AIS + VIS + FUS Entegre İşlem Zinciri (Expert Seviyesi)

Bu belge üç ana modülün (AIS, VIS, FUS) aynı zaman çizelgesi üzerinde nasıl çalıştığını, tetik koşullarını ve veri el sıkışmalarını ayrıntılı olarak açıklar. Kurulum veya genel proje açıklaması yerine saf işleyişe odaklanır.

Integrated AIS + VIS + FUS Pipeline (Expert Synchronized View)

<!-- Added artifact references -->

\*\*Artifacts:\*\* See generated `docs/Integration\_Documentation.docx` and integration flowchart image `docs/flowchart\_integration.png` once you run the helper scripts.

# 1. Zamanlama Modeli

- Frame döngüsü: Her karede VIS (tespit + takip) çalışır.

- Saniye sınırı: `timestamp % 1000 < t` → AIS saniyelik işleme + fusion tetiklenebilir.

- FUS, önceki saniyeden gelen görsel geçmiş (Vis\_tra) + yeni saniyenin AIS verisi + geçmiş eşleşmeleri (mat\_las/bin\_las) kullanır.

# 2. Yüksek Seviye Akış

FRAME LOOP:

capture frame

update\_time

(sec?) AIS.process

VIS.feedCap (detection->tracking)

(sec?) update\_tra

(sec?) FUS.fusion

(sec?) gen\_result

draw\_traj + write/show

repeat

# 3. AIS Mikro Adımlar (Yeni Saniye)

1. initialization() → Temporary tabloları reset.

2. read\_ais(Time\_name) → Saniye CSV okuma.

3. data\_coarse\_process(...) → MMSI / hız / sıçrama / FOV / mesafe filtreleri.

4. data\_pred(...) → TIME\_OFFSET düzeltmesi, eksik saniyelerde ileri projeksiyon (geodetic fwd).

5. data\_tran(...) → transform + visual\_transform ile (lon,lat) → (x,y) piksel.

Çıktı: AIS\_cur (saniyelik) + AIS\_vis (pencereli geçmiş).

# 4. VIS Mikro Adımlar (Her Kare, Saniye Sınırında Agregasyon)

1. detection() (YOLOX)

2. anti\_occ() (kısmi; overlapping analizi)

3. track() (DeepSORT: feature → kalman predict → association → kalman update → id assign)

4. (Saniye) update\_tra() → aynı ID çoklu tespit ortalaması, hız kestirimi, pencere bakımı.

Çıktı: Vis\_cur (saniyelik) + Vis\_tra (geçmiş pencere).

# 5. FUS Mikro Adımlar (Yeni Saniye)

1. traj\_group(AIS) & traj\_group(VIS)

2. initialization() → mat\_las/bin\_las ayrıştırması.

3. cal\_similarity() → Her VIS×AIS çifti:

- Son nokta mesafesi kapısı.

- heading angle farkı.

- DTW\_fast (downsample + fastdtw + açı penalti).

- Önceki eşleşme bonusu.

4. Hungarian (linear\_assignment)

5. data\_filter() → Sert mesafe + açı eşiği.

6. save\_data() → match sayacı artışı + fog tolerance + bin\_cur güncellemesi.

Çıktı: mat\_list (fusion satırları), bin\_cur (stabil bağ).

# 6. Veri El Sıkışmaları

| Kaynak | Üretilen | Tüketen |

| AIS.process | AIS\_cur, AIS\_vis | FUS.fusion, draw\_traj |

| VIS.feedCap | Vis\_tra\_cur\_3 (geçici) | update\_tra |

| update\_tra | Vis\_cur, Vis\_tra | FUS.fusion, draw\_traj |

| FUS.fusion | mat\_list, bin\_cur | draw\_traj, gen\_result |

| gen\_result | CSV kayıtları | Analiz araçları |

| draw\_traj | Görsel overlay | Operatör |

# 7. Zaman Çizelgesi Örneği

Second boundary (t):

AIS.process -> VIS.feedCap(frame\_t) -> update\_tra -> FUS.fusion -> gen\_result -> draw\_traj

Intermediate frame (t+Δ):

VIS.feedCap(frame\_t+Δ) -> draw\_traj (fused state from last second)

# 8. Karar Noktaları

- Saniye olup olmadığı belirleyici (AIS + fusion + sonuç yazımı).

- Gating: Mesafe > max\_dis veya açı > limit → DTW yapılmaz (büyük ceza).

- Fog tolerance: Kısa süreli kayıp eşleşmeyi korur.

# 9. Performans Odakları

- En pahalı: detection().

- Orta: tracking (embedding + association).

- Düşük: AIS transform & fusion (DTW downsample sayesinde).

# 10. Hata Ayıklama İpuçları

- AIS yok → TIME\_OFFSET veya dosya adı uyuşmazlığı kontrol.

- Fusion boş → Gating eşikleri (max\_dis, açı) çok sıkı olabilir.

- Yanlış eşleşme → DTW penalty veya bonus ölçekleri gözden geçir.

# 11. Mermaid Akışı

flowchart TD

FR[Frame] --> UT[update\_time]

UT -->|sec| AISP[AIS.process]

AISP --> VISP[VIS.feedCap]

UT -->|!sec| VISP

VISP -->|sec| UPD[update\_tra]

UPD --> FUS[FUS.fusion]

FUS --> RES[gen\_result]

RES --> DRAW[draw\_traj]

VISP -->|!sec| DRAW

DRAW --> OUT[write/show]

OUT --> FR

# 12. Kısa Özet

AIS saniye başı konumsal gerçeklik -> VIS sürekli görsel kimlik -> FUS her saniye bu iki sekansı sekans-uyum (DTW) + global optimizasyon (Hungarian) ile eşleyip stabil bağ (bin\_cur) üretir.

Hazırlayan: GitHub Copilot