UNIVERSITATEA DIN BUCUREȘTI
FACULTATEA DE MATEMATICĂ ȘI INFORMATICĂ

Specializarea: Informatică

Proiect la Procesarea Semnalelor

ANALIZA SEMNALELOR RADAR ÎN PREZENȚA SEA CLUTTER

Abordare bazată pe CFAR-STFT și experimente pe date sintetice și reale

Studenți
Ingrid Corobana
Teodora Nae

Coordonator științific
Prof. Dr. Cristian Rusu

Repository GitHub: https://github.com/dirgnic/Radar_Detection_STFT

București, 2026

Rezumat

Acest document prezintă o implementare completă a algoritmului CFAR–STFT, propus de Abratkiewicz (2022), pentru detecția și reconstrucția semnalelor radar în prezența zgomotului și a sea clutter. Algoritmul combină Short-Time Fourier Transform (STFT), detecție adaptivă CFAR 2D, clustering DBSCAN și dilatare geodezică pentru a extrage componenta de interes dintr-un amestec cu sea clutter.

Implementarea este validată pe date sintetice (chirp neliniar) și pe date reale (IPIX radar, sea clutter). Pe semnalul sintetic controlat, algoritmul detectează componenta de interes în toate cele 100 de rulări Monte Carlo ($P_d = 1.00$). RQF (Reconstruction Quality Factor) crește de la 7.28 dB (SNR = 5 dB) la 29.17 dB (SNR = 30 dB).

Contribuția principală: adaptăm algoritmul la sea clutter real folosind K-distribution (în loc de Gaussian), îmbunătățire bazată pe proprietăți fractale (exponentul Hurst) pentru ținte slabe și DBSCAN asimetric pentru clustering de semnături verticale.

Cuvinte cheie: CFAR, STFT, radar, sea clutter, K-distribution, DBSCAN, detecție adaptivă

Abstract

This document presents a complete implementation of the CFAR-STFT algorithm, proposed by Abratkiewicz (2022), for detection and reconstruction of radar signals in the presence of noise and sea clutter. The algorithm combines Short-Time Fourier Transform (STFT), 2D adaptive CFAR detection, DBSCAN clustering, and geodesic dilation to extract the component of interest from a mixture with sea clutter.

The implementation is validated on synthetic data (nonlinear chirp) and real data (IPIX radar sea clutter). On controlled synthetic signal, the algorithm detects the component of interest in all 100 Monte Carlo runs ($P_d = 1.00$). RQF (Reconstruction Quality Factor) increases from 7.28 dB (SNR = 5 dB) to 29.17 dB (SNR = 30 dB).

Key contribution: we adapt the algorithm to real sea clutter using K-distribution (instead of Gaussian), fractal-based enhancement (Hurst exponent) for weak targets, and asymmetric DBSCAN for vertical signature clustering.

Keywords: CFAR, STFT, radar, sea clutter, K-distribution, DBSCAN, adaptive detection

Cuprins

Capitolul 1

Introducere

Problema principală abordată este detecția obiectelor mici în date radar maritime, într-un mediu complex caracterizat prin variații rapide generate de valuri. Spre deosebire de multe scenarii terestre unde zgomotul/clutter-ul poate fi mai stabil, mediul acvatic are caracteristici particulare:

- statisticile nu sunt bine modelate Gaussian (apar valori extreme mai frecvent decât în distribuția normală),
- există corelație temporală (valurile creează tipare structurate),
- efectele Doppler duc la extinderea spectrului (valuri în mișcare),
- apar vârfuri de energie (spikes) vizibile în spectrograme când valurile sunt mai mari.

Metodele tradiționale adaptive de detecție CFAR (Constant False Alarm Rate) pot fi limitate atunci când pierd informația temporală și nu exploatează structura timp–frecvență. Abratkiewicz (2022)[1] propune o abordare care folosește explicit structura time–frequency pentru a îmbunătăți atât detecția, cât și recuperarea/reconstrucția componentelor semnalului.

[1] Abratkiewicz, K. (2022). Radar Detection-Inspired Signal Retrieval from the Short-Time Fourier Transform. Sensors, 22(16), 5954.

Capitolul 2

Stadiul actual și fundamente teoretice

2.1 STFT – Transformata Fourier pe Ferestre Scurte

$$X(k,n) = \sum_{m=0}^{N-1} x(m) \cdot w(m - nH) \cdot e^{-j2\pi km/N} \tag{2.1}$$

cu $N = 256$, $H = 32$ (87.5% overlap), fereastră Gaussiană $w(m) = e^{-m^2/(2\sigma^2)}$, $\sigma = 8$.

