

Absorption, Retention and Lock-In: A Closed-Loop Analysis of Tracker-Guided Streaming Subspace Background Models

Ömer Kahyaoğlu

*Mathematical Engineering Department, Faculty of Science and Letters, İstanbul Teknik
Üniversitesi, İstanbul, Türkiye*

Abstract

Streaming low-rank background models keep a forgetting memory of the scene; the same memory absorbs any target that stops, and the hole it leaves when it departs becomes a ghost. We analyse this failure and its cure. First, we show that the weighted principal subspace maintained by a forgetting incremental SVD is, row by row, an exponentially weighted least-squares regression of each pixel on the model's own scores; a stopped target is therefore learned by a per-pixel recursive least-squares ramp whose gain is the leverage of the current frame, independently of the target's size, unless a rank-crossing event comes first. The resulting two-mechanism law predicts absorption from the clean background alone with a median error of 2 frames and a median relative error of 0.130 over 2,880 synthetic runs, whereas the accumulated-leak rule declares 1,200 of these runs never absorbed. Second, we analyse the loop that appears when a tracker protects its targets from being learned: masked coefficients have an exact error expectation, masked pixels are never absorbed but freeze, an expiry clock that the protected error can reset makes its persistence exponential in the protection horizon, the optimal horizon without verification is a quantile of the stop-duration distribution, and a budgeted semantic verifier turns the horizon problem into a classification problem. Third, on CDnet 2014, with an evaluator validated against the official code, a front end selected on a development split and policies selected by a protocol fixed in advance, the verified loop raises the F-measure on intermittent object motion from 0.445 without verifier to 0.614

Email address: kahyaoglu20@itu.edu.tr (Ömer Kahyaoğlu)

(policy chosen leave-one-video-out), against 0.771 with ground-truth masks, at a mean cost of 27.4 ms per frame on one CPU core.

Keywords: background subtraction, robust subspace tracking, incremental SVD, recursive least squares, tracking feedback, closed-loop estimation

1. Introduction

Separating a scene into a low-rank background and sparse foreground is the idea behind robust principal component analysis [1, 2] and a large family of video background models [3, 4, 5]. On a stream, the background must be tracked rather than computed: online methods such as GRASTA [6], ReProCS and NORST [7, 8, 9], incremental principal component pursuit [10] and pROST [11] keep a subspace that follows illumination and slow scene change at a fixed cost per frame. To follow change they must forget, and forgetting has a well-known price: a target that stops is gradually learned as background (*absorption*), and when it leaves, the background it covered is detected as a target (a *ghost*).

The provable trackers bound the share of outliers that any pixel may carry within every analysis window; in NORST it is 0.01 divided by the squared condition number of the coefficient covariance, over windows of $O(r \log n)$ frames [8], so a target that stops for longer than a small fraction of a window is outside every guarantee, and the papers do not describe what happens then. For per-pixel mixture models the time after which a static object becomes background is a known rule of thumb [12], and feeding high-level or tracking information back into background maintenance has been proposed for such models [13, 14, 15, 16]; the deadlock that selective update can cause—a ghost that is never learned—was described in [14]. For low-rank models, detection history has been used to gate a batch low-rank update [17], and online methods weight pixels by their own residual [11, 18]. We did not find an analysis of absorption for subspace models, nor an analysis of the loop that a tracker closes around such a model.

Contributions.

1. An absorption law for forgetting subspace background models (Section 4). Building on the row-wise least-squares structure known from PAST and PETRELS [19, 20], we show that absorption has two mechanisms, a size-independent leverage ramp and a size-dependent rank-

crossing event, and we verify the law on the implemented estimator over 4,320 synthetic episodes.

2. An accuracy analysis of tracker masks (Section 5): an exact error expectation for masked coefficients, and the dynamics of masked rows (no absorption, frozen loadings, release transient).
3. A closed-loop analysis of protection (Section 6): persistence of protected model errors under resettable and non-resettable expiry clocks and under a verifier, and the optimal protection horizon; the resettable-clock mechanism explains a lock-in we observed on real video.
4. A real-data evaluation on CDnet 2014 [21] (Section 7) with a validated evaluator, a declared development/test split, four online-subspace baselines (three from the authors’ code, one re-implemented and validated), retention curves that isolate absorption, and a held-out evaluation of the frozen method.

2. Related work

Robust PCA and video. Principal component pursuit recovers a low-rank matrix and a sparse one under incoherence and random support [1, 2]; its stable version admits dense noise [22], and the inexact augmented Lagrangian method is the standard solver [23]. Moving objects violate the random-support assumption: their supports are contiguous and persistent [24]. Reviews of the video application are [3, 4, 25].

Online and provable subspace tracking. GROUSE and GRASTA move a subspace along the Grassmannian [26, 6]; ReProCS, NORST and their analyses give recovery guarantees under slow subspace change and bounded outlier fractions per window [7, 8, 27, 9]; NORST-miss treats missing entries, again with a bounded missing fraction per row and window and missing sets independent of the coefficients [28]. Incremental PCP [10], pROST [11] and online variants with structured priors [29, 18, 30] complete the picture; the survey [31] covers streaming PCA with missing data, including PAST [19] and PETRELS [20], whose row-wise recursive least-squares form we use. pROST states the conflict between keeping stopped objects and learning revealed background explicitly [11].

Stopped objects, ghosts and feedback. Wallflower lists sleeping persons and moved objects among the canonical difficulties of background maintenance

[32]. Mixture models [33, 34, 12] absorb a static object after a time set by the learning rate [12]. Selective update and ghost handling [14], high-level feedback maps [13], a tracker’s predictive distribution used to gate a per-pixel background update [15], dual backgrounds [16, 35], and stopped- or abandoned-object detectors [36, 37, 38] address the same phenomenon for per-pixel models. Universal unsupervised methods such as SuBSENSE and PAWCS [39, 40] use pixel-level feedback of their own detections.

Perturbation and numerics. Eigenvector perturbation [41, 42, 43] and spiked random-matrix results [44, 45, 46] underlie rank selection by a noise edge; exponential weighting of sample covariances has been studied in [47]. Brand’s incremental SVD [48, 49] is the estimator we analyse; an error analysis of a related incremental SVD appeared recently [50].

3. Model and notation

Fig. 1 shows the system and Table 1 collects the notation. A frame on the work grid is $x_t \in \mathbb{R}^d$. The cleaning operator turns it into a column c_t that the model learns: outliers are soft-thresholded, imputed by the current fit, or released (learned raw) after T_{rel} frames; pixels under tracker masks are imputed. The model maintains the leading k left singular vectors $U = [u_1, \dots, u_k]$ and singular values $s_1 \geq \dots \geq s_k$ of the weighted data matrix

$$X_t = [c_t, \sqrt{f} c_{t-1}, f c_{t-2}, \dots], \quad X_t X_t^\top = \sum_{j \geq 0} f^j c_{t-j} c_{t-j}^\top, \quad (1)$$

updated by Brand’s incremental SVD with truncation to k [48]. A component is *counted* if $s_i^2 > \tau = (m\hat{\sigma}(\sqrt{d} + \sqrt{n_{\text{eff}}}))^2$, the Marchenko–Pastur edge of a noise-only window with margin m ; r components are counted, and the background fit of a frame with scores v (estimated robustly, by alternating least squares and soft thresholding) is $U_r v$.

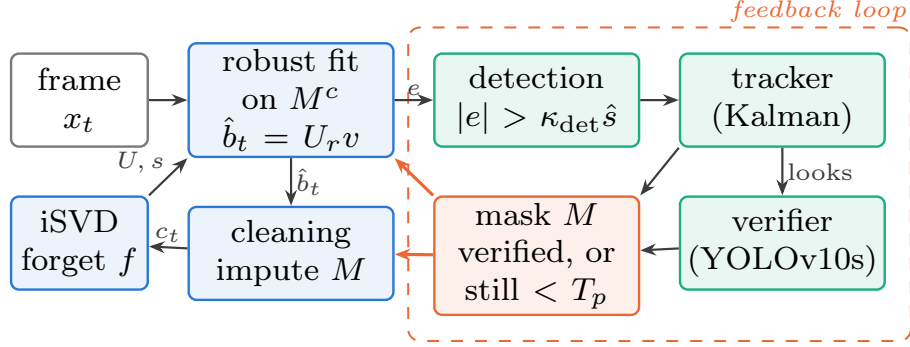


Figure 1: The tracker-guided background model. The protection mask M (orange) closes a loop: it decides which pixels the subspace may learn, and the tracker that produces it sees only what the model fails to explain. Blue: the subspace model; green: perception.

Table 1: Notation.

