デジタルカメラにおける物体追尾および動き予測技術

Object Tracking and Motion Prediction Technology for Digital Camera

関　海克[*]　　　新浪　紀克[**]
Haike GUAN　　Norikatsu NIINAMI

要　旨

デジタルカメラを用いて，高速で動く被写体を確実に撮影するためには，正確にターゲット物体にフォーカスする必要があるが，被写体は頻繁に変形，ランダムに動くので，被写体を追従，位置特定することには困難が伴う．本論文では，高精度に高速で物体追尾，被写体位置を特定する方法を提案する．本提案手法はYUV特徴量を用いたパーティクルフィルタ物体追尾を行うことで，被写体が変形，ランダムに動き，画面から消失して，再度現れるケースでも高精度で追尾できる．さらに並列処理のアルゴリズムにより，リアルタイム追尾を実現できる．リコー製デジタルカメラCXに実機実装し，動く被写体追尾の定量評価実験を行い，本技術の有効性を示した．

また，動き予測のアルゴリズムも開発した．物体追尾の結果を用いて，一定の時間後の物体動きを予測し，被写体の位置特定ができる．この技術により，カメラ撮影タイムラグによって発生したフォーカスエラーを削減することができる．デジタルカメラで撮影した動画でシミュレーションを行い，動き予測の効果を確認した．

ABSTRACT

Focusing at a moving object accurately is difficult and important to take a photo of the target successfully in a digital camera. Because the object often moves randomly and changes its shape frequently, position and distance of the target should be estimated at real-time so as to focus at the objet precisely.

We propose a new method of real-time object tracking to do auto-focus for moving target in digital camera. Particle filter is used to deal with problem of the target object’s random movement and shape change. Color and edge features are used as measurement of the object’s states. Parallel processing algorithm is developed to realize real-time object tracking easily in hardware environment of the digital camera. Motion prediction algorithm is also proposed to remove focus error caused by difference between tracking result and target object’s real position. Simulation and experimental results in digital camera demonstrated effectiveness of the proposed method.

*　リコーICT研究所　システム研究センター
System Research & Development Center, Ricoh Institute of Information and Communication Technology
**　リコーICT研究所　フォトニクス研究センター
Photonics Research & Development Center, Ricoh Institute of Information and Communication Technology

本稿は，SPIE学会に帰属の著作権の利用許諾を受け，SPIE学会誌，Proceedings of SPIE Vol. 9400, 940009 (2015) に掲載した論文を基に作成した．

1. Introduction

Focusing at a moving object accurately is difficult and important to take a photo of the target successfully in digital camera. As shown in Fig. 1, to take a clear photo of a bird marked by a rectangle, it is necessary to focus at the bird accurately and quickly. Because the bird flies fast and changes its shape frequently, it is difficult to focus at it precisely. Although digital camera has auto-focus function, one often takes poor photo because of miss-focusing the moving target object.

1-1 Problems of moving object auto-focus

Two auto-focus methods are often used in digital cameras. One is phase difference auto-focus method. Another is contrast auto-focus method.

1-1-1 Phase difference auto-focus

Many single lens reflex cameras use phase difference method to do AF (auto-focus). The method uses optical lenses and sensor to detect object distance and focus position. The lenses and sensor are only used for AF, which are different from main lens and sensor.

Two separate lenses are used to image target object as two images at different places in AF image sensor plane. Distance between the two images in AF sensor image plane is used to detect object distance and focus position.

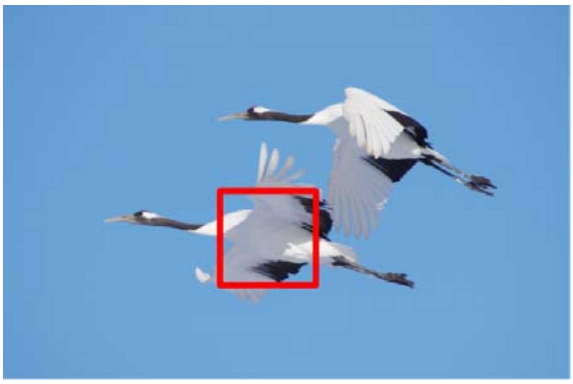


Fig. 1 Auto-focus of moving target object. Rectangle is focus target area.

Because the optical system is separated from main lens and sensor, object distance and focus position can be detected in real-time.