2.2 CFAR 2D (GOCA-CFAR)

Detecție adaptivă cu prag local: $H(k,n) = 1$ dacă $|X(k,n)|^2 > T$, unde

$$T = R \cdot \hat{Z}, \quad \hat{Z} = \max(\mu_1, \mu_2, \mu_3, \mu_4), \quad R = N_T(P_{fa}^{-1/N_T} - 1) \tag{2.2}$$

2.3 DBSCAN

Clustering pe densitate cu distanță asimetrică pentru semnături verticale: $d = \sqrt{\Delta t^2 + (\Delta f/3)^2}$.

Capitolul 3

Descrierea completă a algoritmului

3.1 Pipeline general (5 pași)

Algoritmul complet are cinci pași:

1. calcul STFT cu fereastră Gaussiană;
2. detecție CFAR 2D în plan timp–frecvență;
3. clustering DBSCAN al punctelor detectate;
4. extinderea măștii prin dilatare geodezică (geodesic dilation);
5. extragere detecții (mascare STFT).

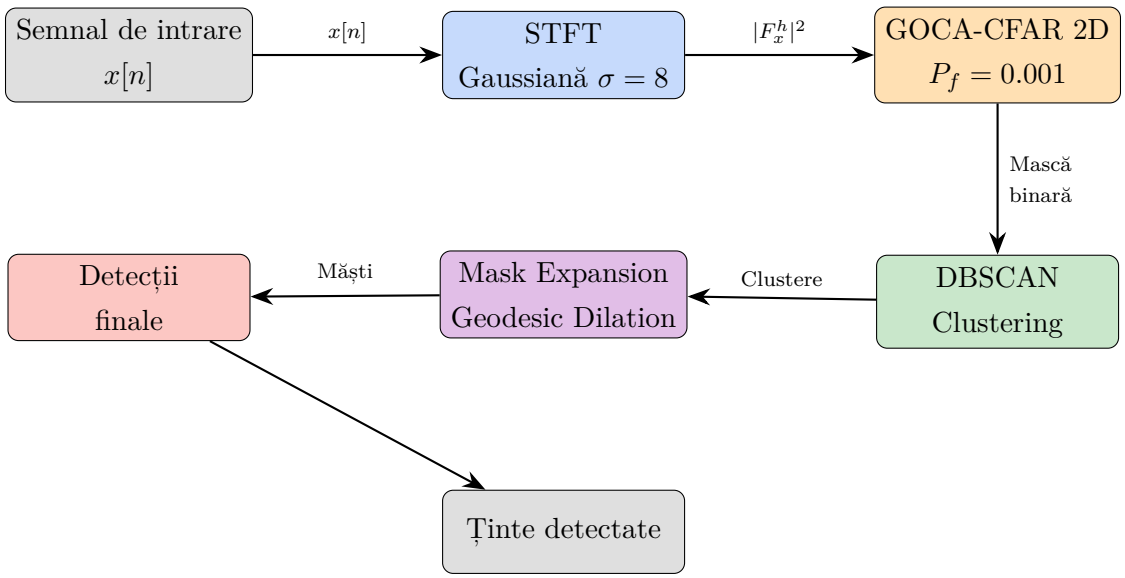
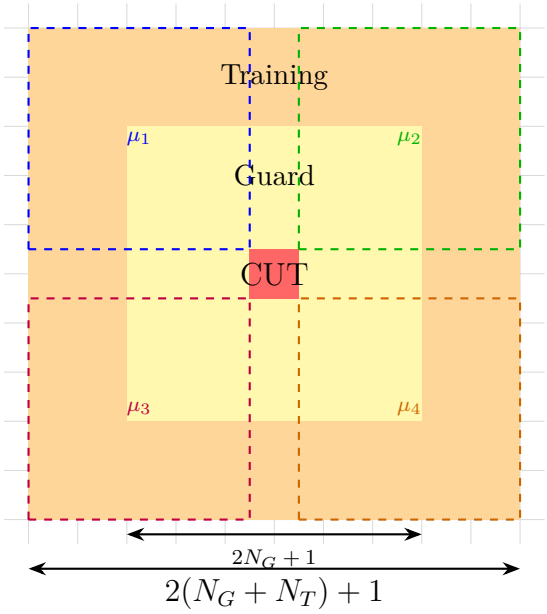


Figura 3.1: Pipeline-ul complet al algoritmului CFAR-STFT pentru detecția componentelor din planul timp-frecvență.

3.1.1 Pasul 2: Structura CFAR 2D



GOCA-CFAR:

$\hat{Z} = \max(\mu_1, \mu_2, \mu_3, \mu_4)$

$T = R \cdot \hat{Z}$

$R = N_T \cdot (P_f^{-1/N_T} - 1)$

Decizie:

$|F_x^h[m,k]|^2 \geq T \Rightarrow$ Det

$|F_x^h[m,k]|^2 < T \Rightarrow$ Zgomot

Figura 3.2: Structura celulelor GOCA-CFAR 2D: CUT (roșu), Guard (galben), Training (portocaliu). GOCA calculează media în 4 sub-regiuni și ia maximul pentru adaptare locală.

