INTERVAL LEAST-SQUARES FILTERING WITH APPLICATIONS TO ROBUST VIDEO TARGET TRACKING

Baohua Li[†]*, Changchun Li*[†]*, Jennie Si*[†]*, and Glen P. Abousleman*[‡]

[†] Arizona State University, Tempe, AZ 85287
[‡] General Dynamics C4 Systems, Scottsdale, AZ 85257

ABSTRACT

An interval recursive least-squares (RLS) filter is developed to produce state estimation and prediction by narrow intervals, in which true values are contained with high confidence. The interval filter is robust to variations of the filter parameters and state observations. Using this filter, a video target tracking algorithm is proposed to estimate the target position in each frame. The tracking algorithm is robust to both noise in the video sequence and estimation error of the affine model. The experiments show that the tracking algorithm using the interval RLS filter outperforms that using an RLS filter.

Index Terms— Robust filter, interval estimation, recursive least-squares, video target tracking

1. INTRODUCTION

Kalman filtering [1] and recursive least-squares (RLS) filtering [2] are efficient algorithms to estimate and predict states in discrete-time linear dynamic systems. They have wide applications such as radar and image tracking, airplane navigation, and chemical process control. However, using both methods, state estimation and prediction are sensitive to filter parameters and state observations. Even if those parameters and/or observations have small errors, estimation and prediction could be far from true values. In reality, it is challenging to obtain accurate parameters and observations due to noise, modeling error, or measurement limitation. Specifically, for tracking video targets using an RLS filter [5], when video quality is degraded by noise, the estimation errors of the affine model parameters are nontrivial. Therefore, the target position estimation may deviate from true values in the current frame, the predicted search region can drift away from the target in the next frame, and the transformed target template may be different from the target appearance in the current frame. Thus, their accumulative effects can cause a loss of target lock. To address the issues of sensitivity and uncertainty, parameters and observations are allowed to vary within given ranges, resulting the development of robust methods. For instance, with interval filter parameters and state observations, an interval Kalman filter [3] produces robust interval state estimation and prediction. However, in applications such as tracking video targets, state noise stream and observation noise stream can be temporally dependent or cross-correlated. The interval Kalman filter may not preform well in this case. It also degenerates when noise covariances are impossible to estimate in real time.

In this paper, an efficient RLS filter using interval arithmetics [4] is developed to give narrow interval state estimation and prediction containing true values with high confidence, where filter parameters and observations are assumed to vary in closed intervals. This interval RLS filter is robust to variations of parameters and observations. It does not require noise covariances, and with convoluted noise dependency, the performance may greatly exceed that of interval Kalman filtering [2]. A video target tracking algorithm is then proposed, which uses the interval RLS filter to estimate the target position. It is robust to the error of the affine model. The performance is evaluated by tracking a rigid and deformable target in the respective two real-world video sequences.

2. INTERVAL RLS FILTERING

An $m \times n$ interval matrix is denoted by $M^I = [M^I(i,j)]_{m\times n}$ or $M^I = [\underline{M}, \overline{M}]_{m\times n}$, where $M^I(i,j)$ represents an interval element at the i^{th} row and the j^{th} column, $\underline{M}, \overline{M} \in \Re^{m\times n}$ represent the lower and upper limits, respectively. A linear time-varying system with interval parameter matrices and observations is formulated as follows:

$$X_{k+1} = A_k^I X_k + \xi_k, \quad (k = 0, 1, 2, ...,) \tag{1}$$
$$Y_{k+1}^I = B_{k+1}^I X_{k+1} + \eta_{k+1}. \tag{2}$$

In the state equation (1), X_k, ξ_k, A_k^I and X_{k+1} represent an $n \times 1$ state, state noise, and $n \times n$ interval dynamic matrix at time k, $n \times 1$ state at time $k+1$, respectively. In the observation equation (2), Y_{k+1}^I, η_{k+1} and B_{k+1}^I represent an $m \times 1$ interval observation and observation noise, $m \times n$ interval observation matrix at time $k+1$, respectively.