$x_t \in \mathbb{R}^d, c_t$	frame on the work grid; cleaned column learned at t
$f, n_{\text{eff}} = 1/(1 - f)$	forgetting factor; effective memory (frames)
U, s_i, r, τ	kept basis, singular values, counted rank, rank threshold
$v, z_j = U^\top c_{t-j}$	scores of the current frame; scores of past columns
h_t	leverage of frame t (gain of the per-pixel recursion)
Ω, c	support and contrast (noise units) of a stopped target
$\kappa, \kappa_{\text{det}}$	cleaning threshold; detection threshold (noise units)
ι_t	innovation let through by the cleaning operator
M, ρ_i	tracker mask; eigenvalues of $U_M^\top U_M$
$T_p, T_{\text{rel}}, D_{\text{raw}}$	protection horizon; release delay; raw-learning time
α, β	verifier miss rate on objects; confirm rate on errors

4. Absorption

4.1. Each pixel is a weighted regression on the model's scores

Lemma 1 (Per-pixel regression). *Let $X_t = USV^\top + E$ be the rank- k truncation, so that $U^\top E = 0$ and $EV = 0$, and $z_j = U^\top c_{t-j}$. Then for every pixel p*

$$U_{p,:} = \left(\sum_{j \geq 0} f^j c_{t-j,p} z_j^\top \right) \left(\sum_{j \geq 0} f^j z_j z_j^\top \right)^{-1}. \quad (2)$$

Proof. Since $U^\top E = 0$, $U^\top X_t = SV^\top =: A$, whose j -th column is $\sqrt{f^j} z_j$. Row p of X_t is $U_{p,:}A + E_{p,:}$, and $EV = 0$ gives $X_{p,:}V = U_{p,:}S$, i.e. $U_{p,:} = X_{p,:}VS^{-1} = X_{p,:}A^\top(AA^\top)^{-1}$ because $AA^\top = S^2$. Writing the products as sums over j gives (2). $\square$

The model's value at pixel p for a frame with scores v is $U_{p,:}v$: the exponentially weighted least-squares prediction of that pixel from the scores. The same structure underlies PAST and PETRELS [19, 20, 31]; we use it to describe how a stopped target enters the model.

Lemma 2 (Leverage recursion). *Assume the scores do not depend on the values of the pixels of a small support Ω (they are estimated from the other pixels). Then for $p \in \Omega$ the loading $w_{p,t} = U_{p,:}$, expressed in a fixed basis of the counted subspace, follows exponentially weighted recursive least squares,*

$$\begin{aligned} R_t &= fR_{t-1} + z_t z_t^\top, \quad w_{p,t} = w_{p,t-1} + R_t^{-1} z_t \iota_{t,p}, \\ \iota_{t,p} &= c_{t,p} - w_{p,t-1}^\top z_t, \end{aligned} \tag{3}$$

and the fit of the current frame moves by $h_t \iota_{t,p}$ with the leverage

$$h_t = z_t^\top R_t^{-1} z_t = \frac{\eta_t}{f + \eta_t} \in [0, 1), \quad \eta_t = \sum_{k \leq r} \frac{v_k^2}{s_k^2}, \tag{4}$$

the second equality holding in the model's own basis, in which $R_{t-1} = \text{diag}(s_1^2, \dots, s_r^2)$. For stationary scores, $\mathbb{E}h_t \approx r(1 - f)$.

Proof. With fixed regressors z_j , (2) is the solution of a weighted least-squares problem whose information matrix obeys $R_t = fR_{t-1} + z_t z_t^\top$; the Sherman–Morrison formula gives the recursion and $z_t^\top (w_{p,t} - w_{p,t-1}) = h_t \iota_{t,p}$. In the eigenbasis of the weighted covariance restricted to the counted subspace $R_{t-1} = S_r^2$, and $z_t^\top (fS_r^2 + z_t z_t^\top)^{-1} z_t = \eta_t / (f + \eta_t)$ with $\eta_t = z_t^\top S_r^{-2} z_t$, which equals $\sum_k v_k^2 / s_k^2$ at $z_t = v$. For stationary scores $R_t \approx \mathbb{E}[zz^\top] / (1 - f)$, so $\mathbb{E}h_t \approx (1 - f) \text{tr}(I_r) = r(1 - f)$. $\square$

4.2. The absorption law

A target of contrast c (in units of the noise scale) stops on a support Ω at time t_0 . Let δ_t be its share in the fit and ι_t the innovation that the cleaning

operator lets through on Ω : $\iota_t = \kappa$ while $c - \delta_t > \kappa$ for soft thresholding at $\kappa\hat{\sigma}$ (and $c - \delta_t$ afterwards); $\iota_t = 0$ for imputed or masked pixels; $\iota_t = 0$ during the release delay and $c - \delta_t$ after it for impute-and-release.

Proposition 1 (Two absorption mechanisms). *A stopped target is absorbed at $D^* \approx \min(D_{\text{lev}}, D_{\text{rank}})$, the earlier of the times given by the following two mechanisms.*

(a) Leverage ramp. *By Lemma 2, the share δ follows the recursive least-squares recursion driven by ι ; with frozen scores, $\delta_{t+1} = \delta_t + h_t \iota_t$. Absorption by this mechanism happens at $D_{\text{lev}} = \min\{D : c - \delta_{t_0+D} < \kappa_{\text{det}}\}$. With constant leverage h ,*

$$\begin{aligned} D_{\text{lev}} &\approx \frac{c - \kappa_{\text{det}}}{\kappa h} && \text{for soft cleaning,} \\ D_{\text{lev}} &\approx \frac{\ln(c/\kappa_{\text{det}})}{-\ln(1-h)} && \text{for raw learning,} \\ D_{\text{lev}} &= \infty && \text{for imputed or masked pixels.} \end{aligned}$$

D_{lev} does not depend on $|\Omega|$.

(b) Rank crossing. *The part of the leak not yet absorbed by the ramp accumulates in an uncounted direction. When*

$$\sum_{k < D} f^{D-1-k} |\Omega| \iota_{t_0+k}^2 + d \geq \tau / \hat{\sigma}^2, \quad (5)$$

a component carrying the target's shape is counted and the next projection explains the target; D_{rank} is the first such D . It decreases with $|\Omega|$.

Derivation. (a) is Lemma 2 applied to each pixel of Ω with the operator's innovation; the closed forms follow by summing a constant increment (soft) or iterating $c - \delta \mapsto (1 - h)(c - \delta)$ (raw). For (b), the leak's unexplained energy in the weighted window is the first term of (5) (in units of $\hat{\sigma}^2$). It joins an uncounted direction that already carries noise: in the truncated incremental SVD this is the residual of the newest column, of energy $\approx d\hat{\sigma}^2$, which we verified on the estimator (the leading uncounted singular value sits at $\approx \hat{\sigma}\sqrt{d}$ before onset). The direction is counted once the total exceeds

| τ . This step is a calibrated approximation rather than a theorem. $\square$

Corollary 1. *(i) Soft-threshold trackers absorb every stopped target, small ones included, in a time linear in its contrast and inversely proportional to the leverage $h \approx r(1 - f)$: faster forgetting and a background with more counted components absorb faster. (ii) Retention is a design parameter only for operators with zero innovation (imputation, masks); every leaking operator has a finite, predictable retention. (iii) Without the noise term d in (5) and without mechanism (a), the law reduces to the accumulated-leak rule $\sum_k f^k |\Omega| \iota^2 \geq \tau / \hat{\sigma}^2$, which predicts that small targets are never absorbed under soft cleaning.*

4.3. Verification on the implemented estimator

We generated synthetic scenes (120×160 grid, a smooth mean image plus two slowly varying patterns, i.i.d. Gaussian noise) with one square target of constant contrast that stops at t_0 , and ran the implemented estimator (float32, truncated incremental SVD with $k = 6$, the rank rule above, robust projection) under soft cleaning and under imputation with a 30-frame release. For each run, a clean pass with the same noise realisation supplied the scores, leverages and onset state from which Proposition 1 was evaluated. Over 4,320 runs (10 seeds; $f \in \{0.90, 0.95, 0.97, 0.99\}$; sides 2 to 32 cells; contrast 5, 10, 20; three levels of background activity), Table 2 and Fig. 2 report the 2,880 runs with contrast at least 10σ (contrast 5σ lies close to the detection threshold $\kappa_{\text{det}} = 3.89$). The median error is 2 frames, the median relative error 0.130 (95 % bootstrap CI [0.114, 0.143]); the crossing time is predicted with median error 0 frames (IQR $[-1, 0]$; 1,357 runs). Where no crossing occurred (1,354 runs), the leverage law over-predicts slightly: the median ratio of predicted to measured time is 1.087 for soft cleaning and 1.053 with release. The accumulated-leak rule declares 1,200 of the runs never absorbed and has a median relative error of 0.370 on the others. Replacing the truncated incremental SVD by an exact SVD of the weighted window changes the absorption time by 0.2 frames on average (maximum 4; 108 paired runs), so the analysis of the exact weighted subspace carries over to the implementation.