Although it is a real-time auto-focus method, focus area should be set on the target object correctly. If target object moves out of preset target area, camera will focus at background or other objects and target object image will be blurred.

1-1-2 Contrast auto-focus

Compact digital cameras often use contrast to do auto-focus. Video stream for AF (Auto-Focus) is inputted as camera lens scans forward and back to search focus position where target image area has the highest contrast. Since the lens should scan forward and back, it takes time and auto-focus speed is low. The target object area is often set at center of image plane at first. Then auto-focus is made in this area and focus position is locked at detected focus position until photo is taken. If object moves quickly, it will go out of preset focus area when photo is taken. Camera will focus at background or other objects.

By both phase difference and contrast auto-focus methods, if target object moves quickly, camera will often fail to focus at the target. In this case, camera will focus at background or other object so that target object image is blurred.

Because it is difficult to set focus area of the moving object in real-time, these two AF methods often fail to focus at the moving target.

We propose a new real-time object tracking method to set moving target object area both accurately and quickly to do AF successfully to reduce failure photos.

1-2 Object tracking for auto-focus

Many object tracking methods have been proposed[1). Object tracking is to find target object area in each frame after setting the object area at first frame. Mean-shift is used to do object tracking[2,3). Mean-shift object tracking uses local search by iteration. This method can find target area quickly. But it may fall into local minimum and fail to track the object when background is cluttered or target object moves quickly. Kalman filter is also applied to object tracking[1). Kalman filter object tracking method assumes that object states change is linear and statistical distribution of the object is Gaussian. This method can track object very well when state change is linear and statistical distribution is Gaussian. But it is difficult to deal with problems of target object's random movement and shape change when auto-focus is made in digital camera because this is a non-linear, non-Gaussian case.

Particle filter is used to do object tracking. Because particle filter tracking uses non-linear and non-Gaussian model, it can deal with problems of random movement and shape change[4-7). However calculation cost of this method is very high. Particle filter method uses many particles to estimate states of target area. It takes a lot of time to calculate features and weights of the particles. Embedding particle filter object tracking algorithm in digital camera faces problem of calculation cost. A digital camera has a very low speed CPU and it is very difficult for the CPU to perform real-time object tracking accurately by this method.

We propose to use *YUV* color information and particle filter to do real-time object tracking in digital camera. Parallel processing algorithm is developed to realize real-time particle filter object tracking easily in hardware environment of the digital camera. Features and weights of particles are calculated in parallel.

The proposed object tracking algorithm is embedded in digital camera, using SIMD processor to enforce parallel real-time processing. The hardware can be used efficiently by our algorithm. Features of multiple particles are calculated at same time in parallel. Several look-up tables are designed to make calculation of particle filter tracking more quickly without losing tracking accuracy. Both CPU and SIMD processor are used together to do real-time particle filter object tracking. Experimental results in digital camera demonstrate effectiveness of the proposed method.

1-3 Motion prediction

When target object moves quickly, focus error will be caused by difference states of object between tracking results and states of real position when the photo is taken. After getting tracking result, it takes some time to control camera to do exposure and release shutter to take photo. During this period of time, the target object has moved to other position. Focus error will be caused by position change.

We propose using motion prediction to remove focus error caused by the time difference. 3 dimensional object tracking in *(x, y, z)* space is used to do object motion prediction. Polynomial fitting prediction and magnification prediction methods are used to do motion prediction.

Video files taken by the digital camera are used to do experiments to confirm effectiveness of 3 dimensional tracking in *(x, y, z)* space and object movement prediction.

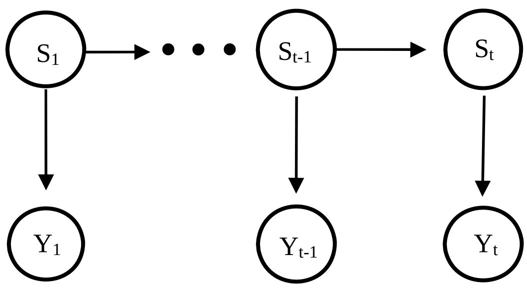


Fig. 2 Prediction, measurement and estimation of object states in Particle filter object tracking.