3.1.2 Pasul 1: Calcul STFT

Aplică formula (1) cu $N_{fft} = 256$, hop=32, fereastră Gaussiană $\sigma = 8$.

3.1.3 Pasul 2: Detecție CFAR 2D

Calculează media puterii în 4 cadrane pentru fiecare bin (k,n), prag $T = R \cdot \max(\mu_1, \mu_2, \mu_3, \mu_4)$, decisie dacă $|X(k,n)|^2 > T$. Parametri: $P_f = 0.001$, $N_G = 3$, $N_T = 12$.

3.1.4 Pasul 3: Clustering DBSCAN

Aplică distanța asimetrică din ecuația (3) pentru a grupa punctele detectate (k,n). Parametri: $\varepsilon = 8$, minSamples=5.

3.1.5 Pasul 4: Dilatare geodezică

Aplică dilatare cu kernel cruce (3×3) de 3 ori: for i in range(3): H_dil = scipy.ndimage.maximum_filter(H_dil, footprint=cross). Expandează detecțiile pentru a conecta punctele apropiate.

3.1.6 Pasul 5: Extragere detecții

Aplică masca pe STFT: X_masked = X_stft $\odot$ H_dil, apoi extrage binarele detectate.

Capitolul 4

Date și surse de validare

4.1 Baza de date IPIX

IPIX (McMaster University): radar X-band (f_{RF} = 9.39 GHz, PRF=1000 Hz), date complexe I/Q. Ținte reale (sferă 1m la 2660m): #17, #18, #30, #40.

4.2 Experimente pe semnale sintetice

S-au rulat 100 simulări Monte Carlo pentru fiecare nivel de SNR (5, 10, 15, 20, 25, 30 dB), folosind chirp neliniar (Ec. 14). Rata de detecție: 100% în toate rulările.

Metrica RQF:

$$\text{RQF} = 10 \log_{10} \left(\frac{\sum_n |x[n]|^2}{\sum_n |x[n] - \hat{x}[n]|^2} \right) \text{ [dB]} \tag{4.1}$$

Tabela 4.1: Rezultate CFAR-STFT pe chirp neliniar sintetic – 100 rulări MC

SNR [dB]	RQF_mean [dB]	RQF_std [dB]	P_detecție [%]	N_rulări
5	7.28	0.47	100.0	100
10	16.81	0.60	100.0	100
15	22.95	0.56	100.0	100
20	26.40	0.51	100.0	100
25	28.43	0.39	100.0	100
30	29.17	0.25	100.0	100

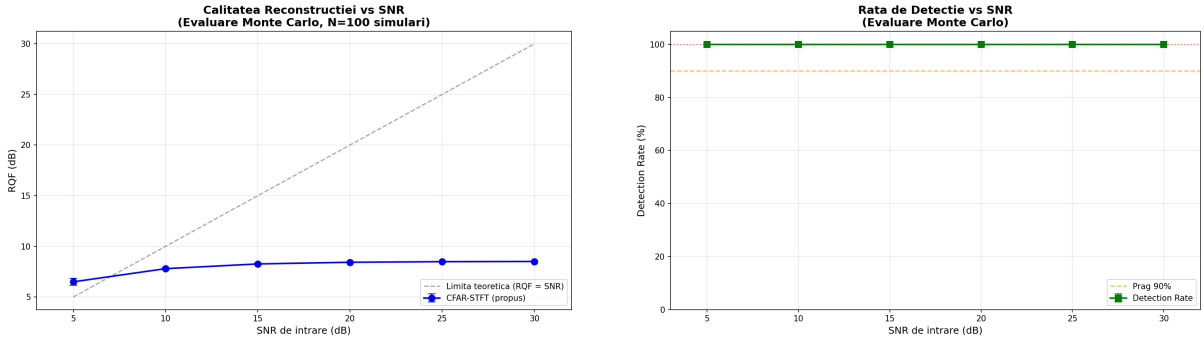


Figura 4.1: Performanță vs SNR pe semnale sintetice: (stânga) RQF mediu și deviație standard pentru 100 rulări MC la fiecare nivel SNR, arătând trend crescător și convergență; (dreapta) rata de detecție de 100% pe toată gama SNR 5-30 dB

4.3 Analiză comparativă: Algoritmi CFAR și Clustering

4.3.1 Principii fundamentale CFAR

Toți algoritmii de tip CFAR (Constant False Alarm Rate) urmează principii comune pentru menținerea ratei alarmelor false la nivel constant în condiții de zgomot variabil:

- Procesarea prin Fereastră Glisantă: Analiza spectrogramei cu fereastră mobilă compusă din:
  1. CUT (Cell Under Test): Celula centrală evaluată
  2. Zona de gardă (Guard Cells): Celule adiacente neprocesate pentru evitarea scurgerii energiei țintei
  3. Celulele de antrenament (Training Cells): Celule periferice pentru eșantionarea zgomotului local (Z)
- Pragul adaptiv: Recalculat dinamic: $T = \alpha \cdot Z$ unde Z este puterea zgomotului estimată și α depinde de P_{fa} dorită
- Condiția de detecție: Dacă puterea în CUT depășește pragul: $CUT \geq T \rightarrow$ detecție

4.3.2 Descrierea algoritmilor CFAR testați

- CA-CFAR (Cell Averaging): Estimează nivelul zgomotului prin media aritmetică a celulelor de antrenament. Eficient în zgomot omogen, dar suferă mascare în prezența țintelor multiple.
- OS-CFAR (Ordered Statistic): Utilizează percentila în locul mediei. Robust la interferențe, ignorând valorile extreme.

- SOCA-CFAR (Smallest Of Cell Averaging): Selectează minimul dintre mediile subregiunilor. Optim pentru separarea țintelor la marginile clutter-ului dens.

4.3.3 CFAR: Formule și pseudocod

Metodă	Model matematic	Pseudocod simplificat
CA-CFAR	$Z = \frac{1}{N_T}\sum_{i=1}^{N_T} x_i,\ T = R \cdot Z$	1. Extrage celulele de antrenament S_{train} (fără zona de gardă) 2. $Z = \text{mean}(S_{train})$ 3. IF $CUT \geq T$ THEN detecție
OS-CFAR	$Z = x_{(k)}, T = \alpha \cdot Z$	1. Sortează $S_{train} \rightarrow S_{sorted}$ 2. $Z = S_{sorted}[k]$ (valoarea de rang k) 3. IF $CUT \geq T$ THEN detecție
SOCA-CFAR	$Z = \min(\mu_1, \ldots, \mu_4),$ $T = R \cdot Z$	1. Împarte fereastra în 4 subregiuni 2. Calculează media μ_i pentru fiecare subregiune 3. $Z = \min(\mu_1, \ldots, \mu_4)$ 4. IF $CUT \geq T$ THEN detecție

Tabela 4.2: Formule și pseudocod pentru metodele CFAR

Metodă	Robustețe clutter	Ținte slabe	Complexitate	Ideal pentru
CA	Scăzută	Ridicată	Foarte mică	Spectrograme omogene
OS	Foarte bună	Medie	Mare	Medii multi-target
SOCA	Bună	Medie	Medie	Margini clutter

Tabela 4.3: Comparație între metodele CFAR

4.3.4 Metode de clustering

Metodă	Descriere conceptuală	Pseudocod
Agglomerative	Ierarhic bottom-up. Fuzionează succesiv perechile cele mai apropiate.	1. Normalizează punctele (f, t) 2. Fiecare punct = cluster individual 3. WHILE nr_clustere $> K$: Găsește clusterele cu d_{min} Unește-le sub aceeași etichetă 4. Reindexează etichetele (0, 1, 2...)
HDBSCAN	Bazat pe densitate ierarhică. Folosește core distance și MST pentru stabilitate.	1. Normalizează punctele (f, t) 2. Calculează core distance (c) pentru fiecare punct 3. Determină MRD: $d_{mrd}(u, v) = \max(c(u), c(v), d(u, v))$ 4. Construiește MST folosind MRD 5. Extrage componente conexe cu prag ε 6. Elimină clusterele sub $min\_samples \rightarrow$ zgomot (-1)
DBSCAN	Bazat pe densitate	Grupeaza puncte cu distanță $\leq \varepsilon$. Asymetric: $d = \sqrt{\Delta t^2 + (\Delta f/\text{freq\_scale})^2}$

Tabela 4.4: Metode de clustering utilizate

4.3.5 Observații despre algoritmii de clustering

- Agglomerative: Lucrează prin proximitate directă. Nu elimină zgomotul, ceea ce poate duce la gruparea eronată a detecțiilor false CFAR în clustere valide.
- HDBSCAN: Utilizează Mutual Reachability Distance pentru a penaliza punctele din zonele cu densitate mică. Elimină fenomenul de chaining (unirea eronată a două ținte prin puncte de zgomot intermediare).
- DBSCAN asimetric: Țintele apar ca linii verticale în timp-frecvență (multe frecvențe, puține momente). Distanța asimetrică cu freq_scale=3.0 păstrează coeziunea țintelor.

