

# Monitoraggio di aree estese tramite algoritmi di computer vision.

Lorenzo Seidenari, Francesco Turchini, Tiberio Uricchio and Alberto Del Bimbo

Università degli Studi di Firenze

nome.cognome@unifi.it

## Abstract

In questo lavoro proponiamo un sistema di tracciamento visuale combinato con un algoritmo di rilevamento di anomalie in grado di estrarre automaticamente informazioni rilevanti per la protezione delle aree osservate. Si riporta infine un'analisi quantitativa delle prestazioni.

## 1 Introduzione

Il monitoraggio di aree urbane ampie critiche è di primaria importanza. E' possibile utilizzare algoritmi di *computer vision* recenti, sfruttando un numero limitato di telecamere al fine di evitare il dispiegamento di un numero elevatissimo di sensori. In questo lavoro proponiamo un sistema di tracciamento visuale combinato con un algoritmo di rilevamento di anomalie in grado di estrarre automaticamente informazioni rilevanti per la protezione delle aree osservate. Viene riportata un'analisi quantitativa degli algoritmi su dataset pubblici e proprietari.

Il sistema proposto ha due funzionalità principali: 1) la rilevazione di pattern inattesi su base statistica, descritto in Sez. 2; 2) l'analisi dei pattern di movimento degli oggetti di interesse rilevati, descritto in Sez. 3;

## 2 Anomaly detection

Per rilevare anomalie proponiamo un *ensemble* di politopi [Casale *et al.*, 2014]. Questo metodo è in grado di approssimare lo spazio contenente le rappresentazioni locali dei frame usando un insieme di politopi. Il sistema si può considerare semi-supervisionato in quanto viene addestrato solo con esempi provenienti da video contenenti comportamenti usuali. Questa tecnica consente l'addestramento senza annotazione umana in quanto una volta che il sistema è installato, i frame regolari possono essere collezionati automaticamente in un periodo di tempo molto breve. Inoltre, acquisendo continuamente nuovi descrittori e stimando nuovamente i politopi è possibile aggiornare il sistema. Questa tecnica non consente il trasferimento di un modello da una scena sorvegliata ad un'altra, tuttavia l'addestramento del sistema è possibile banalmente acquisendo nuovi descrittori.

## Modello

Dato un insieme di punti  $\mathbf{X} = \{x_1, \dots, x_m\}$ . Il *convex hull* di  $\mathbf{X}$  è definito come:

$$C(\mathbf{X}) = \left\{ \sum_{i=1}^{|\mathbf{X}|} \gamma_i x_i \mid x_i \in \mathbf{X}; \sum_i \gamma_i = 1, \gamma_i \geq 0 \forall i \right\} \quad (1)$$

Un esempio non osservato può essere ritenuto anomalo se esterno al *convex hull*. Al fine di poter variare la sensibilità del sistema si muovono i vertici dell'*hull* rispetto al centroide nel seguente modo. In particolare dato il centroide  $c_i$  e i suoi vertici  $\mathbf{V} \subset \mathbf{X}$ , calcoliamo il nuovo *hull* con:

$$\mathbf{V}_\alpha = \left\{ v + \alpha \frac{(v - c_i)}{\|v - c_i\|}, v \in \mathbf{V} \right\} \quad (2)$$

## Ensemble di politopi

Per rendere più robusto il sistema si considerano  $T$  *hull* convessi, ciascuno addestrato su una feature estratta tramite Improved Dense Trajectories [Wang *et al.*, 2016]. Se consideriamo uno spazio finale di dimensione  $d$  e la dimensione  $D_f$  di ciascuna feature  $f$ . Per ogni feature e per ogni proiezione casuale  $i$  si genera una matrice di proiezione casuale  $R_i^f$ . La proiezione si ottiene tramite:

$$\mathbf{X}_{R_i^f} = \{R_i^f x, \forall x \in \mathbf{X}\} \quad (3)$$

Ogni traiettoria è considerata anomala se i relativi descrittori sono al di fuori di ciascun politopo.

$$x_f \notin C^f(\mathbf{X}_{P_i^f}) \forall f, i \quad (4)$$

Si applica k-means alle traiettorie marcate come anomale e si usano le celle di voronoi con un numero di anomalie superiori ad una data soglia per la localizzazione spaziale.

