ICPR 2026, Lyon, France

KRAfT: Kalman Residual Diffusion with Formation Awareness for UAV Swarm Tracking

Md. Hasibur Rahman and Sanjay Madria

Dept. of Computer Science,

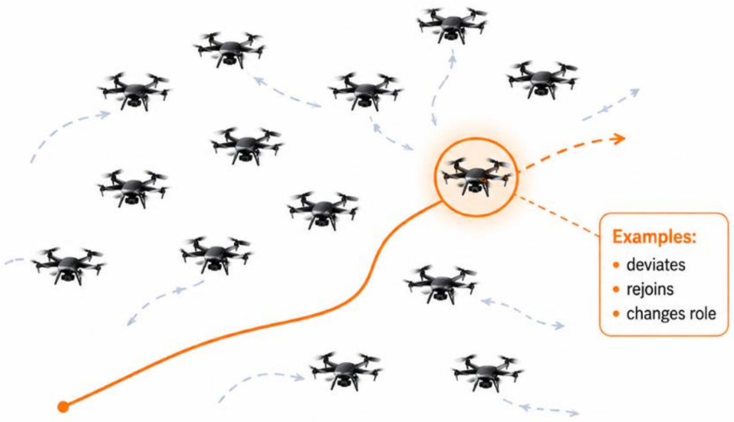
Missouri University of Science & Technology, Rolla, MO, USA

This research was partially sponsored by the Army Research Office (ARO)

August 17, 2026

Motivation

- UAV swarms provide **wide-area coverage, rapid response, and cooperative operation** for defense, surveillance, disaster response, and airspace monitoring.

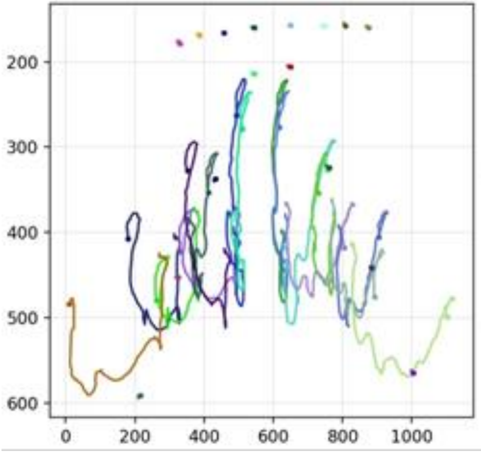


- **Why Individual Tracking?**
  - Swarm-level detection tells us **where the group is**, but not:
  - Who is who? • Who deviates? • Who disappears/rejoins? • How does the swarm evolve?

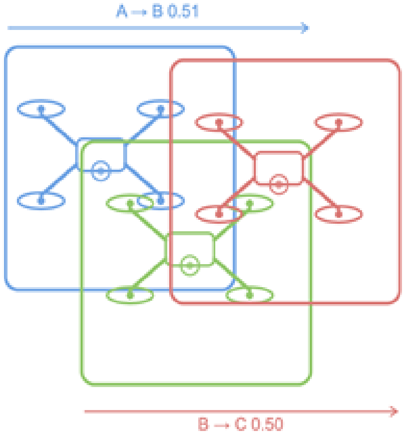
- **Why It Matters**
  - ✓ Persistent UAV identities enable:
  - ✓ Situational awareness • Behavior/anomaly analysis • Formation understanding • Reliable decision making

Swarm Tracking Challenges

- UAVs are tiny
- UAVs look very similar
- Occlusion is frequent
- **Motion is agile and nonlinear**
- Dense swarms create ambiguous associations
- IoU + Kalman is often not enough in dense swarms



Non-linear agile motion



Simple IoU confusion

Limitations of Prior Trackers

- Kalman-based trackers *(ByteTrack ECCV'22)*
  - Stable and low cost
  - weak for sharp nonlinear motion
- Diffusion / learned motion trackers *(DiffMOT CVPR'24)*
  - capture nonlinear motion
  - can drift with noisy or missing detections while depend on short term history
- Appearance-based association *(BoTSorT arXiv'22)*
  - less useful because UAVs are visually similar
- Need a tracker that is stable, handle non-linearity and agility, and is geometry-aware