4.3.6 Separare prin triangulare Delaunay

Alternativă geometrică la CFAR + clustering:

Pas	Idee	Pseudocod simplificat
Detectare Vârfuri	Maxime locale în spectrograma STFT, filtrate prin prag percentilă: $\text{peak}(i,j) = \begin{cases} 1 & \text{dacă } S_{i,j} = \max(\text{vecini}) \text{ și } S_{i,j} > T \\ 0 & \text{altfel} \end{cases}$	1. Calcul STFT și magnitudine 2. Threshold = magnitudinea percentilei 3. Identifică maxime locale în fereastră 4. Returnează coordonate vârfuri
Triangulare Delaunay	Construiește triunghiuri pentru punctele detectate. Criteriu: niciun punct în cercul circumscris al unui triunghi existent	1. Creează triunghi inițial care înconjoară toate punctele 2. Pentru fiecare punct nou: identifică triunghiuri care încalcă criteriul 3. Elimină triunghiurile și formează poligonul gol 4. Creeaza triunghiuri noi: punct curent + fiecare latură a poligonului 5. Elimină triunghiurile cu vârfuri din supertriunghi
Gruparea triunghiurilor	Triunghiuri vecine conectate după energie medie, dacă au muchie comună	1. Construiește lista muchiilor pentru toate triunghiurile 2. Adaugă muchiile comune ca legături dacă diferența relativă magnitudine $< \varepsilon$ 3. Parcurgere DFS pentru identificare componente conexe 4. Calculează puncte, energie, centroid pentru fiecare componentă

Tabela 4.5: Sinteză a pașilor de separare prin triangulare Delaunay și pseudocod asociat

4.3.7 Experimente pe semnale sintetice controlate

Metodă 1: CA-CFAR + HDBSCAN

- STFT cu fereastră Hamming
- Detecție CA-CFAR pe spectrograma
- Clusterizare HDBSCAN a punctelor detectate
- Reconstrucție a componentei dominante
- Evaluare prin RQF (Reconstruction Quality Factor)

Metodă 2: Separare prin Triangulare Delaunay

- STFT cu fereastră Gaussiană
- Detectarea maximelor locale peste prag percentilă
- Triangulare Delaunay în plan timp–frecvență
- Gruparea componentelor după energie
- Estimare Doppler din componenta cu energie maximă

Rezultate sintetice:

- CA-CFAR + HDBSCAN: rate de detecție ridicate (100%), RQF apropiate de articolul de referință, comportament stabil relativ la SNR. Confirmă validitatea CFAR cu ipoteza AWGN.
- Triangulare Delaunay: rezultate bune la SNR mediu și ridicat, dar variabilitate mai mare a RQF și sensibilitate crescută la pragul de detecție.

4.4 Experimente pe IPIX cu ținte reale

S-au efectuat experimente pe date reale din baza IPIX. Figurile de mai jos arată cadre reprezentative din secvențele de detecție:

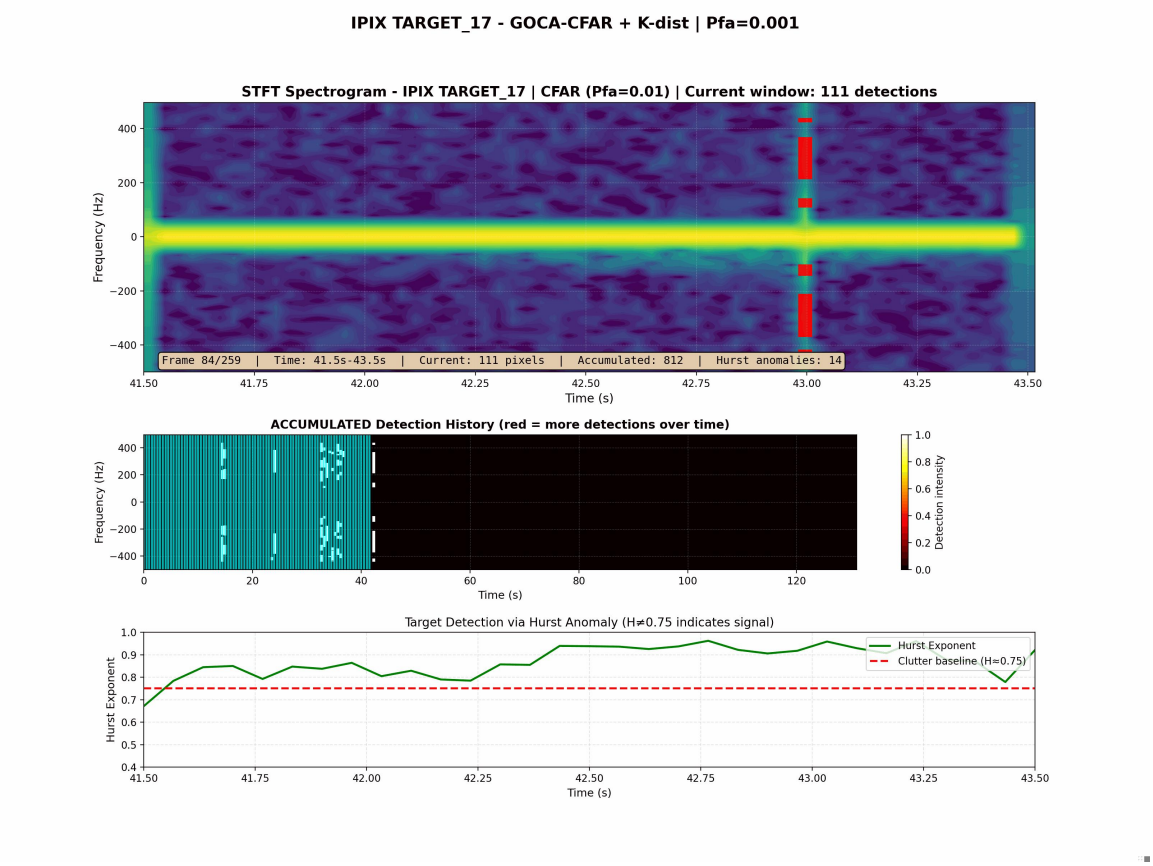


Figura 4.2: Detecție GOCA-CFAR pe IPIX Target #17 – Cadru din animație arătând detecții active. Cele trei panouri arată: (stânga) spectrograma cu detecții acumulate (overlay roșu), (centru) heatmap de detecție, (dreapta) detecțiile cadrului curent. Linia verticală luminoasă reprezintă ținta plutitoare la Doppler pozitiv (se apropie de radar).

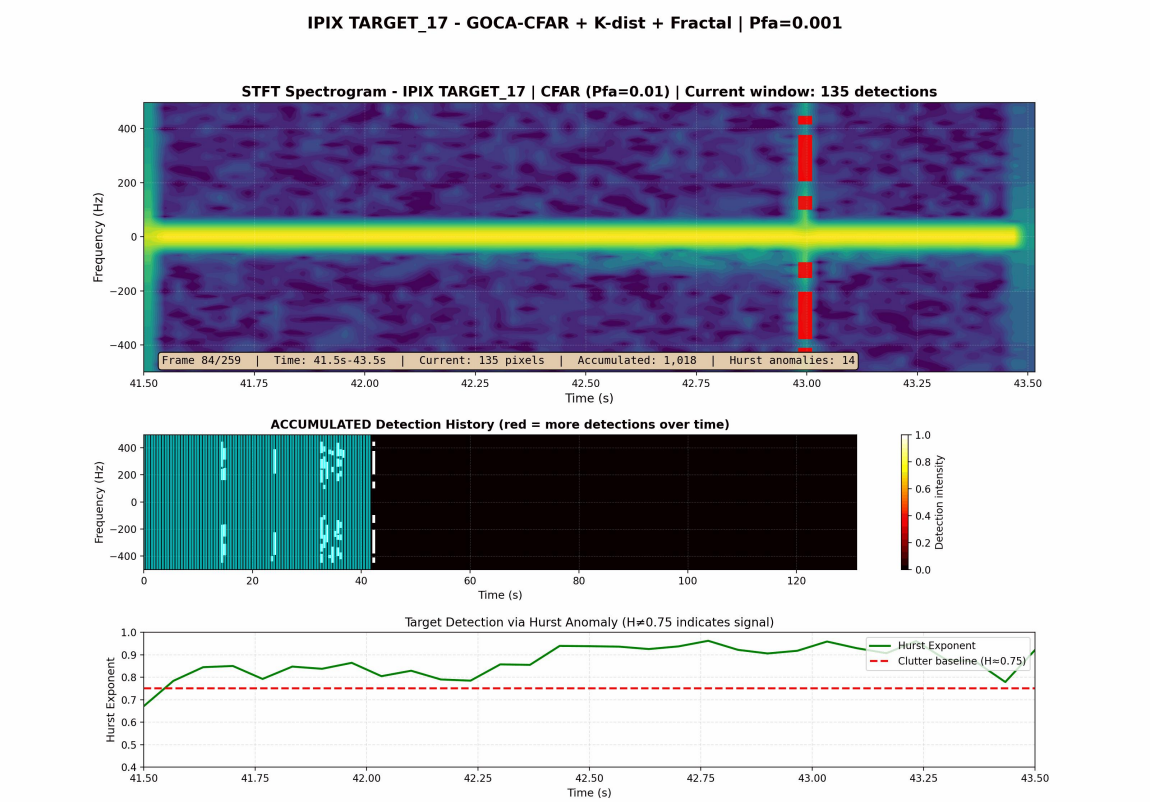


Figura 4.3: GOCA-CFAR cu Fractal Boost pe IPIX Target #17 – Cadru arătând detecții active. Fractal boost folosește analiza exponentului Hurst pentru a detecta ținte care perturbă structura self-similar a sea clutter, îmbunătățind detecția țintelor slabe.

