References

1. Bakhtari, A.; Benhabib, B. An active vision system for multitarget surveillance in dynamic environments. *IEEE Trans. Syst., Man, Cybern., Part B: Cybern.* **2007**, *37*, 190–198.
2. Collins, R.T.; Liu, Y.; Leordeanu, M. Online selection of discriminative tracking features. *IEEE Trans. Pattern Anal. Mach. Intell.* **2005**, *27*, 1631–1643.
3. Ali, A.; Kausar, H.; Khan, M.I. Automatic visual tracking and firing system for anti aircraft machine gun. In *Proceedings of International Bhurban Conference on Applied Sciences & Technology*, Islamabad, Pakistan, 19–22 January 2009.
4. Watanabe, Y.; Lesire, C.; Piquereau, A.; Fabiani, P. System development and flight experiment of vision-based simultaneous navigation and tracking. In *Proceedings of AIAA InfoTech Aerospace 2010*, Atlanta, GA, USA, 20–22 April 2010.
5. Lamberti, F.; Sannagianluca, A.; Paravati, G. Improving robustness of infrared target tracking algorithms based on template matching. *IEEE Trans. Aerosp. Electron. Syst.* **2011**, *47*, 1467–1480.
6. Murray, D.; Basu, A. Motion tracking with an active camera. *IEEE Trans. Pattern Anal. Mach. Intell.* **1994**, *16*, 449–459.
7. Shirzi, M.A.; Hairi-Yazdi, M.R. Active tracking using intelligent fuzzy controller and kernel-based algorithm. In *Proceedings of 2011 IEEE International Conference on Fuzzy Systems*, Taipei, Taiwan, 27–30 June 2011.
8. Luo, R.C.; Chen, T.M. Autonomous mobile target tracking system based on grey-fuzzy control algorithm. *IEEE Trans. Ind. Electron.* **2000**, *47*, 920–931.
9. Zhou, H.; Yuan, Y.; Shi, C. Object tracking using SIFT features and mean shift. *Comput. Vis. Image Underst.* **2009**, *113*, 345–352.
10. Shi, J.; Tomasi, C. Good features to track. In *Proceeding of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Seattle, WA, USA, 21–23 June 1994.
11. Chen, W.-G. Simultaneous object tracking and pedestrian detection using hogs on contour. In *Proceeding of 2010 IEEE 10th International Conference on Signal Processing (ICSP)*, Beijing, China, 24–28, October 2010; pp. 813–816.
12. Cox, I.; Hingorani, S. An efficient implementation of reid's multiple hypothesis tracking algorithm and its evaluation for the purpose of visual tracking. *IEEE Trans. Pattern Anal. Mach. Intell.* **1996**, *18*, 138–150.
13. Sawasaki, N.; Morita, T.; Uchiyama, T. Design and implementation of high-speed visual tracking system for real-time motion analysis. In *Proceedings of the 13th International Conference on Pattern Recognition*, Vienna, Austria, 25–29 August 1996; pp. 478–483.
14. Omachi, S. Fast template matching with polynomials. *IEEE Trans. Image Process.* **2007**, *16*, 2139–2149.
15. Hajdu, A.; Pitas, I. Optimal approach for fast object-template matching. *IEEE Trans. Image Process.* **2007**, *16*, 2048–2057.
16. Zhang, X.; Liu, H.; Li, X. Target tracking for mobile robot platforms via object matching and background anti-matching. *J. Robot. Auton. Syst.* **2010**, *58*, 1197–1206.
17. Shen, C.; Brooks, M.J.; van den Hengel, A. Fast global kernel density mode seeking with application to localization and tracking. In *Proceedings of the 10th IEEE International Conference on Computer Vision*, China, Beijing, 17–21 October 2005.

18. Comaniciu, D.; Ramesh, V. Kernel-based object tracking. *IEEE Trans. Pattern Anal. Mach. Intell.* **2003**, *25*, 564–577.
19. Comaniciu, D.; Meer, P. MEANSHIFT: A robust approach toward feature space analysis. *IEEE Trans. Pattern Anal. Mach. Intell.* **2002**, *24*, 603–619.
20. Zhou, H.; Taj, M. Target detection and tracking with heterogeneous sensors. *IEEE J. Sel. Top. Signal Process.* **2008**, *2*, 503–513.
21. Martinez-del-Rincon, J.; Orrite-Urunuela, C.; Herrero-Jaraba, J.E. An efficient particle filter for color-based tracking in complex scenes. In *Proceeding of IEEE Conference on Advanced Video and Signal Based Surveillance*, London, UK, 5–7 September 2007; pp. 176–181.
22. Yang, W.; Li, J.; Shi, D.; Hu, S. Mean shift based target tracking in FLIR imagery via adaptive prediction of initial searching points. In *Proceeding of 2005 Second International Symposium on Intelligent Information Technology Application*, Shanghai, China, 20–22 December 2008; pp. 852–855.
23. Sullivan, J.; Blake, A.; Isard, M.; MacCormick, J. Object localization by Bayesian correlation. In *Proceedings of the Seventh International Conference on Computer Vision*, Kerkyra, Greece, 20–27 September 1999; pp. 1068–1075.
24. Black, M.J.; Jepson, A.D. EigenTracking: Robust matching and tracking of articulated objects using a view-based representation. *Int. J. Comput. Vis.* **1998**, *26*, 63–84.
25. Levy, A.; Lindenbaum, M. Sequential karhunen- loeve basis extraction and its application to image. *IEEE Trans. Image Process.* **2000**, *8*, 1371 – 1374.
26. Lin, R.; Yang, M. Object tracking using incremental fisher discriminant analysis. In *Proceedings of the 17th International Conference on Pattern Recognition*, Cambridge, UK, 23–26 August 2004; pp. 757–760.
27. Cheng, Y. Mean shift, mode seeking, and clustering. *IEEE Trans. Pattern Anal. Mach. Intell.* **1995**, *17*, 790–799.
28. Bentley, J.L. Multidimensional binary search trees in database applications. *IEEE Trans. Softw. Eng.* **1979**, *SE-5*, 333–340.
29. Salakhutdinov, R.; Roweis, S. Adaptive overrelaxed bound optimization methods. In *Proceedings of the Twentieth International Conference on Machine Learning (ICML-2003)*, Washington, DC, USA, 21–24 August 2003; pp. 664–671.
30. Kim, J.S.; Yeom, D.H.; Joo, Y.H. Fast and robust algorithm of tracking multiple moving objects for intelligent video surveillance systems. *IEEE Trans. Consum. Electron.* **2011**, *57*, 1165–1170.
31. Xu, L.; Jordan, M.I. On convergence properties of the EM algorithm for Gaussian mixtures. *Neural Comput.* **1996**, *8*, 129–151.
32. Ortiz, L.E.; Kaelbling, L.P. Accelerating EM: An empirical study. *Uncertain. Artif. Intell.* **1999**, 512–521.
33. You, Z.; Sun, J.; Xing F. A novel multi-aperture based sun sensor based on a fast multi-point MEANSHIFT (FMMS) algorithm. *Sensors* **2011**, *11*, 2857–2874.