KRAfT Overview

- Kalman Residual Diffusion Fusion **(KDF)** for better motion prediction

$$\Delta^{KF}_{t-1\to t} = \hat{\mathbf{x}}^{KF}_{t|t-1} - \hat{\mathbf{x}}_{t-1|t-1}$$

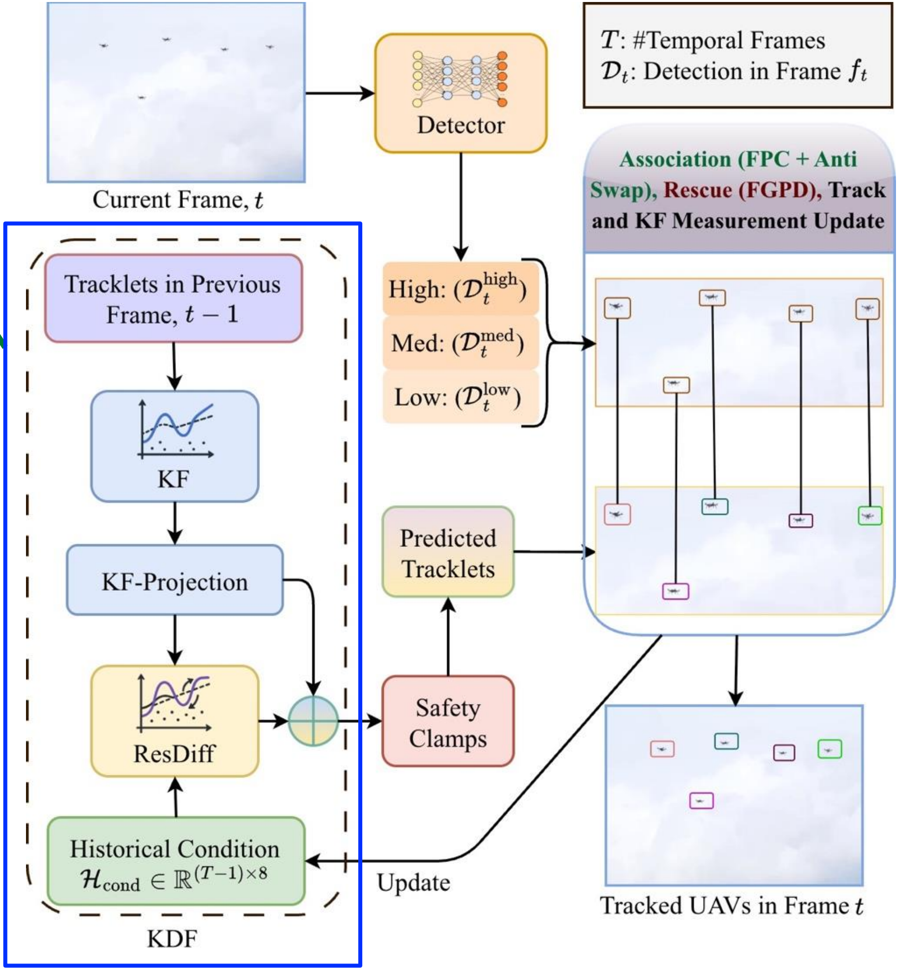
$$\widetilde{\Delta} = \Delta^{KF}_{t-1\to t} + \gamma\, \widehat{\Delta}^{\text{Diff}}$$