Fig. 3 shows one run of each mechanism. A small target ($|\Omega| = 16$) leaves the counted rank unchanged and is absorbed by the ramp: its residual contrast falls below κ_{det} at frame 35, and the ramp predicted from the clean background crosses at frame 47, the over-prediction noted above. A large

Table 2: Synthetic: accuracy of Proposition 1 on the implemented estimator (contrast $\geq 10\sigma$; errors in frames; relative errors after the start of learning).

Subset	Runs	Median $ e $	Within 3	Median rel. err.
All (contrast $\geq 10\sigma$)	2,880	2	68 %	0.130
Soft cleaning	1,440	3	55 %	0.150
Impute + release	1,440	1	82 %	0.083
$f = 0.9$	720	1	72 %	0.102
$f = 0.95$	720	2	63 %	0.154
$f = 0.97$	720	1	69 %	0.118
$f = 0.99$	720	2	68 %	0.142
Activity $0.5\times$	960	2	67 %	0.112
Activity $1\times$	960	2	69 %	0.125
Activity $2\times$	960	2	68 %	0.150

target ($|\Omega| = 1024$) is counted as a new component at the predicted crossing (frame 5) and absorbed at once.

5. Accuracy under tracker masks

A tracker supplies a mask M of pixels that must not be learned (Section 6). Let U be orthonormal (the counted basis), $\rho_1, \dots, \rho_r$ the eigenvalues of $U_M^\top U_M$, $\rho = \max_i \rho_i < 1$, and let the frame be $x = Ua + \varphi + n$ with noise of variance σ^2 per entry and a target φ supported on Ω .

Proposition 2 (Masked coefficients). *The masked least-squares coefficients $\hat{a} = \arg \min_b \|x_{M^c} - U_{M^c} b\|$ are $\hat{a} = G^{-1} U_{M^c}^\top x_{M^c}$ with $G = I - U_M^\top U_M$, and*

1. $\hat{a} - a = G^{-1} U_{M^c}^\top (n_{M^c} + \varphi_{\Omega \setminus M})$;
2. $\mathbb{E} \|\hat{a} - a\|^2 = \sigma^2 \sum_i (1 - \rho_i)^{-1} + \|G^{-1} U_{M^c}^\top \varphi_{\Omega \setminus M}\|^2$;
3. $\|\hat{a} - a\| \leq (\|U_{M^c}^\top n_{M^c}\| + \|U_{M^c}^\top \varphi_{\Omega \setminus M}\|) / (1 - \rho)$;
4. $\|U_M(\hat{a} - a)\| \leq \sqrt{\rho} \|\hat{a} - a\|$;
5. G costs $O(|M|r^2)$ and the rest $O(dr)$.

Proof. $U^\top U = U_M^\top U_M + U_{M^c}^\top U_{M^c} = I$ gives G , whose eigenvalues are $1 - \rho_i > 0$. Substituting x_{M^c} gives (1). The noise term has covariance

$$\sigma^2 G^{-1} U_{M^c}^\top U_{M^c} G^{-1} = \sigma^2 G^{-1},$$

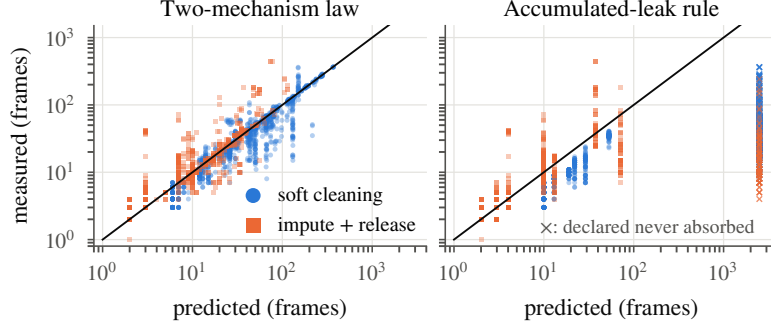


Figure 2: Synthetic: predicted vs. measured absorption time (frames after the start of learning, plus one; log scale). Left: Proposition 1; right: the accumulated-leak rule, crosses marking runs it declares never absorbed. Blue: soft cleaning; orange: imputation with release.

Table 3: Idealised check of Proposition 2 (squared coefficient error in units of σ^2 ; mean $\pm$ standard error over 4000 trials).

Mask	$\rho_{\max}$	Predicted	Measured (MC)	Bound/meas.
6×8	0.04	4.04	4.10 ± 0.05	1.03
20×25	0.33	4.55	4.59 ± 0.05	1.37
40×50	0.83	10.78	10.77 ± 0.15	2.76
55×75	0.98	74.63	74.64 ± 1.15	4.72
20×25 , leak 10%	0.33	1002.80	1004.29 ± 1.20	1.11
20×25 , leak 30%	0.33	8381.43	8382.44 ± 3.47	1.09

of trace $\sigma^2 \sum_i (1 - \rho_i)^{-1}$, and the cross term with the deterministic leakage has zero mean, which gives (2). (3) follows from $\|G^{-1}\|_2 = (1 - \rho)^{-1}$ and (4) from $\|U_M\|_2 = \sqrt{\rho}$. $\square$

Masking costs what the masked rows carry of the background, not how many rows there are; leakage of the target outside the mask costs far more. Table 3 confirms item 2 by Monte Carlo on a smooth image basis (60×80 , $r = 4$, 4000 trials per case): measured and predicted errors agree within Monte Carlo error, the bound of item 3 held in all 24,000 trials, and its looseness grows from 1.03 to 4.72 as the mask covers more of the dominant structure; a band of leaked target pixels 10 % of the mask’s height raises the error by a factor of about 219.

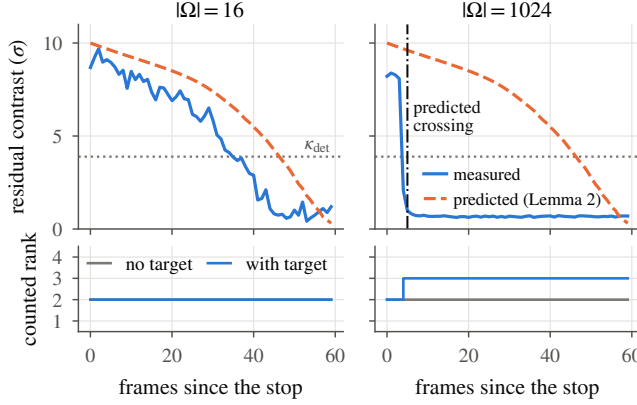


Figure 3: The two mechanisms on the implemented estimator (synthetic, $f = 0.97$, contrast 10σ , soft cleaning, one run). Top: residual contrast of the stopped target and the ramp of Lemma 2 predicted from the clean background; bottom: counted rank with and without the target.

Proposition 3 (Masked rows). *Under the assumption of Lemma 2, let the pixels of M be imputed by the current fit. Then (1) their innovation is zero and they are never absorbed, for any stop duration; (2) their loadings are frozen: changes of the scores (global illumination, coefficient drift) are still followed, whereas a change of the pixels’ own loadings accumulates as a bias that grows linearly in the duration for linear drift; (3) after release, the accumulated bias is unlearned with the raw-learning dynamics of Proposition 1; (4) target pixels outside M evolve by Proposition 1 with their own innovation.*

Proof. Substitute $c_{t,p} = w_{p,t-1}^\top z_t$ in Lemma 2: the update term vanishes, which gives (1) and (2). After release the innovation is the residual, as in the raw case of Proposition 1, which gives (3); (4) holds because the recursion is per pixel. $\square$

6. The closed loop

6.1. Protection and its failure mode

The tracker protects the boxes of its confirmed tracks (enlarged by 20%). Protecting a real stopped object prevents absorption (Proposition 3); protecting a *model error*—a ghost of an object that left, an illumination change, flickering clutter—keeps the error alive, because the model is prevented from

learning the truth. An unverified track is protected until an expiry statistic reaches the horizon T_p ; release then takes T_{rel} frames and learning D_{raw} .