## 3 Tracciamento e Conteggio

Il sistema progettato è in grado di monitorare un'area con decine di posti auto senza la necessità di barriere, rilevando sia l'occupazione degli stalli che il posteggio dei veicoli fuori posto come mostrato in Fig. 1. Il nostro metodo sfrutta YOLOv2 [Redmon e Farhadi, 2016], un rilevatore di oggetti a singolo stadio, ottimale dal punto di vista del trade-off accuratezza/velocità. L'algoritmo 1 descrive la fase di associazione *greedy* tra detection e tracce esistenti. Un filtro di Kalman

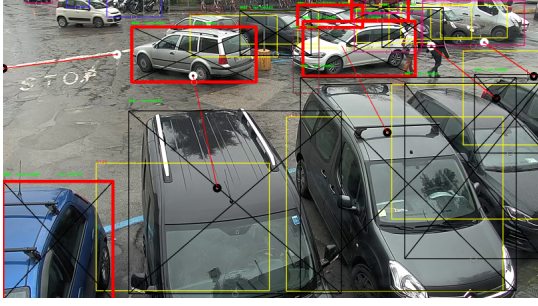


Figura 1: Rilevamenti di veicoli in Piazza Tasso a Firenze. I rettangoli gialli indicano gli stalli di parcheggio, i rettangoli neri marchiati da una croce indicano le auto parcheggiate correttamente, infine i rettangoli rossi indicano auto posizionate irregolarmente.

### Algorithm 1 Tracking by detection

FUNCTION associate ( $\mathcal{T}_{t-1}, \mathcal{D}_t$ )

Input  $\mathcal{T}_{t-1} : \{t_1 \dots t_n\}, \mathcal{D}_t : \{d_1 \dots d_m\}, \mathbf{M}_{ij} = \frac{d_i \cap t_j}{d_i \cup t_j}$

Output  $\mathcal{T}_t$

```

1: while  $\max_{ij} \mathbf{M}_{ij} > \tau$  do
2:   if not  $\mathbf{A}_{ij} \wedge \mathbf{M}_{ij} > \tau$  then
3:      $\langle \hat{i}, \hat{j} \rangle \leftarrow \arg \max_{ij} \mathbf{M}_{ij}$ 
4:      $t_{\hat{i}} \leftarrow d_{\hat{j}}$ 
5:      $\mathbf{A}_{\hat{i}} \leftarrow \text{TRUE}$ 
6:      $\mathbf{A}_{\hat{j}} \leftarrow \text{TRUE}$ 
7:   end if
8: end while
9: Unassigned detections initialize new tracks.
 $\mathcal{T}_t \leftarrow \mathcal{T}_{t-1} \cup \{d \mid \mathbf{A}_{ij} = \text{TRUE}\}$ 
10: Remove tracks not assigned for  $\omega$  frames.
 $\mathcal{T}_t \leftarrow \mathcal{T}_{t-1} \setminus \{t_i \mid l_i > \omega\}$ 

```

è mantenuto attivo per ogni traccia. Al termine dell'associazione, per le tracce associate si aggiorna lo stato del filtro di Kalman con la nuova misura (detection), per le altre si procede all'aggiornamento dello stato in accordo con un modello di moto a velocità costante.

