Contents lists available at ScienceDirect

Computer Vision and Image Understanding

journal homepage: www.elsevier.com/locate/cviu

Distributed multi-target tracking and active perception with mobile camera networks[☆]

Sara Casao [a,*,1], Álvaro Serra-Gómez [b,1], Ana C. Murillo [a], Wendelin Böhmer [b], Javier Alonso-Mora [b], Eduardo Montijano [a]

[a] DIIS - I3A, Universidad de Zaragoza, Spain
[b] Cognitive Robotics at TU Delft, The Netherlands

ARTICLE INFO

Dataset link: https://sites.google.com/unizar.es/poc-team/research/hlunderstanding/collaborativecameras

MSC:
41A05
41A10
65D05
65D17

Keywords:
Multi-camera scene analysis
Collaborative and autonomous decision making

ABSTRACT

Smart cameras are an essential component in surveillance and monitoring applications, and they have been typically deployed in networks of fixed camera locations. The addition of mobile cameras, mounted on robots, can overcome some of the limitations of static networks such as blind spots or back-lightning, allowing the system to gather the best information at each time by active positioning. This work presents a hybrid camera system, with static and mobile cameras, where all the cameras collaborate to observe people moving freely in the environment and efficiently visualize certain attributes from each person. Our solution combines a multi-camera distributed tracking system, to localize with precision all the people, with a control scheme that moves the mobile cameras to the best viewpoints for a specific classification task. The main contribution of this paper is a novel framework that exploits the synergies that result from the cooperation of the tracking and the control modules, obtaining a system closer to the real-world application and capable of high-level scene understanding. The static camera network provides global awareness of the control scheme to move the robots. In exchange, the mobile cameras onboard the robots provide enhanced information about the people on the scene. We perform a thorough analysis of the people monitoring application performance under different conditions thanks to the use of a photo-realistic simulation environment. Our experiments demonstrate the benefits of collaborative mobile cameras with respect to static or individual camera setups.

1. Introduction

Multi-camera systems are common in applications such as surveillance or monitoring. The use of multiple cameras increases the coverage and the amount of information collected from large-scale scenes. Although the most frequent configuration in surveillance applications is a network of static cameras, including mobile cameras brings plenty of potential benefits. In addition to the improved coverage capabilities of such a hybrid system, mobile cameras can be guided to acquire more detailed information and particular viewpoints when needed. Enhancing collaborative behavior among them is then essential to achieve an efficient mutual scene understanding (Mekonnen et al., 2013; Li et al., 2018; Miller et al., 2022).

One of the main challenges of collaborative camera network systems is to attain robustness and efficiency. Hence, there has been a tendency to transition from centralized to distributed setups that can easily scale and are more robust against individual node failures (Zhou et al., 2022; Yu et al., 2022). Another common challenge in multi-camera systems is finding a suitable viewpoint that maximizes gaining new knowledge for a given recognition task. For instance, solving tasks such as person identification or clothing brand recognition requires a specific viewpoint, which should be free from occlusion or blind spots. Active perception enables the capability of moving a camera to the location of the most informative perspective. Developing and evaluating distributed solutions, where mobile cameras with autonomous decision-making are involved, is not a trivial task. To address all of these challenges we propose a novel active and distributed framework. Our system has static cameras to monitor the scene and mobile cameras to strengthen the visualization of certain attributes with high-resolution close-up target images, as summarized in Fig. 1.

The mobile cameras, drones in our case, are guided by a control policy built upon previous work (Serra-Gómez et al., 2023). This policy continuously determines the cameras' next position and orientation

[☆] This work was supported by DGA project T45_23R, by MCIN/AEI/ERDF/European Union NextGenerationEU/PRTR project PID2021-125514NB-I00, and the Office of Naval Research Global project ONRG-NICOP-N62909-19-1-2027.

[*] Corresponding author.
E-mail address: scasao@unizar.es (S. Casao).