Proposition 4 (Irreducible ambiguity). *Let two scenario classes coincide in law on the frames up to time t : an object stopped at t_0 that will leave after t , and one that will stay. Every causal protection policy takes the same decision at t (in law) in both. If, given the elapsed stop time $s = t - t_0$, the frames carry no information on the remaining stop time, every causal protection policy is equivalent to a (possibly randomised) stopping rule on s .*

Proof. A causal decision is measurable with respect to the frames up to t , whose laws coincide in the two classes; under the conditional independence, the decision can depend on the frames only through s . $\square$

Proposition 5 (Persistence of a protected error). *Let a model-error region be protected from time 0 and never travel.*

1. *If the expiry statistic grows at unit rate while the region does not travel (non-resettable), the error persists at most $T_p + T_{\text{rel}} + D_{\text{raw}}$.*
2. *If the statistic restarts at the events of a Poisson process of rate λ , the expected protection time is*

$$m(T_p) = \frac{e^{\lambda T_p} - 1}{\lambda}. \quad (6)$$

3. *If a verifier looks before each expiry, confirms an error with probability β , and a confirmation renews protection for T_p , the expected persistence is at most $T_p/(1 - \beta) + T_{\text{rel}} + D_{\text{raw}}$, and the error heals almost surely if $\beta < 1$. A real object stopped for D is retained for $\min(D, T_p)$ without verification and, if verified before each expiry, for D with probability $(1 - \alpha)^{\lceil D/T_p \rceil - 1}$.*

Proof. (1) is immediate. (2) Let m be the expected waiting time for the first gap of length T_p between events. Conditioning on the first event time $X \sim \text{Exp}(\lambda)$, $m = T_p P(X \geq T_p) + \mathbb{E}[(X + m)\mathbf{1}\{X < T_p\}]$; solving gives $me^{-\lambda T_p} = \lambda^{-1}(1 - e^{-\lambda T_p})$, i.e. $m = (e^{\lambda T_p} - 1)/\lambda$. (3) The number of protection periods is geometric with success probability $1 - \beta$, hence finite almost surely if

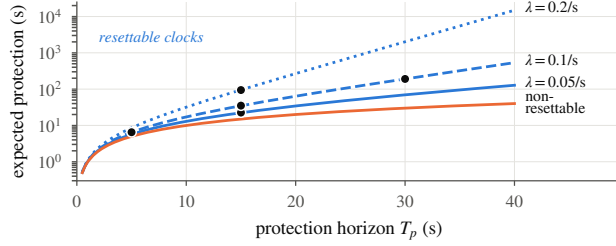


Figure 4: Expected protection of a model error vs. the horizon: resettable clocks (Proposition 5, item 2) against a non-resettable one (item 1); dots: simulation at 30 frames per second.

| $\beta < 1$; the retention statement counts the looks that must all succeed. $\square$

A stillness clock driven by the tracker’s speed estimate is resettable: the centroid of a ghost jitters, and every jitter that exceeds the speed threshold restarts it. On CDnet *parking*, with such a clock, a track on a region that the ground truth marks as static (0.29 to 0.67 of its box) stayed protected for at least 66 s against a horizon of 15 s, its stillness never exceeding 12.0 s. Item 2 explains this: the clock kept restarting before it reached the horizon, and already for Poisson restarts at one per 5 s the expected protection is $(e^3 - 1) \cdot 5 \approx 95$ s. Measuring stillness by displacement from an anchor (the centre must leave a radius of one body size to restart the clock) makes the clock non-resettable by jitter and moves the loop into item 1. Fig. 4 compares the two regimes with simulation (94.3 s measured vs. 95.4 s predicted at $\lambda = 0.2/\text{s}$, $T_p = 15$ s).

6.2. The optimal horizon

Proposition 6 (Quantile horizon). *With n_o object stops of duration $D \sim F_D$ and n_e ghosts per unit time, cost c_a per unit time a stopped object is not protected and c_e per unit time a ghost is kept by protection, a time-based policy with horizon T and a non-resettable clock costs $J(T) = c_a n_o \mathbb{E}(D - T)_+ + c_e n_e T$. J is convex and minimised at*

$$T^* = F_D^{-1}(1 - q), \quad q = \frac{c_e n_e}{c_a n_o} \quad (T^* = 0 \text{ if } q \geq 1). \quad (7)$$

For errors whose source lasts $E \sim F_E$, the optimum solves $P(D > T)/P(E > T) = q$.

Proof. $\frac{d}{dT} \mathbb{E}(D - T)_+ = -P(D > T)$ and $\frac{d}{dT} \mathbb{E} \min(E, T) = P(E > T)$; for ghosts $E = \infty$ and $J''(T) = c_a n_o f_D(T) \geq 0$. $\square$

In simulation with log-normal and bimodal (traffic light and parking) stop laws, the empirical minimiser of J equals the quantile in 6 of 6 cases to the resolution of the search grid (0.5 s).

6.3. Verification

A semantic verifier changes the problem. We use a pretrained detector (YOLOv10s [51] trained on COCO [52], run from its ONNX export on one CPU core) under the system’s token-bucket budget: each frame grants half of the frame period, and looks are chosen greedily by priority per millisecond, a first look at a new track (“verify”) having the highest priority, ahead of periodic refreshes and full-frame sweeps. A track counts as verified once a detection has matched it, and as rejected once a look has found nothing while none has matched it (after two such looks it is dropped as clutter). Verified tracks stay protected while the track lives, rejected ones lose protection at once, and only tracks not yet looked at are subject to T_p . In our CDnet runs every confirmed track was looked at long before the horizon (over the six IOM videos, the longest still time of a confirmed track not yet looked at was 3.6 s), so the horizon never bound: policies that differed only in T_p produced identical masks. With a verifier, the operative bound is item 3 of Proposition 5; the quantile horizon is the rule for operation without a verifier or without the budget for one.

Algorithm 1 One frame of the tracker-guided background model.

- 1: $M \leftarrow$ boxes of confirmed tracks that are verified, or unverified with stillness below T_p
 - 2: fit scores v on the pixels outside M (robust masked least squares)
 - 3: residual $e \leftarrow x - U_r v$; detect $|e| > \kappa_{\text{det}} \hat{s}$ with per-pixel scale $\hat{s}$; majority filter
 - 4: screen detections for ghosts, illumination changes and scene changes
 - 5: schedule verifier looks within the token-bucket budget; update verdicts
 - 6: update the Kalman tracker (two-stage association, track life cycle, anchor stillness)
 - 7: cleaned column: impute outliers (with a one-cell halo) and M by the fit; learn stale unprotected outliers raw
 - 8: incremental SVD update with forgetting f ; re-orthonormalise every 64 updates
-

6.4. The per-frame algorithm

Algorithm 1 summarises one frame of the system; all steps except the verifier cost $O(dr^2)$.

7. Experiments on CDnet 2014

7.1. Protocol

Data and evaluator. CDnet 2014 [21] has 53 videos in 11 categories; we use the official temporal regions of interest and label semantics (static and shadow pixels are negatives, outside-ROI and unknown pixels are ignored), and the official averaging (per-video metrics, mean over the videos of a category, mean over categories). Our evaluator reproduces the confusion matrices of the official MATLAB code exactly on 10 of 10 videos checked. Masks are computed on the work grid and mapped to sensor pixels by nearest neighbour; the frame rate is taken as 30 frames per second.

Splits. The development split consists of the five 2012 categories other than intermittent object motion (IOM): baseline, dynamic background, camera jitter, shadow and thermal (25 videos). IOM (6 videos) is the test category for the claims about stopped objects. The 22 videos of the five categories added in 2014 were held out and evaluated once, after the method was frozen.

Front end. The segmentation front end was selected on the development split only, among 36 configurations (work-grid budget, per-pixel false-alarm rate, majority filter, stabilisation), by a rule fixed before the runs (best development F-measure, ties within 0.005 broken by speed). The selected configuration raises the development mean F-measure from 0.526 to 0.614; averaged over the grid, the 5×5 majority filter adds +0.091, full resolution +0.015, and stabilisation -0.001.

Policies. Five protection policies were compared (Fig. 5) under a protocol fixed before the runs (Table 5): a policy is admissible if it is not worse than V0 on the development split (one-sided paired Wilcoxon signed-rank test over the 25 videos, $\alpha = 0.05$), and the IOM result is leave-one-video-out: each IOM video is scored with the admissible policy that is best on the other five. All policies were admissible, and every fold chose V2, which we therefore froze.