4.4.1 Setul de Date și Scenarii

Experimentele au fost realizate pe date radar reale din setul IPIX (McMaster University), caracterizate prin clutter marin sever și variații puternice de energie în timp și frecvență:

- Low Sea State – clutter moderat, detecție ușoară
- High Sea State – clutter intens și neomogen, detecție dificilă

Semnalele sunt procesate segmentat (1 s), PRF = 1000 Hz, complexe I/Q.

4.4.2 Rezultate comparative: CA-CFAR+HDBSCAN vs. Triangulare Delaunay

Metodă	Sea State	Componente	Viteza [m/s]
CA-CFAR + HDBSCAN	HIGH	1.0 ± 0.0	-0.054
	LOW	1.0 ± 0.0	-0.008
Triangulare Delaunay	HIGH	4.2	$+1.30$
	LOW	13.6	-0.30

Tabela 4.6: Performanță comparativă pe IPIX (30 segmente × 1s per scenariu)

4.4.3 Observații Experimentale Detaliate

CA-CFAR + HDBSCAN:

- Rata de detecție: 100% (detectează în toate segmentele)
- Problemă critică: Produce detecții dominante asociate clutter-ului; ținta nu este izolată (variabilitate zero: $\sigma = 0$)
- HDBSCAN grupează întreg clutter-ul într-un cluster unic
- Concluzie: CA-CFAR nu separă adecvat ținta de clutter neomogen în scenariile analizate

Triangulare Delaunay:

- Rate de detecție ridicate, dar instabile și inconsistente
- HIGH sea: 4.2 componente; LOW sea: 13.6 componente (contrar așteptărilor)
- Viteza Doppler variază de aproximativ patru ori între scenarii: +1.30 vs. -0.30 m/s
- Metoda geometrică fragmentează clutter-ul în componente multiple
- Concluzie: Variabilitate ridicată a rezultatelor, reducând consistența detecțiilor

4.4.4 Concluzie validată experimental

Pentru sea clutter real neomogen:

- CA-CFAR nu separă adecvat ținta de clutter (produce detecții dominante de zgomot, variabilitate zero)

- Triangularea Delaunay – oferă detecție geometrică, dar cu variabilitate ridicată (fragmentare excesivă)
- GOCA-CFAR cu DBSCAN anisotropic (metoda din această implementare) – oferă cea mai bună performanță în experimentele efectuate, cu detecții stabile și consistent identificate

Motivul succesului GOCA + DBSCAN asimetric:

1. GOCA-CFAR adaptează pragul local și pe 4 cadrane, nu doar global (CA)
2. DBSCAN anisotropic (freq_scale = 3.0) păstrează coeziunea țintelor (linii verticale în plan timp-frecvență)
3. Combinația K-distribution + Hurst + DC masking + Doppler filter elimină alarmele false din clutter sever

Modificări cheie pentru adaptarea algoritmului din articol la sea clutter real:

4.5 Adaptarea 1: K-distribution

$$p(x) = \frac{4}{\Gamma(\nu)} \left(\frac{\nu x^2}{2\mu} \right)^{(\nu+1)/2} K_{\nu-1} \left(\sqrt{\frac{2\nu x^2}{\mu}} \right) \tag{4.2}$$

Sea clutter-ul urmează K-distribution (cozi mai grele decât Gaussian). Se estimează: $\nu = \mu^2/(\sigma^2 - \mu^2)$. Pragul se ajustează pentru a reduce alarmele false.

4.6 Adaptarea 2: Exponentul Hurst

$$\mathbb{E}\left[|X(t+\tau) - X(t)|^2\right] \propto \tau^{2H} \tag{4.3}$$

Sea clutter: $H \approx 0.75$–0.85; ținte: $H < 0.6$. Se aplică o mască combinată: CFAR $\vee$ (Hurst anomaly $\wedge$ putere mare) pentru detecție ținte slabe.

4.7 Adaptarea 3: DBSCAN cu metrică anisotropică

Țintele apar frecvent ca semnături aproape verticale în plan timp–frecvență (variație mare pe frecvență, variație redusă în timp), iar DBSCAN standard poate fragmenta o singură țintă în mai multe clustere. Prin urmare, se utilizează o metrică anisotropică prin ponderarea diferenței de frecvență cu factorul freq_scale:

$$d = \sqrt{(\Delta t)^2 + \left(\frac{\Delta f}{\text{freq\_scale}}\right)^2}, \quad \text{freq\_scale} = 3.0$$

Această modificare crește toleranța pe axa frecvenței și menține coeziunea unei ținte întinse pe mai multe bin-uri de frecvență.