## 4 Risultati

### 4.1 Rilevamento Anomalie

L'algoritmo di rilevamento anomalie è stato testato su UCSPed1 e UCSPed2, ottenendo risultati allo stato dell'arte come mostrato in Tabella 1.

| Method                                | Ped1        |             | Ped2        |             |
|---------------------------------------|-------------|-------------|-------------|-------------|
|                                       | Frame       | Pixel       | Frame       | Pixel       |
| <b>Ours</b>                           | 78.1        | <b>62.2</b> | 80.7        | <b>75.7</b> |
| Xu et al. [Xu et al., 2015]           | 78.0        | 59.9        | <b>83.0</b> | -           |
| MDT Spatial [Li et al., 2014]         | 56.2        | 54.2        | 71.3        | 63.4        |
| MDT Temporal [Li et al., 2014]        | 77.1        | 48.2        | 72.1        | 56.8        |
| 150 fps [Lu et al., 2013]             | <b>85.0</b> | 59.1        | -           | -           |
| Bertini et al. [Bertini et al., 2012] | 66.0        | 29.0        | 68.0        | -           |

Tabella 1: Comparazione in termini di Rate of Detection su Ped1 and Ped2, sia a livello frame che a pixel.

### 4.2 Analisi Parcheggio e Conteggi

Per quanto riguarda il parcheggio abbiamo analizzato più di 8h di sequenze acquisite all'interno di un parcheggio in Piazza Tasso a Firenze. Il posizionamento della telecamera, vincolato da parte dell'amministrazione, crea un caso d'uso molto complesso con molteplici e ampie occlusioni. Nonostante ciò il sistema ha un'ottima accuratezza nel rilevare l'occupazione dei posteggi e rileva quasi l'80% delle infrazioni di parcheggio.

| Accuratezza Posti | Recall Anomalie | Precision Anomalie |
|-------------------|-----------------|--------------------|
| 93.88%            | 79.27%          | 60.75%             |

Tabella 2: Performance del nostro sistema in condizioni complesse in un parcheggio reale.

Il conteggio di veicoli è stato testato su 3 sequenze acquisite all'interno di un porto allo sbarco di un traghetto. Come si può notare dalla Tabella 3 il sistema ha una buona accuratezza, considerato che le sequenze hanno un elevato tasso di occlusione.

| Classe  | Conteggio | GT | Classe  | Conteggio | GT | Classe  | Conteggio | GT |
|---------|-----------|----|---------|-----------|----|---------|-----------|----|
| Persona | 29        | 36 | Persona | 79        | 83 | Persona | 24        | 20 |
| Auto    | -         | -  | Auto    | 22        | 21 | Auto    | 1         | 1  |
| Furgone | -         | -  | Furgone | 6         | 5  | Furgone | 10        | 10 |
| Moto    | 0         | 6  | Moto    | -         | -  | Moto    | 1         | 6  |

Tabella 3: Conteggio di veicoli

### Riferimenti bibliografici

- [Bertini et al., 2012] Marco Bertini, Alberto Del Bimbo, e Lorenzo Seidenari. Multi-scale and real-time non-parametric approach for anomaly detection and localization. *Computer Vision and Image Understanding*, 116(3):320–329, 2012.
- [Casale et al., 2014] Pierluigi Casale, Oriol Pujol, e Petia Radeva. Approximate polytope ensemble for one-class classification. *Pattern Recognition*, 47(2):854–864, 2014.
- [Li et al., 2014] Weixin Li, Vijay Mahadevan, e Nuno Vasconcelos. Anomaly detection and localization in crowded scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(1):18–32, 2014.
- [Lu et al., 2013] Cewu Lu, Jianping Shi, e Jiaya Jia. Abnormal event detection at 150 fps in matlab. In *Proc. of International Conference on Computer Vision (ICCV)*, pages 2720–2727, 2013.
- [Redmon e Farhadi, 2016] Joseph Redmon e Ali Farhadi. Yolo9000: better, faster, stronger. *CoRR*, abs/1612.08242, 2016.
- [Wang et al., 2016] Heng Wang, Dan Oneata, Jakob Verbeek, e Cordelia Schmid. A robust and efficient video representation for action recognition. *International Journal of Computer Vision*, 119(3):219–238, 2016.
- [Xu et al., 2015] Dan Xu, Elisa Ricci, Yan Yan, Jingkuan Song, e Nicu Sebe. Learning deep representations of appearance and motion for anomalous event detection. *Computer Vision and Image Understanding*, 156:117–127, 2015.