Baselines. Four online baselines ran in MATLAB on all 53 videos, streamed on the same frames and luminance as our method but box-filtered to at most 20 000 cells (our front end uses up to 80 000), each with its own foreground rule. GRASTA [6] (the LRSLibrary copy of the authors’ implementation, all pixels observed) and NORST [8] (the authors’ code) used the parameters of the authors’ video demonstrations. OR-PCA [53] (the LRSLibrary copy of the authors’ core) has a default sparsity weight $1/\sqrt{d}$ below the noise level, so that its sparse part absorbs whole frames. For incremental PCP [10] no code was obtainable; we re-implemented its Algorithm 2 with the published defaults (rank 1, a 150-frame window, two inner loops) and its foreground rule, a per-frame unimodal threshold [54] of $|S|$, omitting only the abrupt-change re-initialisation, whose threshold is not published. Its rank-1 SVD operations are exact to machine precision on matrices of exact rank, and on grayscale *highway* at full resolution it reaches $F = 0.763$, against the published 0.7455. The sparsity weights of OR-PCA and incremental PCP, which neither the defaults nor the paper fix, were selected on the development split under a rule fixed before the runs (Table 4): a grid of four values, extended by doubling because both methods first selected its upper edge. The official results of other methods are quoted from the benchmark’s site (accessed on 23 October 2026).

Table 4: Sparsity weight λ of the two tuned baselines (frames in $[0, 1]$): development mean F-measure, selected value in bold, “–”: not run by the stopping rule; † F-measure of the re-implemented incremental PCP on grayscale *highway* at full resolution (published: 0.7455).

λ	0.01	0.02	0.04	0.08	0.16	0.32
OR-PCA, development	0.206	0.402	0.546	0.594	0.517	–
incPCP, development	0.260	0.313	0.413	0.509	0.521	0.326
incPCP, <i>highway</i> †	0.460	0.419	0.541	0.760	0.763	0.511

Table 5: Protection policies on the frozen front end: development mean F-measure, mean paired gain over V0 on the 25 development videos and the one-sided p -value for being worse than V0, and IOM F-measure.

Id	Policy	Dev. F	Paired gain	p_{worse}	IOM F
V0	no verifier; motion rule; $T_p = 15$ s	0.614	–	–	0.445
V4	no verifier; relaxed; $T_p = \infty$	0.604	–0.010	0.50	0.577
V1	verifier; motion rule; $T_p = 15$ s	0.655	+0.043	0.97	0.588
V2	verifier; no motion rule; $T_p = 15$ s	0.651	+0.039	0.99	0.614
V3	verifier; no motion rule; $T_p = \infty$	0.651	+0.039	0.99	0.614

7.2. Results

Intermittent object motion. Table 6 reports the IOM videos and Fig. 6 compares every method. The verified loop raises the mean F-measure from 0.445 (V0) and 0.417 (no feedback) to 0.614 (leave-one-video-out), closing 52 % of the gap to the ground-truth masks (0.771). It exceeds the four online subspace baselines (NORST 0.458, OR-PCA 0.439, incremental PCP 0.331, GRASTA 0.241) and, among official results (Table 7), pROST (0.413) and the mixture models (0.521, 0.532), while SuBSENSE (0.657) and PAWCS (0.776) remain ahead. The verifier is a pretrained network; comparisons should account for that.

Retention and ghosts. Fig. 7 shows the effect on single frames, and Fig. 8 isolates absorption: recall of ground-truth foreground pixels as a function of the time since they became foreground. Without feedback, recall on pixels that have been foreground for 3–10 s is 0.09 and for 10–30 s 0.04; the verified loop keeps 0.71 and 0.66. The price is more ghosts: on static pixels vacated by a target 10–30 s earlier, the false-positive rate rises from 0.007 without feedback to 0.013 with the verified loop.

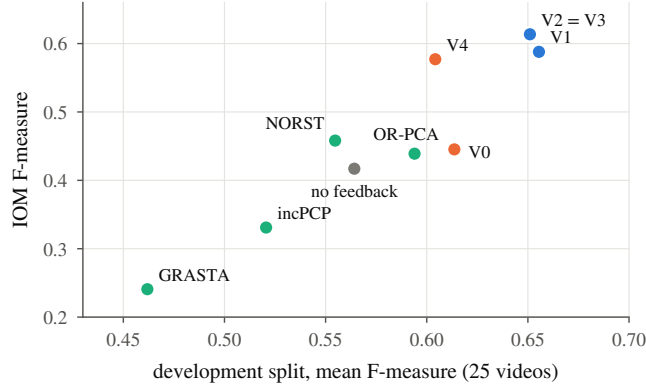


Figure 5: Every protection policy and baseline on both splits: development mean F-measure (selection) against IOM F-measure (test). Blue: policies with the verifier; orange: without; green: online subspace baselines; grey: no feedback.

Table 6: CDnet IOM, F-measure per video. Off: no feedback; V0: tracker feedback without verifier; V2*: verified loop, leave-one-video-out; Orc.[†]: boxes of ground-truth objects as protection masks, with the same front end (diagnostic reference, not a method).

Video	Off	V0	V2*	Orc. [†]	GRASTA	NORST	OR-PCA	incPCP
abandonedBox	0.432	0.493	0.479	0.908	0.319	0.540	0.501	0.384
parking	0.478	0.496	0.691	0.828	0.045	0.315	0.478	0.227
sofa	0.491	0.553	0.761	0.694	0.348	0.665	0.561	0.417
streetLight	0.289	0.527	0.956	0.954	0.124	0.892	0.424	0.211
tramstop	0.389	0.264	0.326	0.610	0.285	0.189	0.275	0.334
winterDriveway	0.422	0.340	0.468	0.630	0.324	0.149	0.395	0.412
Mean	0.417	0.445	0.614	0.771	0.241	0.458	0.439	0.331

Development split. The verified loop also improves the development split (Table 8; 0.614 to 0.651, two-sided paired Wilcoxon $p = 0.030$), with the gains concentrated on the baseline and shadow categories. On single categories some baselines are ahead of it: NORST on baseline, camera jitter and thermal, OR-PCA on baseline, camera jitter and shadow; over the split both remain behind (0.555 and 0.594).

Held-out categories. On the 22 videos of the five categories added in 2014, evaluated once, the mean F-measure is 0.552 with the verified loop, 0.526 for V0 and 0.546 without feedback: there the verified loop is on a par with no feedback, whereas the unverified loop loses. Over all 11 categories the verified loop reaches 0.603 (V0: 0.558; no feedback: 0.542). On the held-out

Table 7: Official CDnet results of other methods on IOM (changedetection.net, accessed 23 October 2026).

Method (official results)	IOM F
GMM (Stauffer–Grimson)	0.521
GMM (Zivkovic)	0.532
pROST	0.413
SuBSENSE	0.657
PAWCS	0.776

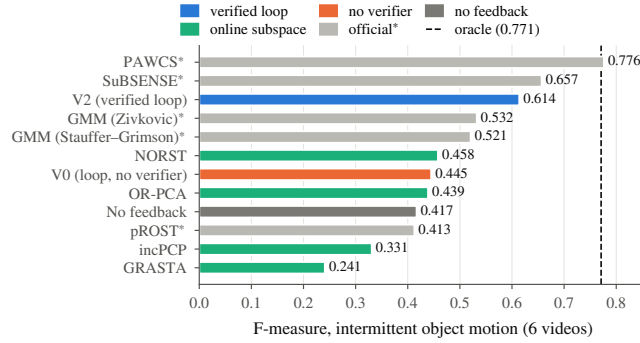


Figure 6: IOM F-measure of every method. *Official changedetection.net results (accessed 23 October 2026); the online subspace baselines ran on our frames and grid.

categories the baselines reach 0.413 (GRASTA), 0.284 (NORST), 0.417 (OR-PCA) and 0.435 (incPCP), and over all 11 categories 0.419 (GRASTA), 0.423 (NORST), 0.500 (OR-PCA) and 0.465 (incPCP). *Cost.* On one idle desktop core (one thread), on three IOM videos at their full 320×240 resolution, the pipeline takes 8.3 ms per frame on average (95th percentile 10.0 ms) without the verifier and 27.4 ms (231.9 ms) with it; on the same videos at 160×120 , the baselines take 3.0 ms (GRASTA), 79 ms (NORST), 2.6 ms (OR-PCA) and 0.3 ms (incPCP) in MATLAB. Table 9 gives every category and Fig. 9 compares the verified loop with no feedback and with the best online baseline in each.