[1] Sara Casao and Álvaro Serra-Gómez contributed equally in this work.

https://doi.org/10.1016/j.cviu.2023.103876
Received 30 May 2023; Received in revised form 11 October 2023; Accepted 24 October 2023
Available online 2 November 2023

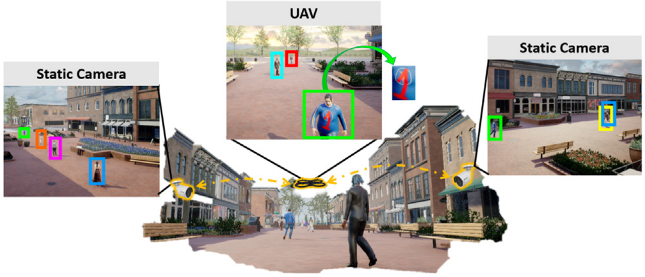

Fig. 1. Overview of our multi-camera collaborative system. The system comprises a camera network that performs a distributed multi-target tracking process. The static cameras monitor the scene and the mobile cameras are guided by a control policy to capture close-up images of viewpoints likely to strengthen the classification of certain attributes.

to capture viewpoints that maximize the acquisition of relevant information for certain people's attributes class. Differently from our prior work, here we consider multiple drones working together with a network of static cameras that provide information about the targets' position and orientation using real data, taking into account the challenges associated to the use of a real tracking system.

The distributed tracking process in charge of this task is based on Casao et al. (2021). Our contribution in this module is related to the implementation, making the transition to a real system easier thanks to the integration with ROS to handle communications. The assessment of the framework is performed with a photo-realistic simulator. In particular, we use the open-source Unreal Engine together with the AirSim simulator (Shah et al., 2018), which provide a photo-realistic environment to simulate drones and static camera data generation. Additionally, we employ specific tools for creating scenes involving multiple pedestrians from Casao et al. (2023).[2]

To summarize, the main contributions of this work are:

- A novel hybrid multi-camera framework, composed of static and mobile nodes, that collaboratively tackles the problem of people monitoring. To do so, it combines distributed tracking and active perception of semantic knowledge from the scene.
- *Active Perception:* We extend prior work to consider multiple mobile cameras and real perception provided by the distributed tracking algorithm.
- *Distributed Tracking:* We incorporate distributed communications using ROS and perform the evaluation with a photo-realistic simulator, contributing to bridge the gap with real-world applications.

2. Related work

2.1. Multi-camera multi-target tracking

Multi-camera centralized setups are commonly used in real-world applications to cover larger areas (Guo et al., 2022; Quach et al., 2021) or acquire a greater amount of information (Byeon et al., 2018; Zhang et al., 2020). These centralized approaches process the entire camera network information in one unique node, making it difficult to scale up. Thus, there is a trend toward distributed setups to increase the applicability of multi-camera systems (Xompero and Cavallaro, 2022). While theoretical works have proposed solutions to problems such as event-trigger mechanisms for bandwidth requirements (Ge et al., 2019) or consensus algorithms to unify local estimations (Soto et al., 2009; Li et al., 2023), only a few works have addressed the distributed multi-target tracking with real data. For example, Kamal et al. (2015) combine the Information-weighted Consensus Filter (ICF) with the Joint Probabilistic Data Association Filter (JPDAF), which uses the previous target states, to fill the gap of relating measurements and trackers in the consensus algorithm. Based on the same ICF consensus method, He et al. (2019) address the association of measurements and trackers through a global metric that merges appearance and geometry cues. To associate trackers across cameras, they employ the Euclidean distance between the 3D position of the targets. Different from Kamal et al. (2015) and He et al. (2019), we tackle the problem of having mobile nodes in the camera network. Besides, we analyze in both data associations, trackers with measurements and cross-camera trackers, the geometric information together with the appearance representation.

2.2. Collaborative systems for perception tasks

Multiple works have developed collaborative systems to address complex perception tasks. One of the most common problems tackled is active object tracking, where visual observations are transformed into a camera control signal to improve the tracking process, e.g., turning left or moving forward (Schranz and Andre, 2018). The combination of a fixed camera, that globally monitors the scene, with a pan–tilt-zoom (PTZ) camera, used to increase the image quality of the target of interest, is proposed in Li et al. (2018) In Li et al. (2020), this setup is extended to a centralized PTZ camera network, where reinforcement learning techniques are employed to learn the new pose of the cameras for finding the target and tracking it as long as possible. In order to follow an object capable of moving in all directions, Trujillo et al. (2019) develop a cooperative aerial robotic approach with two drones for achieving overlapping images and forming a pseudo-stereo vision system. The collaboration of hybrid systems has been studied for different tasks such as dynamic obstacle avoidance, where the information of the static cameras is leveraged by the mobile robot (Mekonnen et al., 2013), or the localization, planning, and navigation of ground robots using a semantic map created by a high-altitude quadrotor (Miller et al., 2022). Furthermore, some works have focused on distributed collaborative perception tasks. Yu et al. (2022) propose an approach for distributed learning where each robot only shares the weights of the network for privacy protection and Zhou et al. (2022) present a general-purpose graph neural network for fusing node information and obtaining accurate perception tasks. Closer to our work, Bisagno et al. (2018) leverage the collaboration of fixed cameras, PTZ, and UAVs for crowd scene covering in a distributed manner. Different from Bisagno et al. (2018), we do not assume as known the target positions, which entail addressing the challenges of a distributed tracking system.