Since the associative law does not hold for interval matrices, new rules for multiplication of three and four interval

The work was supported by the NSF and by General Dynamics C4 Systems.

matrices are regulated to narrow results as follows:

$$W_1^I \cdot W_2^I \cdot W_3^I = ((W_1^I W_2^I) W_3^I) \cap (W_1^I (W_2^I W_3^I)) \tag{3}$$

$$W_1^I \cdot W_2^I \cdot W_3^I \cdot W_4^I = ((W_1^I \cdot W_2^I \cdot W_3^I) W_4^I) \cap (W_1^I (W_2^I \cdot W_3^I \cdot W_4^I)) \tag{4}$$

Let $\Omega = \{Z_1 Z_2 Z_1' : Z_1 \in Z_1^I, Z_2(\text{Positive semidefinite}) \in Z_2^I\}$. Define $F_1(Z_1^I, Z_2^I) = Z_1^I \cdot Z_2^I \cdot (Z_1^I)'$, $F_2(Z_1^I, Z_2^I) = Z^I$ where at $i = j$,

$$\begin{aligned} Z^I(i,j) &= \sum_{n=1}^{m} (Z_1^I(i,n))^2 Z_2^I(n,n) \\ &+ 2 \sum_{k>n} Z_1^I(i,n) Z_1^I(i,k) Z_2^I(n,k), \end{aligned} \tag{5}$$

and at $i \neq j$,

$$Z^I(i,j) = \sum_{n=1}^{m} Z_1^I(i,n) Z_1^I(j,n) Z_2^I(n,n) + \sum_{k>n} (Z_1^I(i,n) Z_1^I(j,k) + Z_1^I(i,k) Z_1^I(j,n)) Z_2^I(n,k). \tag{6}$$

The square of an interval used in (5) is defined by $v^I = [\underline{v}, \overline{v}]$,

$$(v^I)^2 = \begin{cases} [\underline{v}^2, \overline{v}^2] & \overline{v} \geq \underline{v} \geq 0 \\ [\overline{v}^2, \underline{v}^2] & \underline{v} \leq \overline{v} \leq 0 \\ [0, \max\{\underline{v}^2, \overline{v}^2\}] & \underline{v} < 0 < \overline{v} \end{cases}.$$

In interval arithmetics, sub-distributivity law holds, i.e., for intervals $v_1^I, v_2^I, v_3^I, (v_1^I + v_2^I) v_3^I \subseteq v_1^I v_3^I + v_2^I v_3^I$. Thus, the interval expression inside the second sum of (6) is better. In fact, $\Omega \subseteq F_3(Z_1^I, Z_2^I) = \bigcap_{i=1}^{2} F_i(Z_1^I, Z_2^I)$. Since all possible matrices, $Z_1 Z_2 Z_1'$, in Ω are positive semidefinite,

$$F_4(Z_1^I, Z_2^I) = (F_3(Z_1^I, Z_2^I) \cap W^I) \bigcap (F_3(Z_1^I, Z_2^I) \cap W^I)', \tag{7}$$

where

$$W^I(i,j) = \begin{cases} [0, +\infty] & i = j \\ F_3(Z_1^I, Z_2^I)(i,j) & i \neq j \end{cases}.$$

$F_4(Z_1^I, Z_2^I)$ still includes all $Z_1 Z_2 Z_1'$, and it is narrower than any other $F_i(Z_1^I, Z_2^I)$.

Denote an identity matrix by I and a forgetting factor by λ. Using (3), (4), (7), positive definite properties of some matrices, and intersection of multiple different interval matrix expressions extended from the essentially same matrix formulations, interval filter is given below to obtain narrow state estimation at time $k+1$, $X_{k+1/k+1}^I$, and one-step-ahead prediction at time $k+2$, $X_{k+2/k+1}^I$, which is extended from the RLS filter [2].