Where the loop fails. Two videos fail, for opposite reasons (probe P09 on the frozen method). In *abandonedBox* the stopped object is a box, which has no class in the detector’s vocabulary: only 19 % of the pixels of objects stopped for 3 s or more lie in protection boxes, against at least 56 % in the other five

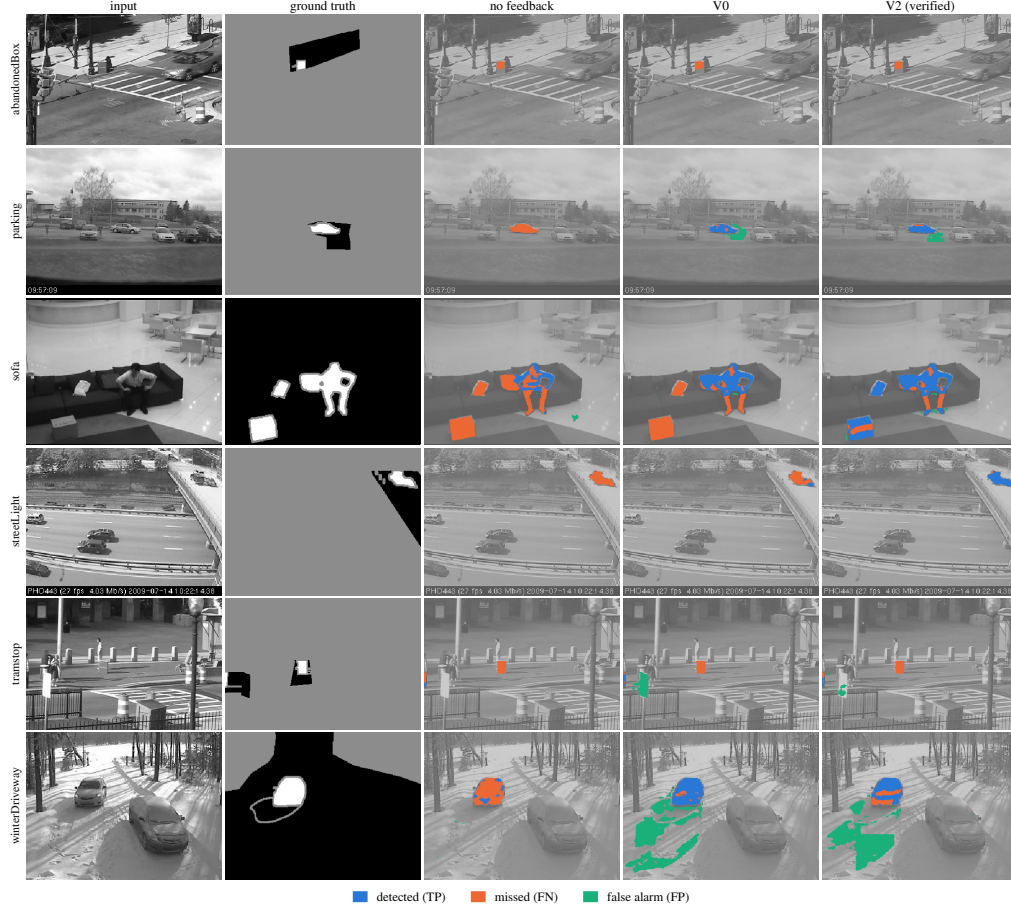


Figure 7: Error maps on the six IOM videos, each at the frame with the most ground-truth pixels stopped for at least 3s, chosen without looking at any output (frames 2787, 1661, 1344, 2153, 1409, 1889). Blue: detected; orange: missed; green: false alarm; grey in the ground truth: outside the region of interest or unknown.

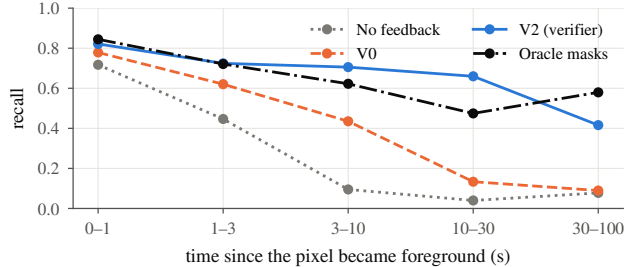


Figure 8: CDnet IOM (real data): recall of ground-truth foreground pixels by the time since they became foreground, pooled over the 6 videos.

Table 8: Development split (frozen front end), mean F-measure per category.

Category	Off	V0	V1	V2	GRASTA	NORST	OR-PCA	incPCP
Baseline	0.611	0.687	0.782	0.748	0.610	0.759	0.765	0.681
Dyn. bg.	0.550	0.573	0.570	0.564	0.374	0.122	0.230	0.399
Jitter	0.616	0.640	0.641	0.656	0.489	0.694	0.732	0.515
Shadow	0.559	0.559	0.692	0.681	0.530	0.582	0.683	0.617
Thermal	0.485	0.610	0.593	0.606	0.306	0.617	0.560	0.391
Mean	0.564	0.614	0.655	0.651	0.462	0.555	0.594	0.521

videos, and the verified loop scores slightly below V0 there (Table 6). This is consistent with the miss rate α of Proposition 5: under the implemented rule, an object that the verifier cannot confirm loses its protection at the first empty look. In *tramstop* the loop errs the other way: feedback multiplies false positives by 10 relative to no feedback, and 88 % of them lie inside protection boxes, where the model does not learn. Protection keeps these errors alive, as in the lock-in of Section 6; we did not measure how many of these regions the verifier confirmed.

7.3. Diagnostics of the loop

Before the front end was frozen, we decomposed the gap between the unverified loop and the oracle (Table 10, diagnostic). Coverage is the share of ground-truth pixels of objects stopped for at least 3s that lie inside the protection mask; the lock-in share is the share of false positives that lie inside wrongly protected static regions. The motion requirement of the unverified loop (“protect only tracks that travelled”) removes most of the coverage, because tracks of cars that drove in fragment and are reborn while stopped;

Table 9: All 11 CDnet 2014 categories, mean F-measure (frozen method; the five categories added in 2014 evaluated once, after freezing). No fb.: no feedback.

Category	V2	V0	No fb.	GRASTA	NORST	OR-PCA	incPCP
Baseline	0.748	0.687	0.611	0.610	0.759	0.765	0.681
Dyn. bg.	0.564	0.573	0.550	0.374	0.122	0.230	0.399
Jitter	0.656	0.640	0.616	0.489	0.694	0.732	0.515
Shadow	0.681	0.559	0.559	0.530	0.582	0.683	0.617
Thermal	0.606	0.610	0.485	0.306	0.617	0.560	0.391
IOM	0.614	0.445	0.417	0.241	0.458	0.439	0.331
Bad weather	0.746	0.680	0.687	0.393	0.381	0.660	0.566
Low fr.	0.613	0.611	0.606	0.399	0.506	0.499	0.432
Night	0.381	0.368	0.394	0.361	0.284	0.445	0.377
PTZ	0.242	0.210	0.270	0.211	0.125	0.218	0.156
Turbul.	0.777	0.759	0.770	0.699	0.125	0.265	0.646
Overall (11 cat.)	0.603	0.558	0.542	0.419	0.423	0.500	0.465

Table 10: Diagnostic decomposition on IOM with the default front end, before it was frozen (Off, P0 = current loop, relaxed variants, oracle).

Policy	IOM F	Baseline F	Coverage	Lock-in share
Off	0.294	0.582	—	—
P0	0.352	0.669	0.23	0.75
No expiry	0.305	0.689	0.24	0.82
No motion rule	0.468	0.721	0.66	0.96
Both relaxed	0.546	0.756	0.90	0.99
Oracle [†]	0.710	0.832	1.00	0.65

relaxing the rules restores coverage but turns almost all errors into protection-caused errors. The remaining gap is discrimination between stopped objects and model errors, which is what the verifier supplies.

8. Discussion and limitations

Proposition 1(b) is a calibrated approximation: its noise term was verified on the estimator, not derived; a random-matrix treatment of exponentially weighted windows would make it a theorem. The synthetic generator has a mean image and smooth drifting patterns; scenes whose background is not well described by a few components may absorb through mechanisms we did

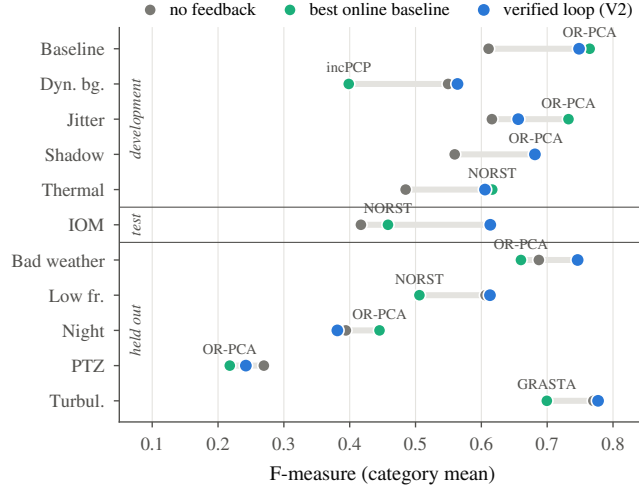


Figure 9: Per category: the verified loop (blue), no feedback (grey) and the best of the four online baselines (green, named).

not model, such as the flicker of a pixel’s outlier status that lets it be learned before release, which we observed on real video. The real-data front end trails the best per-pixel methods; the verifier is a pretrained network, and objects outside its vocabulary are absorbed; the frame rate of CDnet videos was taken as 30 per second for the seconds-based horizons. Incremental PCP is our re-implementation, validated against one published result, and two baselines needed their sparsity weight tuned on the development split, which GRASTA and NORST did not. Timings were measured on a desktop core, not on an embedded board, and the synchronous verifier stalls the frames on which it runs (95th percentile 231.9 ms); running it asynchronously would avoid the stall but was not evaluated.