This paper is organized as follows. In Section 2 Real-time particle filter tracking and the parallel processing method we proposed for auto-focus in digital camera is explained. Section 3 describes target object motion prediction by 3D tracking. Section 4 presents embedding real-time object tracking algorithm in digital camera, experimental results of real-time object tracking made by digital camera, experimental results of object movement prediction made by video files of digital camera are explained. Finally, Section 5 gets conclusion and discusses the results and open issues for future research.

2. Real-time object tracking for auto-focus of digital camera

As shown in Fig. 1, to do auto-focus of target object area marked by the rectangle, it is necessary to specify the area at real-time in digital camera. In our proposed method, real-time object tracking is used to get target area for each image frame. Because the target object moves both quickly and randomly, changes its shape frequently, it is difficult to track the target. Cluttered background and similar objects next to the target make it easy to lose the target. In our method, particle filter object tracking is used to deal with these problems[4-7].

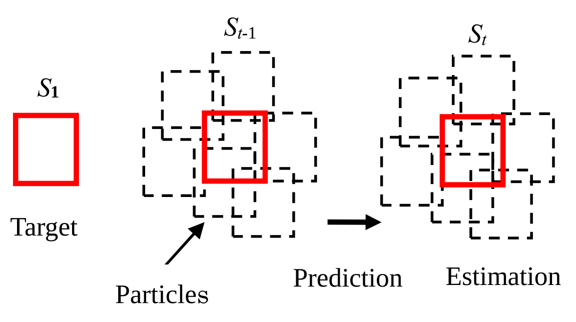


Fig. 3 Propagation and estimation of particles. Rectangle of solid line means estimated target area. Rectangles of broken line are sampling particles set.

2-1 Particle filter object tracking

Particle filter object tracking uses prediction, measurement and estimation processes to do object tracking. As shown in Fig. 2, state of object at time t, S_t, is estimated by prediction of state S_{t-1} and measurement results of Y_t at time t. S_1 is start state. As shown in Fig. 3, target area is selected and set at first frame in which object state is S_1. Object tracking starts from next frame. State of the object of the target area at time t, S_t, is estimated by a weighted sample set to approximate probability distribution of the object, which is expressed by Equation (1). Estimated state $E[S]$ is a weighted average of sampling particles. w_k is weight coefficient and S_k is state of particle k.

$$E[S]=\sum_{k=0}^{N} w_k S_k \qquad (1)$$

In our algorithm, state S_t is defined by Equation (2) as bellow,

$$S=\{x, y, V_x, V_y, W, H, M\} \qquad (2)$$

Where (x, y) are position of target area, V_x and V_y are speed of the target, in x and y direction. W, H are width and height of target area. M is magnification of target area.

Color features of *(Y, U, V)* are used to measure state of the target in our proposed method. Because *(Y, U, V)* features can be obtained in real-time in digital camera from hardware, this feature data can be inputted more quickly than other features, such as *(R, G, B)* features. Component of *(Y, U, V)* can be used separately to increase processing speed, making trade-off between performance and processing speed. For example, we can only select *Y*, intensity information, to get the highest processing speed. Or make combination of *(Y, U)*, *(Y, V)* or *(U, V)* to get mid-way performance and processing speed. Relation between *(Y, U, V)* and color component *(R, G, B)* is expressed in Equation (3).

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.147 & -0.289 & 0.436 \\ 0.615 & -0.515 & 0.100 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (3)$$

Color histogram of *(Y, U, V)* in target area is used as object model. Rectangle area is used as target area which is defined by width W and height H. Histogram is calculated by using weight kernel function to reduce influence of background, which is expressed in Equation (4).

$$p_y^u = f \sum_{i=1}^{I} k\left(\frac{\|y - x_i\|}{M}\right) \delta\left[h(x_i) - u\right] \quad (4)$$

p_y^u is value of color histogram of bin u. The color histogram is calculated in a rectangle area of width W by height H. y is position of the rectangle's center. x_i is position of image pixel in the rectangle area. f is normalization parameter. M is magnification of image. δ is Kronecker delta function. k(r) is weight kernel function which is defined as bellow by Equation (5). Weight kernel function is used to reduce influence of background. k(r) is 1 at center of the target area and 0 at edge of the target area.

$$k(r) = \begin{cases} 1 - r^2 : r < 1 \\ 0 : otherwise \end{cases} \quad (5)$$

Similarity $\rho[p, q]$ between color histograms p and q is defined by Equation (6).