Initialization:

$$X_{0/0}^I, P_0^I = \alpha I (\alpha > 0), \lambda. \tag{8}$$

Recursion:

$$\begin{aligned} U_{k+1}^{BAP} &= [(\lambda I + F_4(B_{k+1}^I A_k^I, P_k^I))^{-1} \cap W_{k+1}^{BAP}] \\ &\bigcap [(\lambda I + F_4(B_{k+1}^I A_k^I, P_k^I))^{-1} \cap W_{k+1}^{BAP}]', \end{aligned} \tag{9}$$

where

$$W_{k+1}^{BAP}(i,j) = \begin{cases} [\varepsilon, +\infty] & i = j \\ (\lambda I + F_4(B_{k+1}^I A_k^I, P_k^I))^{-1}(i,j) & i \neq j \end{cases},$$

and $\varepsilon \approx 0$ and $\varepsilon > 0$,

$$L_{k+1}^I = P_k^I \cdot (A_k^I)' \cdot (B_{k+1}^I)' \cdot U_{k+1}^{BAP} \tag{10}$$

$$\begin{aligned} U_{k+1}^{LBA} &= L_{k+1}^I \cdot B_{k+1}^I \cdot A_k^I \\ &\bigcap P_k^I F_4((A_k^I)', F_4((B_{k+1}^I)', U_{k+1}^{BAP})) \end{aligned} \tag{11}$$

$$\begin{aligned} U_{k+1}^{ALB} &= A_k^I \cdot L_{k+1}^I \cdot B_{k+1}^I \\ &\bigcap F_4(A_k^I, P_k^I) F_4((B_{k+1}^I)', U_{k+1}^{BAP})) \end{aligned} \tag{12}$$

$$\begin{aligned} U_{k+1}^{AL} &= A_k^I L_{k+1}^I \\ &\bigcap F_4(A_k^I, P_k^I) \cdot (B_{k+1}^I)' \cdot U_{k+1}^{BAP} \end{aligned} \tag{13}$$

$$\begin{aligned} U_{k+1}^{P} &= \lambda^{-1} \{ A_k^I \cdot (I - U_{k+1}^{LBA}) \cdot P_k^I \cdot (A_k^I)' \\ &\bigcap F_4(A_k^I, (I - U_{k+1}^{LBA}) P_k^I) \\ &\bigcap (I - U_{k+1}^{ALB}) F_4(A_k^I, P_k^I) \} \end{aligned} \tag{14}$$

$$P_{k+1}^I = (U_{k+1}^P \cap W_{k+1}^P) \cap (U_{k+1}^P \cap W_{k+1}^P)', \tag{15}$$

where

$$W_{k+1}^P(i,j) = \begin{cases} [\varepsilon, +\infty] & i = j \\ U_{k+1}^P(i,j) & i \neq j \end{cases}$$

$$\begin{aligned} & X_{k+1/k+1}^I \\ =\ & A_k^I (X_{k/k}^I + L_{k+1}^I (Y_{k+1}^I - B_{k+1}^I \cdot A_k^I \cdot X_{k/k}^I)) \\ \bigcap\ & \{ (I - U_{k+1}^{ALB}) \cdot A_k^I \cdot X_{k/k}^I \\ +\ & [(A_k^I \cdot L_{k+1}^I \cdot Y_{k+1}^I) \cap U_{k+1}^{AL} y_{k+1}^I)] \}, \end{aligned} \tag{16}$$

$$X_{k+2/k+1}^I = A_{k+1}^I X_{k+1/k+1}^I. \tag{17}$$

Calibration: If widths of some interval elements in $X_{k+1/k+1}^I$ are greater than a threshold vector, T, then

$$X_{k+1/k+1}^I = C_{k+1}^I, P_{k+1}^I = P_0. \tag{18}$$

Remark 1.
(i) The above interval RLS filter includes solutions of the RLS filter to all possible realizations to the uncertain system. (ii) C_{k+1}^I is assumed to be much narrower than $X_{k+1/k+1}^I$ in (16), and be near to or cover the true state with high confidence. Since the interval estimation in (16) and prediction in (17) may be too wide to provide satisfactory information, the calibration step (18) is added to reinitialize the iterative process, narrow the results, and inhibit divergence. Calibration is necessary only when some interval elements of $X_{k+1/k+1}^I$ in (16) become too wide because the narrow and accurate interval, C_{k+1}^I, usually requires more prior information or other time-consuming methods. (iii) State prediction based on the calibrated interval estimation is also narrowed.