9. Conclusion

A forgetting subspace background model learns each pixel by recursive least squares on its own scores. Absorption of a stopped target is therefore a leverage ramp, independent of the target’s size, unless a rank-crossing event comes first, and its time can be predicted from the background alone. Tracker feedback removes absorption but must be designed as a closed loop: an expiry clock that the protected error can reset makes the error’s persistence exponential in the horizon, the optimal horizon without verification is

Table A.11: Scripts behind the results (paths relative to the folder `experiments/` of the repository).

Result	Script
Absorption law (Table 2, Fig. 2)	<code>e01_absorption_law.py</code> <code>e01_analyse.py</code>
Exact vs. truncated SVD	<code>probes/p02_idealised.py</code>
Masked coefficients (Table 3)	<code>probes/p06_masked_precheck.py</code>
Persistence, horizon (Fig. 4)	<code>probes/p07_loop_precheck.py</code>
Lock-in on <i>parking</i>	<code>probes/p03_lockin_diagnosis.py</code>
CDnet evaluator check	<code>validate_cdnet.py</code>
Front-end selection	<code>j2_frontend_grid.py</code>
Policies (Table 5)	<code>j4_policy_runs.py</code> <code>j4_policy_analyse.py</code>
Verifier latency	<code>probes/p08_verifier_latency.py</code>
Retention (Fig. 8)	<code>cdnet_retention.py</code>
Failure diagnostics	<code>probes/p09_frozen_loop_failures.py</code>
Loop diagnostics (Table 10)	<code>probes/p05_protection_coverage.py</code>
Baselines	<code>baselines/run_all.m</code>
Baseline tuning (Table 4)	<code>baselines/select_lambda.py</code>
incPCP validation	<code>baselines/incpcp_validate.py</code>
Held-out evaluation	<code>heldout_runs.py</code>
Cost	<code>timing_idle.py</code> <code>baselines/timing_idle.m</code>

a quantile of the stop durations, and a budgeted verifier replaces the horizon by a classification with known error rates. On CDnet, these choices raise the F-measure on intermittent object motion from 0.445 to 0.614 at a mean cost of 27.4 ms per frame on one CPU core.

Appendix A. Reproducibility

Every number in this article is produced by the script `paper/scripts/make_artifacts.py` from raw results listed in the experiment registry of the accompanying repository, which records for each experiment the command, the code version and the outputs. Table A.11 lists the scripts behind each part of the article. The frozen configuration is `configs/paper.toml`, with the verifier configuration `configs/paper_verifier.toml`.

Experiments used Python 3.13, NumPy 2.2, SciPy 1.17, ONNX Runtime 1.20 and MATLAB R2024a, one thread per process.

References

- [1] E. J. Candès, X. Li, Y. Ma, J. Wright, Robust principal component analysis?, *Journal of the ACM* 58 (3) (2011) 1–37.
doi:10.1145/1970392.1970395.
- [2] V. Chandrasekaran, S. Sanghavi, P. A. Parrilo, A. S. Willsky, Rank-sparsity incoherence for matrix decomposition, *SIAM Journal on Optimization* 21 (2) (2011) 572–596. doi:10.1137/090761793.
- [3] T. Bouwmans, E. H. Zahzah, Robust PCA via principal component pursuit: A review for a comparative evaluation in video surveillance, *Computer Vision and Image Understanding* 122 (2014) 22–34.
doi:10.1016/j.cviu.2013.11.009.
- [4] T. Bouwmans, A. Sobral, S. Javed, S. K. Jung, E.-H. Zahzah, Decomposition into low-rank plus additive matrices for background/foreground separation: A review for a comparative evaluation with a large-scale dataset, *Computer Science Review* 23 (2017) 1–71. arXiv:1511.01245,
doi:10.1016/j.cosrev.2016.11.001.
- [5] N. Vaswani, T. Bouwmans, S. Javed, P. Narayanamurthy, Robust subspace learning: Robust PCA, robust subspace tracking, and robust subspace recovery, *IEEE Signal Processing Magazine* 35 (4) (2018) 32–55. arXiv:1711.09492, doi:10.1109/MSP.2018.2826566.
- [6] J. He, L. Balzano, A. Szlam, Incremental gradient on the Grassmannian for online foreground and background separation in subsampled video, in: *2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012, pp. 1568–1575.
doi:10.1109/CVPR.2012.6247848.
- [7] C. Qiu, N. Vaswani, B. Lois, L. Hogben, Recursive robust PCA or recursive sparse recovery in large but structured noise, *IEEE Transactions on Information Theory* 60 (8) (2014) 5007–5039.
arXiv:1211.3754, doi:10.1109/TIT.2014.2331344.

- [8] P. Narayanamurthy, N. Vaswani, Nearly optimal robust subspace tracking, in: Proceedings of the 35th International Conference on Machine Learning (ICML), Vol. 80 of Proceedings of Machine Learning Research, 2018, pp. 3701–3709. [arXiv:1712.06061](#).
- [9] P. Narayanamurthy, N. Vaswani, Fast robust subspace tracking via PCA in sparse data-dependent noise, *IEEE Journal on Selected Areas in Information Theory* 1 (3) (2020) 723–744. [arXiv:2006.08030](#), [doi:10.1109/JSAIT.2020.3040365](#).
- [10] P. Rodríguez, B. Wohlberg, Incremental principal component pursuit for video background modeling, *Journal of Mathematical Imaging and Vision* 55 (1) (2016) 1–18. [doi:10.1007/s10851-015-0610-z](#).
- [11] F. Seidel, C. Hage, M. Kleinsteuber, pROST: a smoothed ℓ_p -norm robust online subspace tracking method for background subtraction in video, *Machine Vision and Applications* 25 (5) (2014) 1227–1240. [arXiv:1302.2073](#), [doi:10.1007/s00138-013-0555-4](#).
- [12] Z. Zivkovic, F. van der Heijden, Efficient adaptive density estimation per image pixel for the task of background subtraction, *Pattern Recognition Letters* 27 (7) (2006) 773–780. [doi:10.1016/j.patrec.2005.11.005](#).
- [13] M. Harville, A framework for high-level feedback to adaptive, per-pixel, mixture-of-Gaussian background models, in: A. Heyden, G. Sparr, M. Nielsen, P. Johansen (Eds.), *Computer Vision – ECCV 2002*, Vol. 2352 of Lecture Notes in Computer Science, Springer, 2002, pp. 543–560. [doi:10.1007/3-540-47977-5_36](#).
- [14] R. Cucchiara, C. Grana, M. Piccardi, A. Prati, Detecting moving objects, ghosts, and shadows in video streams, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 25 (10) (2003) 1337–1342. [doi:10.1109/TPAMI.2003.1233909](#).
- [15] L. Taycher, J. W. Fisher III, T. Darrell, Incorporating object tracking feedback into background maintenance framework, in: 2005 Seventh IEEE Workshops on Applications of Computer Vision (WACV/MOTION), 2005, pp. 120–125. [doi:10.1109/ACVMOT.2005.63](#).

- [16] F. Porikli, Y. Ivanov, T. Haga, Robust abandoned object detection using dual foregrounds, *EURASIP Journal on Advances in Signal Processing* 2008 (2008) 197875. doi:10.1155/2008/197875.
- [17] C. Chen, S. Li, H. Qin, A. Hao, Robust salient motion detection in non-stationary videos via novel integrated strategies of spatio-temporal coherency clues and low-rank analysis, *Pattern Recognition* 52 (2016) 410–432. doi:10.1016/j.patcog.2015.09.033.
- [18] M. Shakeri, H. Zhang, COROLA: A sequential solution to moving object detection using low-rank approximation, *Computer Vision and Image Understanding* 146 (2016) 27–39. doi:10.1016/j.cviu.2016.02.009.
- [19] B. Yang, Projection approximation subspace tracking, *IEEE Transactions on Signal Processing* 43 (1) (1995) 95–107. doi:10.1109/78.365290.
- [20] Y. Chi, Y. C. Eldar, R. Calderbank, PETRELS: Parallel subspace estimation and tracking by recursive least squares from partial observations, *IEEE Transactions on Signal Processing* 61 (23) (2013) 5947–5959. arXiv:1207.6353, doi:10.1109/TSP.2013.2282910.
- [21] Y. Wang, P.-M. Jodoin, F. Porikli, J. Konrad, Y. Benezeth, P. Ishwar, CDnet 2014: An expanded change detection benchmark dataset, in: *2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2014, pp. 393–400. doi:10.1109/CVPRW.2014.126.
- [22] Z. Zhou, X. Li, J. Wright, E. Candès, Y. Ma, Stable principal component pursuit, in: *2010 IEEE International Symposium on Information Theory (ISIT)*, 2010, pp. 1518–1522. doi:10.1109/ISIT.2010.5513535.
- [23] Z. Lin, M. Chen, Y. Ma, The augmented Lagrange multiplier method for exact recovery of corrupted low-rank matrices (2010). arXiv:1009.5055.
- [24] X. Zhou, C. Yang, W. Yu, Moving object detection by detecting contiguous outliers in the low-rank representation, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35 (3) (2013) 597–610. doi:10.1109/TPAMI.2012.132.