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

Distance d between color histogram features is defined by Equation (7).

$$d = \sqrt{1 - \rho[p, q]} \quad (7)$$

We use weight function w_k of particle in Equation (1), which is defined by

$$w_k = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{d^2}{2\sigma^2}} = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(1-\rho[p_s(k), q])}{2\sigma^2}} \quad (8)$$

σ is parameter of variance. q is target histogram and $p_s(k)$ is histogram of candidate area. Particle propagation formula is defined by Equation (9).

$$S_t = AS_{t-1} + n_{t-1} \quad (9)$$

S_t is state at time t and S_{t-1} is state at time t-*1*. A is propagation equation. n_{t-1} is random noise. Gaussian noise is added here.

Particles are resampled according to weight function w_k. Particle of small weight is deleted. Particle of big weight is replicated. Total particle number is not changed.

Model is updated by Equation (10).

$$q_t^{(u)} = (1 - \alpha) q_{t-1}^{(u)} + \alpha p^{(u)} \quad (10)$$

where $p^{(u)}$ is model of estimated state. α is rate parameter of estimated model. $q_t^{(u)}$ is model at time t. If similarity between estimated state $p^{(u)}$ and $q_{t-1}^{(u)}$ model at time *t-1* is bigger than a threshold value, model at time $q_t^{(u)}$ is updated by Equation (10). Otherwise, model $q_t^{(u)}$ will not be updated.

2-2 Parallel processing for real-time particle filter tracking

As shown in Equation (1), to get tracking results by particle filter, weight function of N particles should be calculated. Because many particles around target area should be used, as shown in Fig. 3, calculation cost is very high. As analyzed from Equation (4) to Equation (8), to calculate weight function of Equation (8), at first, color histogram should be calculated, which is the most time consuming part. Then similarity and weight functions should be calculated. CPU speed and data transfer speed of digital camera are much lower than that of PC. It is very difficult to only use CPU to do real-time processing of particle filter tracking.

As shown in Equation (1) and Fig. 3, N particles are independent, features of which can be calculated separately. We use this property to do parallel real-time processing. SIMD processor of digital camera is used to calculate features of particles in parallel, which can reduce processing time to realize real-time processing.

3. Motion prediction by 3D object tracking

3-1 Time delay and focus error

When tracking result is obtained, target area is set by the result and phase difference AF or contrast AF method can be used to auto-focus. However, it takes some time to shift lens to focus, release shutter, exposure and take the photo. Therefore there is time difference Δt between time we get tracking result and time we take the photo. Because target object moves quickly, during time Δt target will move to different place from the area of tracking results.

In this case, camera will miss focus at background and image of the target will blur. We proposed a 3 dimensional tracking method to predict object movement to remove focus error.

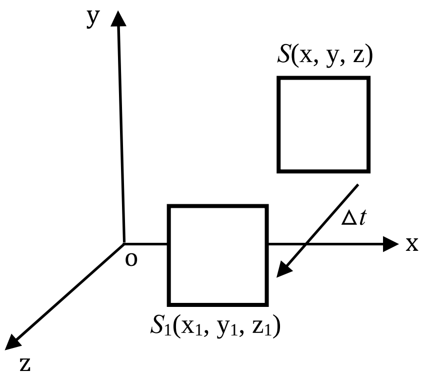


Fig. 4 Motion prediction of target area in *(x, y, z)* space from state of tracking results to state of real position when photo is taken. Δt is time difference.

3-2 3D object tracking and motion prediction

As shown in Fig. 4, object tracking results is $S(x, y, z)$, where (x, y) are position of target area and z is object distance between target object and the digital camera. $S_1(x_1, y_1, z_1)$ is state when the photo is taken. Δt is time difference. State S_1 is predicted by 3D object tracking.