3. VIDEO TARGET TRACKING ALGORITHM

A three-step iterative algorithm for tracking video target using interval RLS filtering is proposed in the section. Let $\langle v^I \rangle = \frac{\underline{v}+\overline{v}}{2}$ representing the center of an interval v^I. In order to describe a transform from a homogeneous Cartesian coordinate $z_1 = [\begin{array}{ccc} x & y & 1 \end{array}]'$ at the k^{th} frame to another coordinate $z_2 = [\begin{array}{ccc} \widetilde{x} & \widetilde{y} & 1 \end{array}]'$ at the $(k+1)^{th}$ frame, an affine model is defined by

$$(z_1 - z_2) = \begin{bmatrix} a_1(k) & a_2(k) & a_0(k) \\ a_4(k) & a_5(k) & a_3(k) \\ 0 & 0 & 1 \end{bmatrix} z_1$$

where parameters $a_0(k)$ and $a_3(k)$ reflect the camera translation, and $a_1(k)$, $a_2(k)$, $a_4(k)$, $a_5(k)$ are for scaling and rotation. A target in the initial frame ($k = 0$) is selected with a bounding box denoted as $\mathcal{B}_t(0)$. The tracking algorithm starts from this initial frame.

The first step is to obtain the affine model. The parameters, $a_i(k)$, are estimated by the fast geometric constraint global motion (fast GCGM) estimation [5] with levels from coarsest to finest. The result is denoted as $a_i^1(k)$, and a small perturbation δ_i^1 is added. Starting from $a_i^1(k) + \delta_i^1$, the fast GCGM estimation is implemented at the finest level to obtain another estimation, $a_i^2(k)$. L-time estimates are generated based on the last one with an added small perturbation. Then, an interval estimation, $a_i^I(k)$, is formed for each parameter, $a_i(k)$, by taking the minimum and maximum values.

The second step is to search the target. An uncertain state equation is established with

$$X_k = [\begin{array}{ccccc} x(k) & v_x(k) & y(k) & v_y(k) & 1 \end{array}]' \qquad (19)$$

$$A_k^I = \begin{bmatrix} 1+a_1^I(k) & 1+a_1^I(k) & a_2^I(k) & a_2^I(k) & a_0^I(k) \\ 0 & 1+a_1^I(k) & 0 & a_5^I(k) & 0 \\ a_4^I(k) & a_4^I(k) & 1+a_5^I(k) & 1+a_5^I(k) & a_3^I(k) \\ 0 & a_4^I(k) & 0 & 1+a_5^I(k) & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \qquad (20)$$

where $x(k), y(k)$ represent the target position, and $v_x(k), v_y(k)$ represent the velocities in x and y coordinates, respectively. In the initial frame, the center of the bounding box is used as an estimate of the target position. Using the state estimation, $X_{k/k}^I$, at the k^{th} frame, the region where the target is most likely to appear is predicted at the $(k+1)^{th}$ frame using the interval RLS filter by $[x_{k+1/k}^I \ y_{k+1/k}^I]' = A_k^I(1,3\|1:5)X_k^I$, where $A_k^I(1,3\|1:5)$ represent a sub-matrix of A_k^I that contains the first and third rows and the first to fifth columns. The target is searched in the rectangular region, $R_{k+1} = \{(x,y){:}x \in x_{k+1/k}^I, y \in y_{k+1/k}^I\}$, by adaptive template matching [5]. The accumulative affine transformation matrix used in adaptive template matching becomes an interval matrix as follows:

$$\Phi_1^I = \begin{bmatrix} a_1^I(0) & a_0^I(0) \\ a_4^I(0) & a_5^I(0) \end{bmatrix}, \Phi_{k+1}^I = \begin{bmatrix} a_1^I(k) & a_2^I(k) \\ a_4^I(k) & a_5^I(k) \end{bmatrix} \Phi_k^I.$$

End points of four elements in each Φ_{k+1}^I construct 16 accumulative affine transformation matrices. Based on those matrices, adaptive template matching is implemented to yield 16 search results. Let $x_s^I(k+1)$ and $y_s^I(k+1)$ be the intervals formed by taking the minimum and maximum values in the respective x-axis and y-axis.