2.3. Active perception for class recognition

The active perception problem of recognizing certain classes is commonly addressed by defining a set of viewpoints in advance, which are then used to plan trajectories for gathering new information. One-step greedy planners select viewpoints specific to objects based on factors such as class uncertainty and observation occlusions (Patten et al., 2016). Instead, non-myopic methods such as Popović et al. (2017) consider both, movement costs and information gained between the object's viewpoints. Alternatively, some approaches formulate the problem as a partially observable Markov Decision Process (POMDP) and design paths over viewpoints by accounting for costs associated with measurements, occlusions, and potential misclassifications (Atanasov et al., 2014). Likewise, Patten et al. (2018) employs a modified version of Monte-Carlo tree search to generate plans. However, these techniques typically rely on a priori access to the black-box model for estimating the usefulness of viewpoints. More recent works use non-myopic learning methods like Deep Reinforcement Learning (DRL) for static multi-target pose estimation and active perception. They optimize camera movements to reduce observation uncertainty (Sock et al., 2020) or maximize information gain (Xu et al., 2021). However, these

[2] Simulated data and photo-realistic environment used available at https://sites.google.com/unizar.es/poc-team/research/hlunderstanding/collaborativecameras.

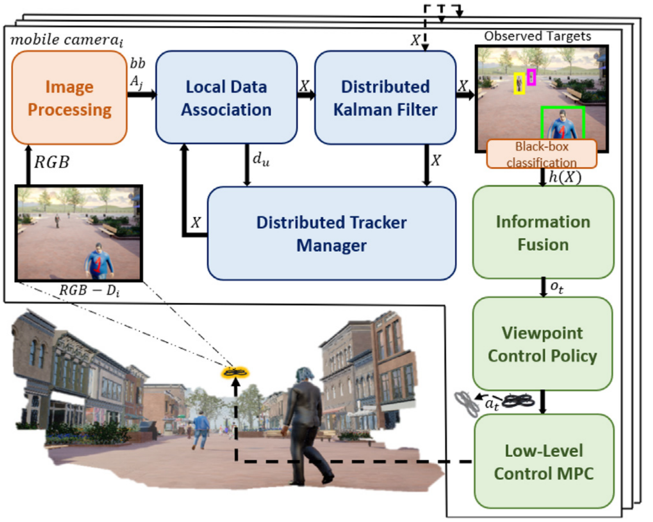

Fig. 2. Method overview deployed in one mobile camera. The whole system is implemented in ROS, initializing each camera as a node and the image processing module as a service. First, the Local Data Association relates people detection (bb) with the corresponding trackers ($\mathcal{X}$). Then, the cameras exchange and fuse data with their neighboring cameras to obtain a collaborative distributed tracking system. The knowledge of the environment is provided to the control policy for obtaining a new recommendation of viewpoint (a_t) to improve the gathering people's information.

approaches either assume static targets, are limited to closed environments (Kent and Chernova, 2020), or require prior knowledge of where the information is visible from Alcántara et al. (2021) and Jeon et al. (2020). Our work leverages an attention-based neural network architecture to encode dynamic targets and to provide viewpoint recommendations that are traced with a low-level controller. In addition, we enable the use of multiple drones and overcome the assumption of possessing prior knowledge about the positions and orientations of the targets by exploiting the collaboration with a multi-target tracking system.