Kalman Motion

DM's Residual correction

$$\hat{\mathbf{x}}_{t|t-1} = \hat{\mathbf{x}}_{t-1|t-1} + \widetilde{\Delta}.$$

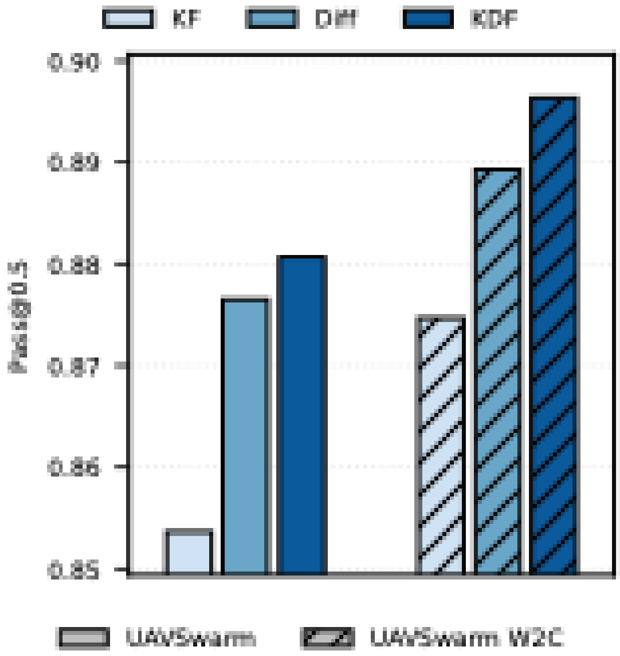
KDF



KDF: Better Motion Prediction

- Maintain track history over recent frames
- Predict next state using **Kalman filter**
- Learn a **residual displacement** from motion history using diffusion process
- Fuse them:
  - **final prediction = Kalman prediction + residual correction**
- Apply **safety clamps** to avoid unrealistic jumps

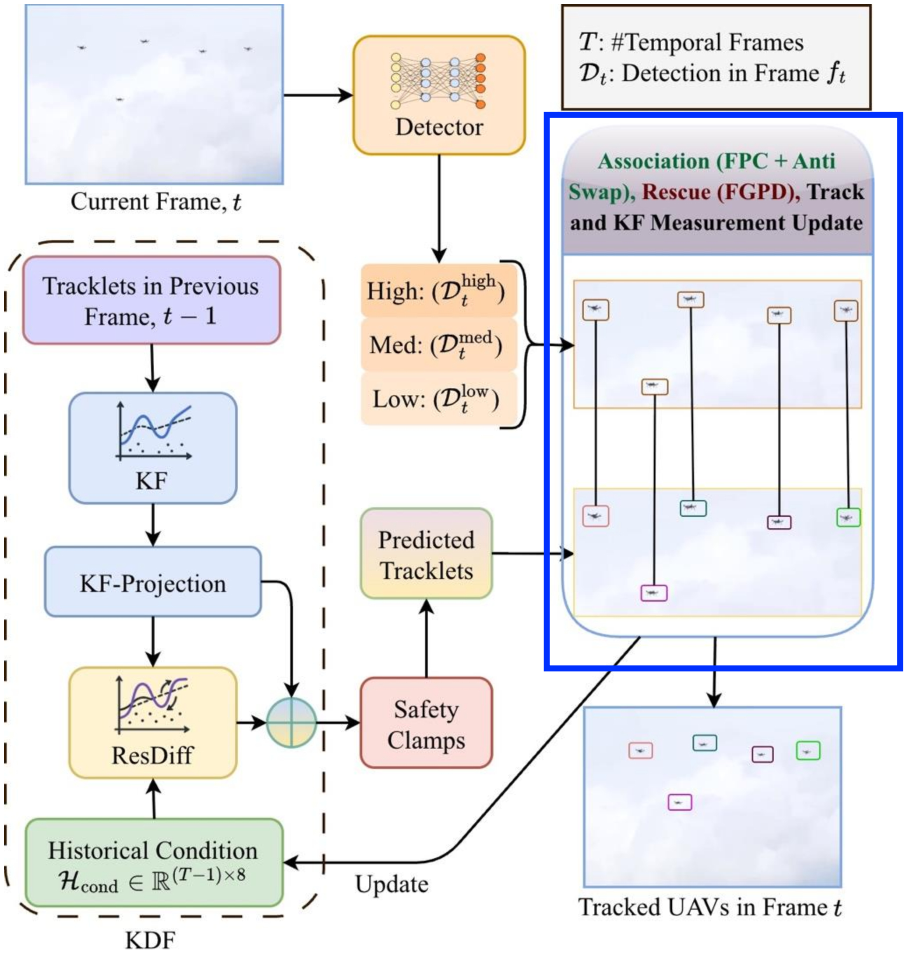
Kalman gives stability, diffusion-style residual handles agility



Pass@0.5 prediction quality of the predictor. An object is counted as a pass if the pre-update motion prediction for an identity achieves IoU ≥ 0.5 with its ground truth box (higher is better)