Fig. 5 shows how camera lens images target area onto image plane and how camera parameters affect motion prediction results. From similar triangles relation, we have equations of (11) and (12).

$$\frac{H}{u} = \frac{H_v}{v} \tag{11}$$

$$\frac{H}{u_1} = \frac{H_{v1}}{v_1} \tag{12}$$

H is height of target area and H_v is height of its image on image plane. u is target object distance from camera when tracking results is obtained and v is its image distance. u_1 is distance of object when photo is taken and v_1 is its image distance. From Equation (11) and (12) we have Equation (13).

$$u_1 = u\frac{H_v}{H_{v1}}\frac{v_1}{v} \tag{13}$$

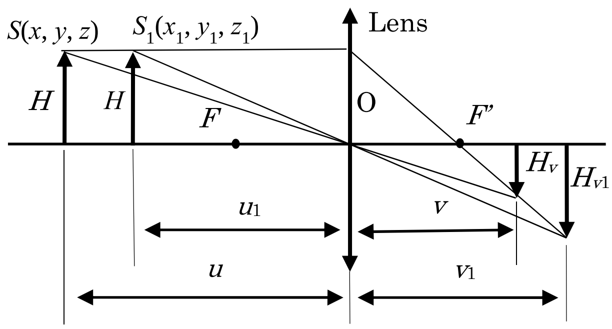


Fig. 5 Optical parameters for motion prediction.

Using lens formula, we have equations of (14) and (15).

$$\frac{1}{u} + \frac{1}{v} = \frac{1}{F} \tag{14}$$

$$\frac{1}{u_1} + \frac{1}{v_1} = \frac{1}{F} \tag{15}$$

Combining Equations (13), (14) and (15), we have Equation (16), from which u_1, real distance of object when photo is taken, can be obtained. We know height of object in image H_v, focal length F of camera and distance u of object, where u can be obtained from contrast AF or phase difference AF when we get tracking results. Distance calculation error caused by object height prediction error is analyzed in Equation (17).

When object is far away and distance u is bigger enough than focal length F, Equation (16) can be approximated by Equation (18). Distance calculation error caused by object height prediction error is analyzed in Equation (19).

$$u_1 = u\frac{H_v}{H_{v1}} + F\frac{H_{v1} - H_v}{H_{v1}} \tag{16}$$

By differentiating two sides of Equation (16), we have Equation (17).

$$\Delta u_1 = (F-u)\frac{H_v}{H_{v1}^{\ 2}}\Delta H_{v1} \tag{17}$$

$$u_1 \approx u\frac{H_v}{H_{v1}} \tag{18}$$

By differential of two sides of Equation (18), we have Equation (19).

$$\Delta u_1 = u\frac{H_v}{H_{v1}^{\ 2}}\Delta H_{v1} \tag{19}$$

In Equation (16), H_{v1} of object size in image plane should be predicted. H_{v1} can be calculated by Equation (9), propagation formula. From tracking results, we can get state information by Equation (2). M is magnification of target area. Image height H_{v1} can be predicted by M and H_v.

To reduce error caused by noise, we propose to use polynomial approximation to predict H_{v1}. Coefficient of polynomial are calculated from last several frames' tracking results. As shown in Fig. 6, height of target object is predicted by polynomial. H_{v1} can be predicted by polynomial coefficients.

Polynomial formula is described in Equation (20). H_m is tracking result height in image frame m. t is time and a_n is polynomial coefficients. Equation (20) can be rewritten by Equation (21), where **H** is data vector, which is expressed in Equation (22). **A** and **T** are expressed in Equation (23) and (24). Polynomial coefficient **A** can be obtained by least squares approximation as Equation (25).

$$H_m = \mathrm{a}_0 + \mathrm{a}_1 t + \mathrm{a}_2 t^2 + \ldots + \mathrm{a}_n t^n \tag{20}$$

$$\mathbf{H} = \mathbf{T} \times \mathbf{A} \tag{21}$$

$$\mathbf{H} = \begin{bmatrix} H_0 \\ H_1 \\ . \\ H_m \end{bmatrix} \tag{22}$$

$$\mathbf{A} = \begin{bmatrix} a_0 \\ a_1 \\ . \\ a_n \end{bmatrix} \tag{23}$$

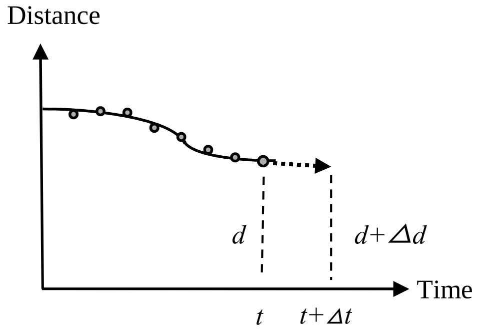


Fig. 6 Polynomial approximation to predict movement of object.

