# Detecting tracking errors via forecasting


ObaidUllah Khalid[1,2]
o.khalid@qmul.ac.uk
Andrea Cavallaro[1]
a.cavallaro@qmul.ac.uk
Bernhard Rinner[2]
bernhard.rinner@aau.at

[1] Centre for Intelligent Sensing
Queen Mary University of London
London, UK

[2] Institute of Networked and Embedded Systems
Alpen-Adria-Universitat Klagenfurt
Klagenfurt, Austria


We propose a framework that detects the failures of a tracker using its output only (Figure 1). The framework is based on a state-background discrimination approach that generates a track quality score, which quantifies the ability of the tracker to remain *on target*.

We define a background region around the target and split it into four sub-regions, each with the same size as the state. We then determine the distributions of the state and each of the smaller background regions using colour distribution fields (DF) [5]. A DF represents a smoothed histogram of the image region composed of several layers.We compare the state and background distributions to quantify the similarity between the two regions to produce the track quality score. However, the raw values of the track quality score [3] may have variable ranges, hence limiting its use to specific sequences or trackers only. To address this limitation, we model the track quality score as time series and employ a forecasting model to detect tracking errors.

Let $\mathbf{I} = \{I_t\}_{t=1}^{T}$ be an image sequence and $x_t$ be the estimated state at time $t = 1, ..., T$. Let $S_t$ be the region in $I_t$ defined by $x_t$. Using motion information $\vec{v}_{\Delta t_1}$ from a past short temporal window $\Delta t_1$ and the target state $x_{t-1}$ we select the background region $\mathbf{B}_t$ in $I_t$ (Figure 2). We split $\mathbf{B}_t$ into four smaller equally sized regions, $b_t^a$, each with the same width and height of $S_t$. We then determine the distribution for $S_t$, $d''_{S_t}$, and each of the smaller background regions $b_t^a$, $d''_{b_t^a}$, using colour DFs [5]. The tracking quality score $y_t$ is determined by quantifying the similarity between the distributions of $\mathbf{B}_t$ and $S_t$ using the $L_1$ distance, where low (high) values of $y_t$ indicate similarity (dissimilarity) between the two regions.

We detect tracking errors by employing time series analysis to model $\mathbf{Y} = \{y_t\}_{t=1}^{T}$, a univariate discrete time series, for forecasting. We use the Auto Regressive Moving Average (ARMA) model [1] which is built using past data and forecasts employing both the past and present data. The difference between the forecast and the original returns a re-scaled signal, which highlights only significant changes. We build the forecasting model using data within a past temporal window $\Delta t_2$ and then forecast future values $\hat{y}_{t+l}$ using the forecasting model and its estimated parameters, $\Psi$, over the forecast length $l \geq 1$ at time $t$.

The forecasting error $|\tilde{e}_{t+l}| = y_{t+l} - \hat{y}_{t+l}$ is employed to determine time instants when a tracking error occurs. Since values of $\hat{y}_{t+l}$ are dependent on past values of $y_t$, between $t - \Delta t_2$ and $t$, $|\tilde{e}_{t+l}|$ temporally smooths $y_t$. Significant changes (tracking errors) in the value of $y_t$ are reproduced by $|\tilde{e}_{t+l}|$ and detected for $|\tilde{e}_{t+l}| \geq \tau_1$, where $\tau_1$ is an experimentally derived threshold.

We use a sparse features based tracker [4], to train the proposed approach Detecting Tracking Errors via Forecasting (DTEF) on 20 sequences from dataset D1 and then test DTEF on 20 sequences from the Object Tracking Benchmark (OTB) dataset. Using precision ($P$), recall ($R$), F-score ($F$) and false positive rate ($FPR$), we compare DTEF with two variations of the proposed approach: NAIVE and RAW; one state-of-the-art (SOA) for tracker error detection [3]: CovF; and two SOA features employed for video tracking [2]: RgbHist and RLHist. Results on the OTB dataset (Table 1) indicate that DTEF outperforms RAW and NAIVE in terms of $R$ by 76% and 7%, respectively. DTEF also outperforms other SOA methods in terms of $F$-score, with an overall improvement of 23%, 31% and 36% compared to CovF, RgbHist and RLHist, respectively. Finally, we demonstrate the flexibility of DTEF via an experimental comparison with the respective SOA methods using baseline tracking results of four trackers and sequences from the VOT2014 challenge.