KRAfT Overview

- Formation-Aware Association Using Procrustes Alignment Cost (FPC) > measures how well a set of current detections matches the predicted formation.
- Formation-Guided Pseudo Detections (FGPD) for missing handling >
- **Observed UAVs + known formation → estimated location of missing UAV → pseudo detection**



Formation-Aware Association

FPC: Use swarm geometry to match identities

- Hungarian association using IOU first $C_{\mathrm{IoU}}(i,j) = 1 - \mathrm{IoU}_{ij}.$
- For a tracking ID, if there are two detections have **close IOU score**, then FPC is used to resolve the ambiguity
- For a given set of non-ambiguous matching between the UAVs, we calculate rotation and scale matrix

$$(\hat{s}_{ij}, \hat{R}_{ij}) = \arg \min_{s>0,\ R\in \mathrm{SO}(2)} \sum_{n\in\mathcal{N}_i} \left\| \tilde{q}_n - s\,R\,\tilde{p}_n \right\|_2^2.$$

- Normalized residuals and the formation penalty

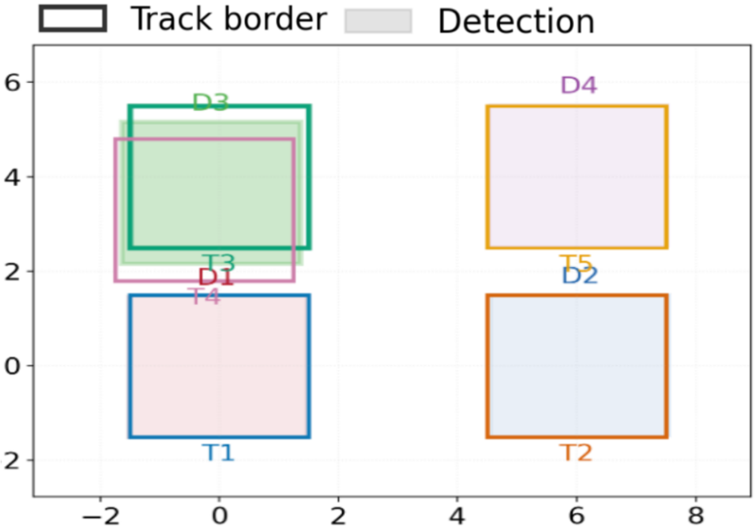
$$r_n^{(ij)} = \frac{\left\| \tilde{q}_n - \hat{s}_{ij}\hat{R}_{ij}\tilde{p}_n \right\|_2}{\sqrt{W^2 + H^2}}, \qquad \phi_{ij} = \mathrm{Huber}\left(\mathrm{median}_{n\in\mathcal{N}_i}\, r_n^{(ij)}\right).$$

Formation-Aware Association (Continued)

The fused cost is,

$$C_F(i,j) = \left(1 - \mathrm{IoU}_{ij}\right) + \lambda\,\phi_{ij} + \omega\, d_{ij}^{\mathrm{app}}$$

> penalizing matches where visual appearance doesn't match



Stable Association:

$T_1 \to D_1, T_2 \to D_2, T_3 \to D_3, T_5 \to D_4$

$\Sigma C_{\mathrm{IoU}} = 0.411 \quad \Sigma\phi = 0.01 \quad C_F = 0.421$

Drift Association:

Replacement drift $T_4 \to D_3, T_3$ unmatched

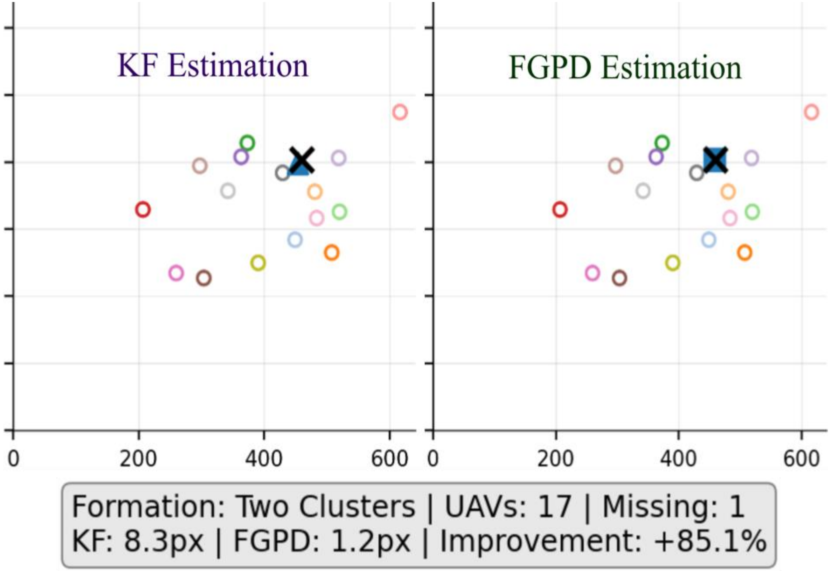
$\Sigma C_{\mathrm{IoU}} = 0.403 \quad \Sigma\phi = 0.160 \quad C_F = 0.499$