$$\mathbf{T} = \begin{bmatrix} 1 & t_0 & t_0^{\ 2} & . & t_0^{\ n} \\ 1 & t_1 & t_1^{\ 2} & . & t_1^{\ n} \\ 1 & t_2 & t_2^{\ 2} & . & t_2^{\ n} \\ . & . & . & . & . \\ 1 & t_m & t_m^{\ 2} & . & t_m^{\ n} \end{bmatrix} \tag{24}$$

$$\mathbf{A} = (\mathbf{T}^{\mathrm{T}}\mathbf{T})^{-1}\mathbf{T}^{\mathrm{T}}\mathbf{H} \tag{25}$$

H_{v1}, height of target area can be predicted by polynomial coefficient **A** and Equation (20) when time difference Δt is inputted. Distance of predicted object is calculated by Equation (16) or Equation (18).

4. Experiment and results

4-1 Embedding real-time object tracking in digital camera

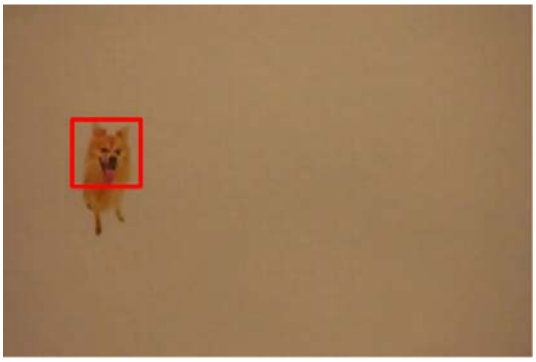
Real-time particle filter object tracking algorithm is embedded in Ricoh digital camera CX. The camera has main CPU of 162 MHz to be used to run object tracking program. The camera has a SIMD processor which has 1024 registers of 512 bit width.

Propagation function which includes calculating color histogram of Equation (4), similarity of Equation (6) and weight function of Equation (8) are embedded by CPU and SIMD processor, features of particles are calculated in parallel.

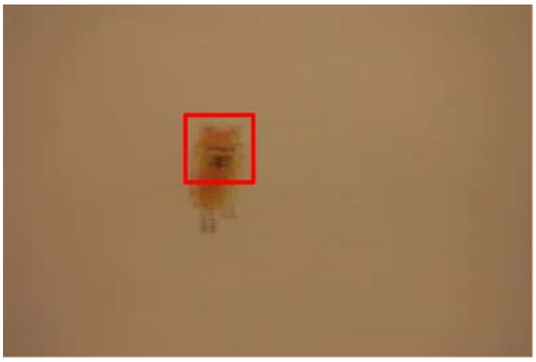
Several look-up tables are used to reduce processing time. In Equation (4), distance r is calculated by a 2 dimensional look-up table. A one-dimensional look-up table is used to calculate square root function in Equation (6). A one-dimensional look-up table is used to calculate exponential function in Equation (8) to get particle weight function.

4-2 Experimental results of real-time object tracking in digital camera

Object tracking experiment by the camera was made to show effectiveness of the proposed method. Processing time and tracking performance were tested by the camera.



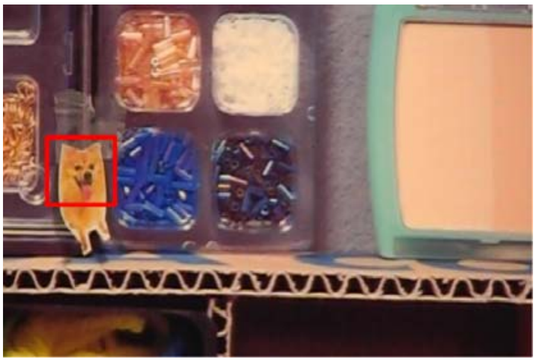
(a)



(b)

(c)

Fig. 7 Evaluation test of real-time object tracking in front of simple background of different frames (a), (b), (c).



(a)

(b)

(c)

Fig. 8 Evaluation test of real-time object tracking in front of cluttered background of different frames (a), (b), (c).

As shown in Fig. 7 and Fig. 8, target object of a dog plate is put on a stage which is moved periodically left and right. The target object can be moved at preset frequency to test object tracking performance. Image size is 320x240 pixels. Target area size is 60 by 60 pixels, which is marked by rectangle. Particle number is set as 60. Fig. 7 is a test case in which target object is set in front of a simple background. Fig. 8 shows a test case in which target object is set in front of a cluttered background.