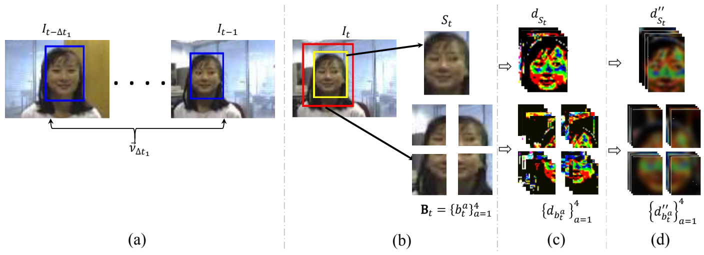


Figure 2: Background and state region selection. (a) $x_{t-\Delta t}$, ..., $x_{t-1}$ (enclosed in the blue bounding boxes) and motion information $\vec{v}_{\Delta t_1}$ over a past temporal window $\Delta t_1$; (b) background region $\mathbf{B}_t$ (enclosed in the red bounding box) and state region $S_t$ (enclosed in the yellow bounding box) selected at frame $I_t$; (c)-(d) distributions of $\mathbf{B}_t$ and $S_t$ represented with colour DF [5]

| | DTEF | NAIVE | RAW | CovF | RgbHist | RLHist |
|---|---|---|---|---|---|---|
| $P$ | .110 | .111 | **.122** | .087 | .083 | .078 |
| $R$ | **.714** | .667 | .405 | **.714** | .595 | .667 |
| $F$ | **.191** | .190 | .188 | .155 | .146 | .140 |
| $FPR$ | .037 | .035 | **.019** | .048 | .042 | .051 |

Table 1: Comparison of tracking error detection performance in terms of $P$, $R$, $F$ and $FPR$, over the OTB dataset. The best results are indicated by bold font. Key — DTEF: Detect Tracking Errors using Forecasting; NAIVE: error detection by forecasting $y_t$ via the Naive forecasting model [1]; RAW: error detection using raw $y_t$ values; CovF [3]; RgbHist [2]; RLHist [2].


[1] J.G. Gooijer and R.J. Hyndman. 25 years of time series forecasting. *Int. Jour. Forecasting*, 22(3):443 – 473, Jul 2006.

[2] J. Ning, L. Zhang, D. Zhang, and C. Wu. Robust object tracking using joint color-texture histogram. *Int. Journal Pattern Recog. Artificial Intell.*, 23(07): 1245–1263, Feb 2009.

[3] J.C. SanMiguel and A. Calvo. Covariance-based online validation of video tracking. *IEEE Elec. Lett.*, 51(3):226–228, Feb 2015.

[4] D. Wang, H. Lu, and M-H. Yang. Online object tracking with sparse prototypes. *IEEE Trans. Image Process.*, 22(1):314–325, Jan 2013.

[5] Y. Wang, H. Chen, S. Li, J. Zhang, and C. Gao. Object tracking by color distribution fields with adaptive hierarchical structure. *The Visual Comput.*, pages 1–13, Nov 2015.


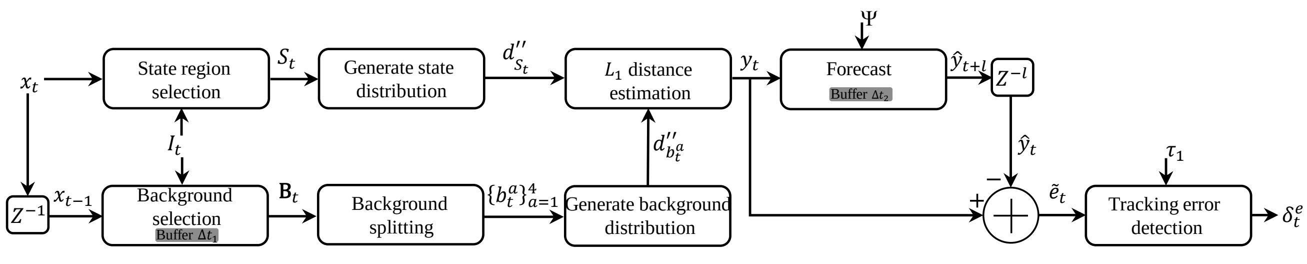


Figure 1: Block diagram of the proposed framework.