

Robustezza del MASA Tracker e Pipeline Euristica

Gianluca Giuliani

Laboratorio Image and Video Analysis

Agosto 2025

Obiettivi del Progetto

- Valutare la **robustezza** del MASA tracker
- Correggere errori di **ID assignment** (occlusioni, rientri)
- Sfruttare le **deep features** per supportare la riassegnazione

Script featureMeanStd.py (1/2)

Funzione

- Caricamento modello MASA
- Hook sul **neck layer** per catturare feature intermedie

Motivazione

- Neck = compromesso tra dettaglio spaziale e semantico

Script featureMeanStd.py (2/2)

Pipeline frame-by-frame

- 1 Rilevamento e tracking con MASA
- 2 Estrazione feature dal neck
- 3 ROI Align \rightarrow feature 7×7
- 4 Salvataggio in .npy con *frame ID*, *track ID*, *bbox*

Output: database di feature per ogni detection

Script heuristic.py (1/3)

Goal: correggere ID assignment con euristiche

- Modello di apparenza incrementale (media + varianza)
- PCA per riduzione dimensionale
- Distanza euclidea per valutare somiglianza

Script heuristic.py (2/3)

Gestione conflitti

- **Pause period:** evita riassegnazioni instabili
- **Skip IDs:** esclusione permanente
- **Alias management:** unione ID frammentati
- **User interaction:** scelta manuale nei casi ambigui

Script heuristic.py (3/3)

Statistiche

- Top1/Top2/Top3: qualità suggerimenti
- IDs saltati
- Auto-associazione basata su consistenza temporale

Limitazioni

- PCA statica (no aggiornamenti online)
- Necessità input umano nei casi ambigui
- No uso di Mahalanobis

Modello di Apparenza Incrementale

Obiettivo: rappresentare ogni identità come distribuzione delle sue feature nel tempo.

Aggiornamento online:

$$\mu_{t+1} = (1 - \alpha) \mu_t + \alpha x$$

$$\Sigma_{t+1} = (1 - \alpha) \Sigma_t + \alpha (x - \mu_{t+1})(x - \mu_{t+1})^T$$

- μ_t : media delle feature (rappresentazione compatta dell'identità)
- Σ_t : stima aggiornata delle deviazioni (non vera covarianza, ma approssimazione pesata)
- α : learning rate (peso della nuova osservazione)
- Similarità calcolata tramite **distanza euclidea** tra x e μ_t

Risultati

- Migliore coerenza nelle associazioni ID
- Associazioni più stabili con occlusioni/rientri
- Semi-automatico utile anche per valutare la robustezza del modello.

Flusso di Esecuzione (1/2)

Flusso con `heuristic.py`

- 1 Caricamento delle feature
- 2 Nuovo ID → confronto con modelli esistenti
- 3 Sistema propone **Top-k candidati**
- 4 Utente: associare / dichiarare nuovo ID / skip
- 5 In caso di conflitto: richiesta decisione

Gestione dei Conflitti

Quando avviene un conflitto?

- Due ID diversi corrispondono alla stessa identità
- Un ID ritorna ma la sua identità è già presente nella stessa frame

Strategia adottata:

- Pausa automatica degli aggiornamenti
- Richiesta di intervento all'utente
- Applicazione di regole euristiche (skip, alias, top-k candidati)

Esempio di Conflitto

Scenario:

- Frame x : ID 26 \rightarrow Identity 1
- Frame $x + 1$: ID 26 \rightarrow Identity 3 (*drift temporaneo*)
- Nello stesso frame appare anche ID 35 \rightarrow Identity 1

Problema: due alias della stessa identità (ID 26 e ID 35) non possono coesistere.

Soluzione:

- Sistema blocca aggiornamenti
- Mostra **Top-k candidati**
- Utente decide:
 - Re-assegnare ad identità corretta
 - Skipare ID

Workflow di Risoluzione del Conflitto

- 1 Rilevato ID che entra in conflitto
- 2 Controllo se alias già attivi nella stessa frame
- 3 Se conflitto \rightarrow stop aggiornamenti
- 4 Presentazione Top-k candidati
- 5 Decisione utente:
 - Associare \rightarrow aggiornamento modello
 - Skippare \rightarrow ID escluso
- 6 Pausa di $pause_sec \times fps$ frame per stabilizzare

Meccanismi di Stabilizzazione

1. `pause_sec`

- Converte in frame: $pause_frames = pause_sec \times fps$
- Dopo un'associazione, lo stesso ID non viene rivalutato per $pause_frames$ frame
- Evita oscillazioni e riassegnazioni instabili

2. `lock_after`

- Se lo stesso candidato è Top-1 per N frame consecutivi ($N = lock_after$)
- \Rightarrow Associazione viene **bloccata automaticamente**

Effetto combinato:

- $pause_sec = \text{raffreddamento temporale}$ (meno riconsiderazioni)
- $lock_after = \text{consolidamento automatico}$ (meno input manuale)

Risultati

Statistiche:

Top1: 11/14 (78.6%)

Top2: 12/14 (85.7%)

Top3: 12/14 (85.7%)

Performance on second video

Statistiche:

Top1: 20/21 (95.2%)

Top2: 20/21 (95.2%)

Top3: 20/21 (95.2%)

Performances on third video

Conclusioni e Sviluppi Futuri

- Riduzione errori di tracking con pipeline semi-automatica
- Futuri miglioramenti:
 - PCA adattiva online
 - Uso di metriche avanzate (Mahalanobis)
 - Feature da altri layer del modello