Table 1 shows testing results of processing time of particle filter object tracking in the digital camera. Reselect function include resampling particles according to its weight values. This function does not take much time. CPU + SIMD did not increase processing speed. Observe function includes calculating color histogram, similarity and weight function. This function is the most time consuming part. CPU + SIMD parallel processing mode is 5 times faster than CPU only mode. For Model Update function, parallel processing mode is 5 times faster than CPU only mode. Total processing time is 44.3 ms, 5 times faster than that of CPU only mode. In digital camera, processing time is about 60 ms, which includes data transfer time.

Performance of proposed real-time particle filter tracking is tested by digital camera. Target object was fixed on a stage unit which is moved periodically left and right. The stage can be moved at preset frequency. As shown in Fig. 7 and Fig. 8, the target object was fixed on the stage which is moved with frequency of 0.5 Hz, 1.0 Hz, 1.5 Hz and 2.0 Hz. Targets in front of both simple and cluttered background are tracked and tested. At first, target area was set by rectangle area, as shown in Fig. 7 (a) and Fig. 8 (a). Then turn on switch of the stage to start moving the target object left and right. If tracking is successful, rectangle will be marked on the target area. Total times of successfully tracking the target are counted during a period of time. Because the object is moved left and right, it is a non-linear motion which is near random movement. Images of Fig. 7 (b) and Fig. 8 (b) are blurred because stage moved the most quickly when the dog plate is at center of image.

Table 1　Processing time test results of CPU mode and SIMD + CPU mode.

Functions	Processing Time (msec)	
	SIMD + CPU	CPU
Reselect	2	1.8
Observe	41.5	205.9
Model Update	0.8	3.5
TOTAL	44.3	221.2

Table 2　Tracking performance evaluation test results of parallel real-time object tracking in digital camera.

Background condition	Object tracking results (Success times/Total times, Test time:16 seconds)			
	0.5 Hz	1.0 Hz	1.5 Hz	2.0 Hz
Simple background	8/8	16/16	24/24	16/32
Cluttered background	8/8	16/16	12/24	16/32

(a)

(b)

(c)

Fig. 9　Tracking results for movement prediction. Rectangle is object tracking result. (a) frame 1, (b) frame 60, (c) frame 100.

We set the period test time of 16 seconds. Test results are shown in Table 2. As frequency increased, successful times of tracking decreased. Cluttered background also reduced tracking accuracy.

4-3 Experimental results of object motion prediction

Experiments of motion prediction were made by videos taken by digital camera to show effectiveness of the proposed method.

As shown in Fig. 9, target was moved from a known distance towards digital camera. As the object was moved, video was recorded by digital camera. With the video file, object tracking and object movement prediction experiment is made. Target object area is selected as rectangle area. Then object tracking of the target area and movement prediction are made with PC.

Particle filter tracking described in section 2 is used to track the target. Particle number of 2000 is used. Image size is 1280x720 pixels. Movement prediction is made with proposed method of section 3. Tracking results are used for object movement prediction.

Experimental results are shown in Fig. 10. Vertical axis is distance of target from camera. Horizontal axis is frame number which is proportional to time. Graphs show tracking results, magnification prediction results and polynomial approximation prediction results. Target Object is moved starting from distance of 70cm from camera. Distance from tracking results is calculated by Equation (18). Distances of object predicted by both magnification prediction and polynomial approximation prediction are shorter than the distance of tracking results, because the object was being moved towards camera.