$\sum C_{\mathrm{IoU}}$ picks: ***Drift***, C_F picks: ***Stable***

Rescue Mechanism for Missed Detections

FGPD: preserve continuity during short detector failures

- Problem:
  - detector may temporarily miss UAVs
  - this causes broken tracks and ID changes
- Solution:
  - infer missing UAV position from matched neighbors while satisfying Procrustes alignment
  - create **temporary pseudo-detection**
  - use it only conservatively and briefly



Neighbors are circles, GT is ×, KF is △, FGPD is □

Datasets and Evaluation

Tab 1: Datasets Statistics

Dataset	Seqs	Frames	FPS	#UAVs/ Frame	Dur. (s)	Head. ent. ↑	Speed std. ↑	Scenario
UAVSwarm-W2C (Ours)	80	21,896	25	1–50	9.12	**0.714**	**55.132**	Real-world, most agile
UAVSwarm (Sensors'22)	72	12,598	30	1–24	5.83	0.569	41.403	Controlled swarm
MUAV (CVPR'25)	200	151,773	25	1–40	30.35	0.692	30.498	Longer sequences

Evaluation Metrics

- Performed K=3-fold cross validation
- HOTA (primary), AssA, DetA, IDF1, MOTA, and FPS

$$HOTA = \sqrt{DetA.AssA} \qquad MOTA = 1 - \frac{FN + FP + IDSW}{GT}$$

Result Analysis

Tab 2: Detection (YOLOv12) results in three datasets.

Dataset	Precision	Recall	mAP_{50}	$mAP_{50:95}$
UAVSwarm–W2C	0.962	0.908	0.953	0.579
UAVSwarm	0.950	0.916	0.951	0.519
MUAV	0.931	0.918	0.949	0.568

Tab 3: Trackers Performance in MUAV

Tracker	**HOTA**	**DetA**	**AssA**	**MOTA**	**IDF1**
BoT-SORT[1]	63.26	63.57	64.30	73.89	76.58
DiffMOT[11]	63.87	64.06	64.76	75.15	77.42
GroupTrack[19]	63.61	63.86	64.59	74.69	77.09
TrackTrack[13]	63.83	63.50	64.77	74.82	77.61
KRAfT (ours)	**65.31**	**65.01**	**66.53**	**77.07**	**79.92**

Tab 4: Trackers Performance in UAVSwarm and UAVSwarm-W2C

Tracker	**UAVSwarm**					**UAVSwarm–W2C**				
	HOTA	**DetA**	**AssA**	**MOTA**	**IDF1**	**HOTA**	**DetA**	**AssA**	**MOTA**	**IDF1**
ByteTrack[20]	72.65	73.96	71.01	84.73	84.33	65.79	67.83	64.03	84.10	78.88
BoT-SORT[1]	74.46	75.14	73.58	86.09	86.12	66.66	69.42	64.30	84.49	79.11
OC-SORT[4]	74.97	75.59	74.25	87.28	86.45	67.22	**69.80**	64.59	85.12	80.47
GroupTrack[19]	76.06	75.64	76.08	89.50	87.58	67.69	69.79	65.54	85.15	81.26
TrackTrack[13]	76.24	75.47	77.58	91.78	90.23	67.81	69.30	66.76	85.12	81.24
DiffMOT[11]	77.86	76.18	79.86	91.39	91.95	67.43	69.60	65.56	85.71	80.84
KRAfT (ours)	**80.37**	**78.39**	**83.23**	**93.77**	**94.32**	**70.59**	69.79	**71.19**	**86.10**	**85.99**