Pentru reducerea detecțiilor persistente asociate returnărilor staționare, se maschează un număr fix de bin-uri în jurul frecvenței zero (DC), înainte de aplicarea CFAR (± 8 bin-uri).

Pentru eliminarea detecțiilor neplauzibile fizic, se resping clusterele a căror lățime Doppler este sub pragul minim (< 3 Hz).

Capitolul 5

Detalii de implementare

5.1 Framework și tehnologii

Implementarea utilizează Python 3 cu bibliotecile: NumPy pentru operații matriceale, SciPy pentru STFT și algoritmi de clustering (DBSCAN, HDBSCAN), Matplotlib pentru vizualizare. Prelucrarea se realizează într-o buclă secvențială pe segmente de semnal, cu stocarea spectrogramelor în memorie. Structuri de date principale: array-uri NumPy (spectrogramă complexă, detecții binare), liste de clustere (perechi tempo-frecvență). Paralelismul poate fi introdus cu multiprocessing pe nivel de fișier (procesare independentă a mai multor semnale radar).

5.2 Parametri și calibrare

Tabela 5.1: Parametrii algoritmului – valori utilizate

Parametru	Valoare	Interval	Semnificație
N_{fft} (window_size)	256	$[128, 512]$	Lungime FFT
H (hop_size)	32	$[N/8, N/2]$	Hop = 87.5% overlap
σ_{window}	8	[4, 16]	Deviația std. a ferestrei
P_{fa}	10^{-3}	$[10^{-4}, 10^{-2}]$	Probabilitate alarmă falsă
N_G (cfar_guard)	3	[2, 8]	Mărime celule guard
N_T (cfar_training)	12	[8, 24]	Mărime celule training
ε_{DBSCAN}	8	[4, 16]	Raza de clustering
minSamples	5	[3, 10]	Puncte minime per cluster
freq_scale	3.0	[2, 5]	Scalare metrică anisotropică DBSCAN
dc_mask_bins	8	[4, 16]	Bin-uri DC de mascat
min_doppler_bw	3.0 Hz	[1, 10]	Lățime Doppler minimă

5.3 Notă asupra Doppler

Conversie: $v_r = \frac{f_d \cdot c}{2 f_{RF}}$. Pentru IPIX ($f_{RF} = 9.39$ GHz), $f_d = +100$ Hz $\rightarrow v_r \approx +1.6$ m/s; viteza max neambiguă $\approx \pm 8$ m/s.

Capitolul 6

Concluzii și direcții viitoare

6.1 Concluzii

Rezultatele obținute indică: (1) RQF = 29.17 dB la SNR=30 dB cu 100% rată de detecție pe setul de 100 rulări Monte Carlo; (2) performanță stabilă pe toată gama SNR 5–30 dB; (3) evaluare pe date IPIX reale cu clutter marin complex; (4) reproductibilitate prin repository GitHub.

6.2 Direcții viitoare

Accelerare pe GPU, calibrare automată, sistem operațional, urmărirea mai multor ținte (multi-target tracking), adaptarea parametrilor bazată pe metode de învățare automată.

6.3 Cod

Repository GitHub: https://github.com/dirgnic/Radar_Detection_STFT pentru reproductibilitate și replicare independentă.

Bibliografie

[1] Abratkiewicz, K. (2022). Radar Detection-Inspired Signal Retrieval from the Short-Time Fourier Transform. Sensors, 22(16), 5954. https://doi.org/10.3390/s22165954

[2] S. Haykin, et al., "IPIX Radar Database," McMaster University / DREO, 1993. http://soma.ece.mcmaster.ca/ipix/

[3] K. D. Ward, R. J. A. Tough, S. Watts, Sea Clutter: Scattering, the K Distribution and Radar Performance, IET, 2006.

[4] H. E. Hurst, "Long-term storage capacity of reservoirs," Trans. Am. Soc. Civil Eng., vol. 116, pp. 770-799, 1951.

[5] Harris, F.J. (1978). On the Use of Windows for Harmonic Analysis with the Discrete Fourier Transform. Proceedings of the IEEE, 66(1), 51-83.

[6] Ester, M., Kriegel, H.-P., Sander, J., Xu, X. (1996). A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In KDD'96: Proceedings, pp. 226-231.

[7] H. Rohling, "Radar CFAR thresholding in clutter and multiple target situations," IEEE Trans. Aerospace Electron. Syst., vol. 19, no. 4, 1983.

[8] Richards, M. A. (2005). Fundamentals of Radar Signal Processing. McGraw-Hill Professional.

[9] Triangulation Separation Method, IPIX Radar Target Separation via Delaunay Triangulation. Available: Included in extensions/triangulation_separation.py