The third step is to estimate the state X_{k+1} and the target position. The estimation $X_{k+1/k+1}^I$ is computed by the interval RLS filter in the $(k+1)^{th}$ frame with (19), (20), and the following matrices:

$$B_{k+1}^I = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}, Y_{k+1}^I = [x_s^I(k+1) \ \ y_s^I(k+1)]'.$$

The position $(\langle X_{k+1/k+1}^I(1,1)\rangle, \langle X_{k+1/k+1}^I(3,1)\rangle)$ is output as the estimate of the target position. It is represented in the frame by a bounding box denoted as $\mathcal{B}_t(k+1)$ with the center as that point and the same size as the initial box $\mathcal{B}_t(0)$. The above three steps proceed iteratively.

Remark 2.

(i) Since perturbations can help fast GCGM estimation to exit wrong solutions or local minima, a_i^I is most likely to be near to or cover the true value. (ii) This algorithm is extended in each step by intervals from that using the RLS filter. The search region using the interval RLS filter is adaptive in size to variations of the affine model, unlike that using an RLS filter [5] with fixed size. As a result, the algorithm is robust to the error of the affine models, and outperforms that using the RLS filter.

4. EXPERIMENTAL RESULTS

Two real-world video sequences were used to evaluate the performance of both the interval RLS filter and the new tracking algorithm. The tracking result is compared with the algorithm using an RLS filter [5]. The videos used in the experiments are gray-scale with a size of 240×320 pixels in each frame. One sequence is a natural driving scene with a car, where the target is the car indicated by the blue bounding box in (b) of Fig. 1 as an example of rigid targets. The other is a scene of two people walking with occlusion by tree, where the target is one person indicated by the blue bounding box in (b) of Fig. 2 as an example of deformable targets.

The parameters used in the proposed tracking algorithm are given as follow. The times estimating the affine model L is set to be 3. The perturbation is 5% of $a_i^l(k)(l < L)$. At the inital frame, velocities $v_x(0), v_y(0)$ are estimated by the interval $[-5,5]$(pixels/frame). When tracking a car (person), in calibration process, the threshold is 20 (10) pixels for the width of $X_{k+1/k+1}^I(1,1)$ along the x-axis, and 15 (20) pixels for the width of $X_{k+1/k+1}^I(3,1)$ along y-axis, and $C_{k+1}^I = [\langle x_s^I(k+1)\rangle \ [-5,5] \ \langle y_s^I(k+1)\rangle \ [-5,5] \ 1]'$. The thresholds are comparable half the size of the initial bounding box along each axis. The center, $(\langle x_s^I(k+1)\rangle, \langle y_s^I(k+1)\rangle)$, is used to calibrate the target position estimation, since it is most likely to be near to the true target position. The target is assumed to move relatively slowly. The interval $[-5, 5]$(pixels/frame) is selected as the velocity estimate.

The results for tracking the car (person) are shown in Fig. 1 (Fig. 2). In (a) and (b) of both figures, the estimated target position and search region are represented by the blue bounding box $\mathcal{B}_t(k)$ and green rectangular shadow, respectively. Using an RLS filter, the car (person) was lost in the $36^{th}(71^{th})$ frame while using an interval RLS filter it was tracked in that frame. The tracking performance for the video sequence is evaluated by an overlap rate defined as $r(k) = S(\mathcal{B}_t(k) \cap \mathcal{B}_g(k))/S(\mathcal{B}_g(k))$, where $S(\cdot)$, $\mathcal{B}_g(k)$, $\mathcal{B}_t(k) \cap \mathcal{B}_g(k)$ represent an area, a bounding box which is drawn to accurately cover the target as a ground truth, and the overlapping region of both bounding boxes. Seeing (c), the overlap rate using RLS filtering is lower than $20\%(40\%)$ or even close to zero starting from about the $36^{th}(60^{th})$ frame, which indicates the target was lost. The overlap rate using inteval RLS filtering is above $50\%(80\%)$ and close to 100% in most of the frames, which indicates the target was always tracked.

Search regions are important for good tracking performance. Too large or too small regions easily cause loss of target lock. RLS filtering generates a fixed-size region which is as large as 200 pixels in the experiments. However, interval RLS filtering generates a region adaptive in size to variations of the affine model. When tracking a car, the sizes shown in (d) of Fig. 1 oscillate between around 100 pixel and 400 pixels. When tracking a person shown in (d) of Fig. 2, the sizes typically fluctuate around 150 pixels.