3. Preliminaries

3.1. Problem formulation

This work addresses the distributed tracking and correct visualization of people's attributes in large-scale environments. We monitor an area populated by a set of I targets, $\{\mathcal{X}_i\}_{i=1}^{I}$, with a system of J cameras, $\{C_j\}_{j=1}^{J}$, where a subset of $Q < J$ cameras can translate and rotate, e.g. they are installed on drones. Each camera in the network captures an RGB image and a depth map to estimate the state of the targets locally by fusing its information with that received from its neighbors, $\mathcal{N}_j$. The state of target i in camera j is defined as $\mathbf{x}_i^j = (x_i^j, y_i^j, z_i^j, w_i^j, h_i^j, \dot{x}_i^j, \dot{y}_i^j)$ represented by a 3D cylinder with (x_i^j, y_i^j, z_i^j) the 3D coordinates of the center cylinder's base, w_i^j the width, h_i^j the height, and $(\dot{x}_i^j, \dot{y}_i^j)$ the velocity of the target in the x and y directions, respectively. The orientation of the target, φ_i^j, is estimated based on their velocities $\dot{x}_i^j$ and $\dot{y}_i^j$. The responsibility for correctly visualizing the attribute's class of the targets lies in the moving cameras (drones). It is important to note that these attributes can only be observed from specific viewpoints, such as determining if the targets are wearing a backpack or glasses. The state of the drones $\mathbf{y}_q = (\mathbf{u}_q, \psi_q)$, assumed as known in this work, is represented as their position $\mathbf{u}_q$ and their heading ψ_q, being $q \in \{1, \dots, Q\}$. Each drone is controlled by a hierarchical policy, where a viewpoint control policy operating at $\frac{1}{\tau_h}$ Hz takes as input the knowledge of the scene and outputs a viewpoint recommendation $\mathbf{a}^q$. Next, the recommended viewpoint is traced with a low-level controller operating at $\frac{1}{\tau_l} \gg \frac{1}{\tau_h}$ Hz. The purpose of the policy is to position the targets' attributes within the field of view (FOV) of the drone. We assume that the drones are faster than the targets and fly at a constant height above them, avoiding collisions.

The goal of the presented work is to achieve an accurate estimation of the targets' position and visualize all people's attributes as quickly as possible.

3.2. Overview

Fig. 2 presents an overview of the proposed method to address the problem described in the previous section. The complete framework has been implemented in ROS, with each camera defined as a node of the system and ensuring synchronization between them. Neural networks have been implemented in the image processing module as services to save memory.

First, each camera captures an RGB image and a depth map (D_i) to compute the re-projection between the image plane and the real-world coordinates. We incorporate depth information to simplify the re-projection but this could be replaced by a network calibration in a more realistic setup. Then, a general detector provides the people bounding boxes (bb) that are used as measures for the tracking system and that are associated with the current trackers through the Local Data Association module (LDA). Once the cameras in the networks exchange the targets' information ($\mathcal{X}$) with their neighbors, the Distributed Kalman Filter (DKF) implemented attempts to obtain consensus on the targets' state. Finally, the Distributed Tracker Manager (DTM) initializes new trackers and associates them locally with the trackers received from the neighboring cameras. The mobile cameras of the system obtain the output of a black-box CNN, with perception information about the visible targets ($h(\mathcal{X})$), and update their class beliefs with an efficient information fusion method. Based on the latter and the estimated state of the targets, the viewpoint control policy recommends a new camera pose (a_t) to maximize the information acquired in the next step. The new viewpoints are then tracked with a low-level controller.

4. Distributed tracking

This section explains the different components of our approach to perform fully distributed multi-target tracking with hybrid collaborative cameras.

4.1. Distributed Kalman filter

We define the target motion model as a discrete-linear dynamic system with constant velocity. Each camera executes a Kalman filter independently producing a local estimation of the target state, $\hat{\mathbf{x}}_i(k)$, and the associated error covariance matrix $\mathbf{P}_i(k)$. Note that local estimations may vary among different cameras. Therefore, the Distributed Kalman-Consensus filter (Soto et al., 2009) is implemented to mitigate these differences and seek to reach a consensus in $\hat{\mathbf{x}}_i(k)$ for all cameras C_j.