- [25] N. Vaswani, P. Narayanamurthy, Static and dynamic robust PCA and matrix completion: A review, *Proceedings of the IEEE* 106 (8) (2018) 1359–1379. [arXiv:1803.00651](#), [doi:10.1109/JPROC.2018.2844126](#).
- [26] L. Balzano, R. Nowak, B. Recht, Online identification and tracking of subspaces from highly incomplete information, in: *48th Annual Allerton Conference on Communication, Control, and Computing*, 2010, pp. 704–711. [arXiv:1006.4046](#), [doi:10.1109/ALLERTON.2010.5706976](#).
- [27] P. Narayanamurthy, N. Vaswani, Provable dynamic robust PCA or robust subspace tracking, *IEEE Transactions on Information Theory* 65 (3) (2019) 1547–1577. [arXiv:1705.08948](#), [doi:10.1109/TIT.2018.2872023](#).
- [28] P. Narayanamurthy, V. Daneshpajoo, N. Vaswani, Provable subspace tracking from missing data and matrix completion, *IEEE Transactions on Signal Processing* 67 (16) (2019) 4245–4260. [arXiv:1810.03051](#), [doi:10.1109/TSP.2019.2924595](#).
- [29] S. Javed, S. H. Oh, A. Sobral, T. Bouwmans, S. K. Jung, OR-PCA with MRF for robust foreground detection in highly dynamic backgrounds, in: *Computer Vision – ACCV 2014, Lecture Notes in Computer Science*, Springer, 2015, pp. 284–299. [doi:10.1007/978-3-319-16811-1_19](#).
- [30] J. Xu, V. K. Ithapu, L. Mukherjee, J. M. Rehg, V. Singh, GOSUS: Grassmannian online subspace updates with structured-sparsity, in: *2013 IEEE International Conference on Computer Vision (ICCV)*, 2013, pp. 3376–3383. [doi:10.1109/ICCV.2013.419](#).
- [31] L. Balzano, Y. Chi, Y. M. Lu, Streaming PCA and subspace tracking: The missing data case, *Proceedings of the IEEE* 106 (8) (2018) 1293–1310. [arXiv:1806.04609](#), [doi:10.1109/JPROC.2018.2847041](#).
- [32] K. Toyama, J. Krumm, B. Brumitt, B. Meyers, Wallflower: principles and practice of background maintenance, in: *Proceedings of the Seventh IEEE International Conference on Computer Vision (ICCV)*, Vol. 1, 1999, pp. 255–261. [doi:10.1109/ICCV.1999.791228](#).

- [33] C. Stauffer, W. E. L. Grimson, Adaptive background mixture models for real-time tracking, in: Proceedings 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), 1999, pp. 246–252. doi:10.1109/CVPR.1999.784637.
- [34] Z. Zivkovic, Improved adaptive Gaussian mixture model for background subtraction, in: Proceedings of the 17th International Conference on Pattern Recognition (ICPR), Vol. 2, 2004, pp. 28–31. doi:10.1109/ICPR.2004.1333992.
- [35] R. Heras Evangelio, T. Sikora, Static object detection based on a dual background model and a finite-state machine, EURASIP Journal on Image and Video Processing 2011 (2011) 1–11. doi:10.1155/2011/858502.
- [36] L. Maddalena, A. Petrosino, Stopped object detection by learning foreground model in videos, IEEE Transactions on Neural Networks and Learning Systems 24 (5) (2013) 723–735. doi:10.1109/TNNLS.2013.2242092.
- [37] Y. Tian, R. S. Feris, H. Liu, A. Hampapur, M.-T. Sun, Robust detection of abandoned and removed objects in complex surveillance videos, IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews) 41 (5) (2011) 565–576. doi:10.1109/TSMCC.2010.2065803.
- [38] K. Lin, S.-C. Chen, C.-S. Chen, D.-T. Lin, Y.-P. Hung, Abandoned object detection via temporal consistency modeling and back-tracing verification for visual surveillance, IEEE Transactions on Information Forensics and Security 10 (7) (2015) 1359–1370. doi:10.1109/TIFS.2015.2408263.
- [39] P.-L. St-Charles, G.-A. Bilodeau, R. Bergevin, SuBSENSE: A universal change detection method with local adaptive sensitivity, IEEE Transactions on Image Processing 24 (1) (2015) 359–373. doi:10.1109/TIP.2014.2378053.
- [40] P.-L. St-Charles, G.-A. Bilodeau, R. Bergevin, Universal background subtraction using word consensus models, IEEE Transactions on Image

- Processing 25 (10) (2016) 4768–4781.
doi:10.1109/TIP.2016.2598691.
- [41] C. Davis, W. M. Kahan, The rotation of eigenvectors by a perturbation. III, *SIAM Journal on Numerical Analysis* 7 (1) (1970) 1–46. doi:10.1137/0707001.
 - [42] P.-Å. Wedin, Perturbation bounds in connection with singular value decomposition, *BIT* 12 (1) (1972) 99–111. doi:10.1007/BF01932678.
 - [43] Y. Yu, T. Wang, R. J. Samworth, A useful variant of the Davis–Kahan theorem for statisticians, *Biometrika* 102 (2) (2015) 315–323. arXiv:1405.0680, doi:10.1093/biomet/asv008.
 - [44] V. A. Marčenko, L. A. Pastur, Distribution of eigenvalues for some sets of random matrices, *Mathematics of the USSR-Sbornik* 1 (4) (1967) 457–483. doi:10.1070/SM1967v001n04ABEH001994.
 - [45] J. Baik, G. Ben Arous, S. Péché, Phase transition of the largest eigenvalue for nonnull complex sample covariance matrices, *The Annals of Probability* 33 (5) (2005) 1643–1697. arXiv:math/0403022, doi:10.1214/009117905000000233.
 - [46] F. Benaych-Georges, R. R. Nadakuditi, The singular values and vectors of low rank perturbations of large rectangular random matrices, *Journal of Multivariate Analysis* 111 (2012) 120–135. doi:10.1016/j.jmva.2012.04.019.
 - [47] S. Pafka, M. Potters, I. Kondor, Exponential weighting and random-matrix-theory-based filtering of financial covariance matrices for portfolio optimization (2004). arXiv:cond-mat/0402573.
 - [48] M. Brand, Incremental singular value decomposition of uncertain data with missing values, in: *Computer Vision – ECCV 2002, Lecture Notes in Computer Science*, Springer, 2002, pp. 707–720. doi:10.1007/3-540-47969-4_47.
 - [49] M. Brand, Fast low-rank modifications of the thin singular value decomposition, *Linear Algebra and its Applications* 415 (1) (2006) 20–30. doi:10.1016/j.laa.2005.07.021.

- [50] Y. Zhang, An improved incremental singular value decomposition and new error bounds, arXiv:2204.05398v3 (updated 2 May 2026); v1 titled "An answer to an open question in the incremental SVD" (2026). arXiv:2204.05398.
- [51] A. Wang, H. Chen, L. Liu, K. Chen, Z. Lin, J. Han, G. Ding, YOLOv10: Real-time end-to-end object detection, arXiv:2405.14458, verified on arXiv (v2) on 2026-10-24 (2024). arXiv:2405.14458.
- [52] T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, P. Dollár, Microsoft COCO: Common objects in context, arXiv:1405.0312, verified on arXiv (v3) on 2026-10-24 (2014). arXiv:1405.0312.
- [53] J. Feng, H. Xu, S. Yan, Online robust PCA via stochastic optimization, in: Advances in Neural Information Processing Systems 26 (NIPS 2013), 2013, pp. 404–412.
- [54] P. L. Rosin, Unimodal thresholding, Pattern Recognition 34 (11) (2001) 2083–2096. doi:10.1016/S0031-3203(00)00136-9.