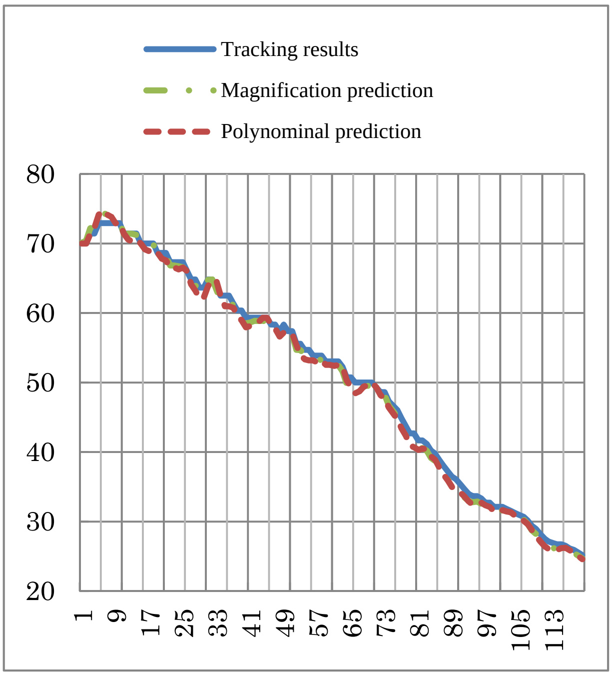


Fig. 10 Target object movement prediction results. Vertical axis is distance (cm) from camera. Horizontal axis is frame number of video.

Magnification prediction used tracking results directly of last frame to predict movement of next time. As shown in Fig. 2 and Fig. 3, states of Equation (2) can be estimated from measurement data by particle filter tracking. M is magnification of target area size to that of last frame. It is used directly to predict size of target area in next frame at Δt time later. In Fig. 10, distance of the target object of 2 frames ahead is predicted by magnification prediction method. Curve of solid line is tracking results, broken line means prediction results.

Polynomial approximation method is used to do object movement prediction. Several frame tracking results are used to do polynomial approximation. In Fig. 10, 5 frames' tracking results were used to calculate 3 order polynomial coefficients. Size of target area, 2 frames ahead, is predicted by Equation (20), and distance is calculated by Equation (18). Time difference Δt can be gotten from camera parameter. Focus position can be preset by the predicted object distance to reduce focus error caused by time difference. Polynomial approximation prediction results are plotted by curve of broken line.

Object distances of Δt time later predicted by both magnification prediction method and polynomial approximation method are shorter than the distance of current tracking results because the object was moving to the camera.

5. Conclusion

A real-time parallel object tracking method is proposed to track a quickly moving target object for digital camera auto-focus. This method can solve problems of shape change and random movement of target object. This algorithm is embedded in digital camera. SIMD registers are used to do parallel processing of calculating features of particles. Movement prediction method is proposed to remove focus error caused by time difference between tracking results and real state of target object when photo is taken. Experimental results show that by using SIMD parallel registers, processing speed increases 5 times, which makes it possible to do real-time tracking. Processing time of less than 60 ms in digital camera with CPU of only 162 MHz is obtained. Experimental results of tracking target moved by stage of controlled frequency show effectiveness of the proposed methods.

Experiment of movement prediction is made to show effectiveness of the proposed method. Both magnification prediction and polynomial approximation prediction methods were tested in the experiments. Movement prediction effect was confirmed. Tracking results of target area size H and magnification M are affected by noise, which reduce movement prediction accuracy. For future work, we will improve this.

References

1) A. Yilmaz, O. Javed, M. Shah: Object Tracking: A Survey, *ACM Computing Surveys*, Vol. 38, No. 4, pp. 1-45 (2006).

2) D. Comaniciu, V. Ramesh, P. Meer: Real-time Tracking of Non-rigid Objects Using Mean Shift, *Proceedings IEEE Conference of Computer Vision and Pattern Recognition*, Vol. 2, pp. 142-149 (2000).

3) D. Comaniciu, V. Ramesh, P. Meer: Kernel-Based Object Tracking, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 25, No. 5, pp. 564-577 (2003).

4) M. Isard, A. Blake: Condensation- Conditional Density Propagation for Visual Tracking, *International Journal of Computer Vision*, Vol. 29, No. 1, pp. 5-28 (1998).

5) M. S. Arulampalam et al.: A Tutorial on Particle Filters for Online Nonlinear/non-Gaussian Bayesian Tracking, *IEEE Transaction on Signal Processing*, Vol. 50, No. 2, pp. 174-188 (2002).

6) K. Nummiaro, E. Koller-Meier, L. V. Gool: An Adaptive Color-based Particle Filter, *Image and Vision Computing*, Vol. 21, No. 1, pp. 99-110 (2003).

7) J. Czyz, B. Ristic, B. Macq: A Particle Filter for Joint Detection and Tracking of Color Objects, *Image and Vision Computing*, Vol. 25, No. 8, pp. 1271-1281 (2007).