The consensus algorithm assumes knowledge of the data association between the local measurement $\mathbf{z}_i(k)$ and the target prediction $\bar{\mathbf{x}}_i(k)$, which is obtained by applying the linear motion model to the previous target state estimation $\hat{\mathbf{x}}_i(k-1)$. The measurement $\mathbf{z}_i(k)$ is the 3D cylinder obtained as the projection of the bounding box given by the detector, and the velocity of the target computed with the last data association, i.e., $\mathbf{z}_i(k) = (x(k), y(k), z(k), w(k), h(k), \dot{x}(k), \dot{y}(k))$. This measurement is coupled in the filter with a zero mean Gaussian noise characterized with $\mathbf{R}_i(k)$ as its covariance matrix. Using mobile cameras requires online updates of the transformation matrix from the image plane to the three spatial global coordinates of the world. The cameras of the photo-realistic environment follow the pinhole model, which combined with the depth information, $d(k)$, enables the conversion of image plane coordinates $v_x(k)$ and $v_y(k)$, to the relative 3D world camera coordinates $x_r(k), y_r(k)$ and $z_r(k)$ by

$$x_r(k) = d(k), y_r(k) = \frac{d(k)}{f}(v_x(k) - c_x), z_r(k) = \frac{d(k)}{f}(v_y(k) - c_y) \quad (1)$$

where f is the focal length and, c_x and c_y are the image center coordinates in x and y, respectively. Then, the relative camera coordinates are transformed into the common global world system demand by the consensus-filter algorithm following

$$\begin{bmatrix} x(k) \\ y(k) \\ z(k) \\ 1 \end{bmatrix} = \begin{bmatrix} R_j(k) & | & T_j(k) \\ 0 & & 1 \end{bmatrix} \begin{bmatrix} x_r(k) \\ y_r(k) \\ z_r(k) \\ 1 \end{bmatrix} \quad (2)$$

being $R_j(k)$ the rotation matrix and $T_j(k)$ the translation vector of the camera at instant k, assumed as known. In a real setup, this information could be computed by offline calibration of the cameras and using onboard sensors such as GPS or IMUs together with SLAM algorithms for the drones. Regarding the velocity, we take advantage of the online tracking to measure the time the target has taken to arrive at the current position at k since the last data association between $\bar{\mathbf{x}}_i$ and $\mathbf{z}_i$.

Once the camera j associates the local measurement $\mathbf{z}_i^j(k)$ with the local target prediction $\bar{\mathbf{x}}_i^j(k)$, the consensus algorithm transforms the measurement and its noise to the information form by

$$\mathbf{u}_i^j(k) = \mathbf{H}^T \mathbf{R}_i^{-1}(k)\mathbf{z}_i^j(k), \quad \mathbf{U}_i^j(k) = \mathbf{H}^T \left(\mathbf{R}_i^j(k)\right)^{-1} \mathbf{H}. \quad (3)$$

The obtained sensor data information, $\mathbf{u}_i^j(k)$, and its inverse-covariance matrix, $\mathbf{U}_i^j(k)$ are exchanged with the neighboring cameras in the network $\mathcal{N}_j$, together with $\bar{\mathbf{x}}_i(k)$. Due to the transformation into the information form, we are able to combine all the measurements received from other cameras with the acquired one by simply adding them,

$$\mathbf{y}_i^j(k) = \sum_{C\in\mathcal{N}_j} \mathbf{u}_i^C(k), \qquad \mathbf{S}_i^j(k) = \sum_{C\in\mathcal{N}_j} \mathbf{U}_i^C(k). \quad (4)$$

Finally, the estimated state is updated by correcting the prediction target state with the data computed in (4) and the predictions from the neighboring cameras following

$$\begin{aligned} \hat{\mathbf{x}}_i^j(k) = {} & \bar{\mathbf{x}}_i^j(k) + \mathbf{M}_i^j(k)\left[\mathbf{y}_i^j(k) - \mathbf{S}_i^j(k)\bar{\mathbf{x}}_i^j(k)\right] \\ & + \gamma \mathbf{M}_i^j(k) \sum_{C\in\mathcal{N}_j} (\bar{\mathbf{x}}_i^C(k) - \bar{\mathbf{x}}_i^j(k)), \end{aligned} \quad (5)$$

where $\mathbf{M}_i^j(k) = (\mathbf{P}_i^j(k)^{-1} + \mathbf{S}_i^j(k))^{-1}$ is the Kalman Gain in the information form and $\gamma = 1/\|\mathbf{M}_i^j(k) + 1\|$.

4.2. Local data association

For simplicity in the explanation, this subsection will focus on the data association in a single camera. Hence, the subscripts j used in the notation will refer to the different measurements locally observed and not the cameras in the network. The accurate update of the DKF relies on a correct association between the set of measurements, $\mathcal{Z} = \{\mathbf{z}_j\}$, and the set of targets prediction, $\bar{\mathcal{X}} = \{\bar{\mathbf{x}}_i\}$, during each estimation cycle. To tackle this issue, we assess two constraints based on geometry and appearance.