Result Analysis (Ablations)

Tab 5: Ablation on the UAVSwarm–W2C (fold 3) dataset demonstrates progressive gains when Anti-Swap with FPC, FGPD, and the appearance cost with FPC are added to the KDF predictor

Variant	HOTA ↑	DetA ↑	AssA ↑	MOTA ↑	MOTP ↑	IDSW ↓	IDF1 ↑
ByteTrack (KF + IoU)	66.03	67.00	65.44	81.76	80.63	453	79.21
DiffMOT (Diffusion + IoU)	67.16	71.32	63.72	83.96	80.54	351	78.39
KDF + IoU	68.32	71.33	65.02	84.01	80.78	339	80.74
+ FPC + Local Refinement	69.88	71.36	67.98	84.07	81.05	312	82.55
+ FGPD Rescue	70.96	71.40	70.15	84.14	81.29	301	83.94
+ Appearance Cost = KRAfT	**71.38**	**71.42**	**71.35**	**84.18**	**81.55**	**297**	**84.76**

Result Analysis (Qualitative)

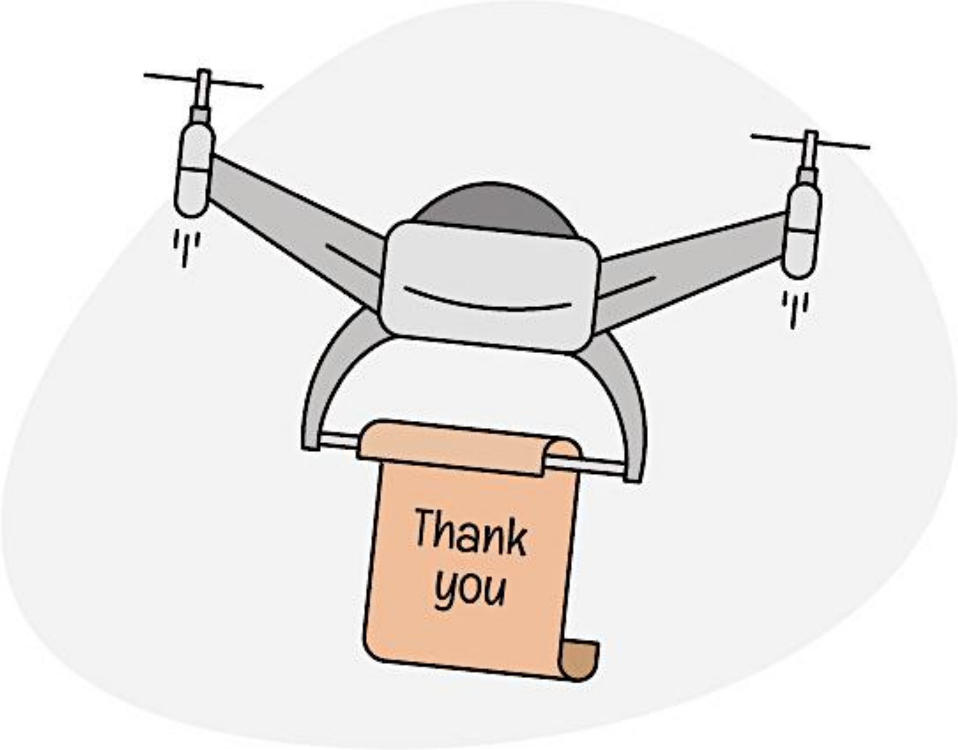
At frame 83, the detector misses track 211, but KRAfT recovers it at frame 92 with the same ID via FGPD.

Result Analysis (Qualitative)

KRAfT on top is maintaining consistent IDs, while baseline tracker (Track-Track) in bottom suffers from ID switching. Color change between the frames means ID switch.

Conclusion and Future Work

- Stability + nonlinear motion + formation geometry
- Perform better across three different swarm datasets
- Improves identity preservation significantly
- May still fail when a UAV behaves very differently from its neighbors
- Formation cue becomes weaker when neighborhood support is poor
- Future work: work in weak or no formation consistency


Thank you