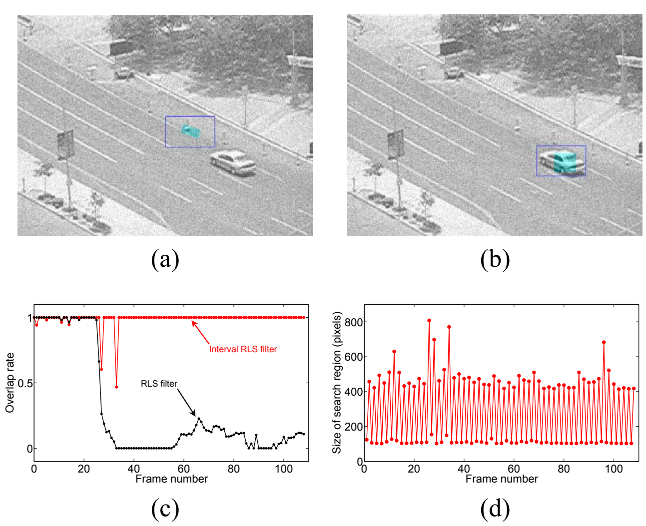

Fig. 1: Performance of tracking a car using an RLS and interval RLS filter: (a) Tracking result at the 36^{th} frame using an RLS filter; (b) Tracking result at the 36^{th} using an interval RLS filter; (c) Overlap rate vs. frame number; (d) Size of search region from an interval RLS filter.

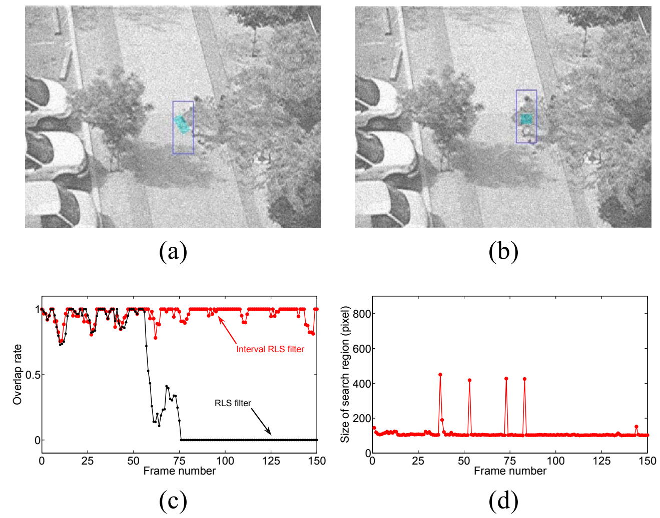

Fig. 2: Performance of tracking a person using an RLS and interval RLS filter: (a) Tracking result at the 71^{th} frame using an RLS filter; (b) Tracking result at the 71^{th} using an interval RLS filter; (c) Overlap rate vs. frame number; (d) Size of search region from an interval RLS filter.

5. CONCLUSION

We developed an interval RLS filter and an associated video target tracking algorithm. The overall system is robust to noise, parameter error and observation error. The experimental results show that the proposed tracking algorithm using the developed interval RLS filter significantly outperforms tracking with the standard RLS filter.

6. REFERENCES

[1] R. E. Kalman, "A new approach to linear filtering and prediction problem," *Transaction of ASME-Journal of Basic Engineering*, Ser. D, 82, pp. 35–45, March 1960.

[2] Yunmin Zhu, "Efficient recursive state estimator for dynamic systems without knowledge of noise covariances," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 35, no. 1, pp. 102–114, January 1999.

[3] G. Chen, J. W. Wang, and L. S. Shieh, "Interval Kalman filtering," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 33, pp. 250–259, 1997.

[4] G. Alefeld, and J. Herzberger, *Introduction to Interval Computations*, New York: Academic Press, 1983.

[5] Changchun Li, Baohua Li, Jennie Si, and Glen P. Abousleman, "Fast video target tracking in the presence of occlusion and camera motion blur," in *Proceedings of SPIE*, vol. 6567, pp. 656707-1–9.