The similarity value in the geometry of both sets is obtained as

$$s_d(\mathbf{z}_j, \bar{\mathbf{x}}_i) = \begin{cases} \frac{1}{\alpha}\, d_M(\mathbf{z}_j, \bar{\mathbf{x}}_i) & \text{if} \quad d_M(\mathbf{z}_j, \bar{\mathbf{x}}_i) < \tau_d \\ 1 & \text{otherwise,} \end{cases} \quad (6)$$

being α a configuration parameter, τ_d a threshold applied to ignore highly unlikely candidates and, $d_M(\mathbf{z}_j, \bar{\mathbf{x}}_i)$ the Mahalanobis distance between the x, y positions. The covariance matrix in the Mahalanobis distance is computed by adding the sub-matrices of $\mathbf{P}_i$ and $\mathbf{R}_j$ that encode the covariance position of the estimation $\bar{\mathbf{x}}_i$ and the measurement $\mathbf{z}_j$, respectively.

Then, those data whose distance is below τ_d are evaluated in appearance. To get representative appearance features for measuring similarity, we use the output of a person re-identification network (Zhou et al., 2021) pre-trained in the MSMT17 Benchmark (Wei et al., 2018) as appearance descriptors. Inspired by this re-identification methodologies, each local tracker creates an online appearance model, $\mathcal{F}_i$, of the target i with budget size. This appearance model, also called gallery, is built based on a scoring system that estimates the usefulness and confidence of each appearance feature. Thus, every feature of the gallery, $\mathbf{f}_i^\ell \in \mathcal{F}_i$, has a score assigned $\varepsilon(k)$ whose value changes depending on two factors. First, the gallery component with the minimum distance to the final associated measurement appearance increases its score by one with

$$\varepsilon_i^\ell(k+1) = \begin{cases} \varepsilon_i^\ell(k) + 1 & \text{if } \ell = \operatorname*{argmin}_{\mathbf{f}\in\mathcal{F}_i} \delta(\mathbf{f}_j, \mathbf{f}), \\ \varepsilon_i^\ell(k) & \text{otherwise,} \end{cases} \quad (7)$$

where $\mathbf{f}_j$ is the appearance feature get from the $\mathbf{z}_j$ bounding box detection. Secondly, the closest component of the appearance model to the gallery centroid, $\bar{\mathbf{f}}_i$, increases by one its value score while the farthest component decreases by one following

$$\varepsilon_i^\ell(k+1) = \begin{cases} \varepsilon_i^\ell(k) + 1 & \text{if } j = \operatorname*{argmin}_{\mathbf{f}\in\mathcal{F}_i} \delta(\bar{\mathbf{f}}_i, \mathbf{f}), \\ \varepsilon_i^\ell(k) - 1 & \text{if } j = \operatorname*{argmax}_{\mathbf{f}\in\mathcal{F}_i} \delta(\bar{\mathbf{f}}_i, \mathbf{f}), \\ \varepsilon_i^\ell(k) & \text{otherwise.} \end{cases} \quad (8)$$

The gallery is updated periodically every N iteration with a new feature. Once the budget size is reached, the component with the lowest score is dropped to make room for the newest one. Finally, the similarity between the appearance feature of the measurement, $\mathbf{f}_j$, and the tracker's gallery $\mathcal{F}_i$ used as a model of the appearance of the prediction state $\bar{\mathbf{x}}_i$ is provided by the minimum cosine distance

$$s_a(\mathbf{f}_j) = \min_{\mathbf{f}_i\in\mathcal{F}_i} \left(1 - \frac{\mathbf{f}_j^T\ \mathbf{f}_i}{\|\mathbf{f}_j\|\|\mathbf{f}_i\|}\right), \quad (9)$$

The final data association assignment between the measurements, $\mathcal{Z} = \{\mathbf{z}_j\}$, and the target predictions, $\bar{\mathcal{X}} = \{\bar{\mathbf{x}}_i\}$, is solved with the Hungarian algorithm (Kuhn, 1955) by defining the cost function as the product of both similarity scores, s_d and s_a.

4.3. Distributed tracker manager

In the practical implementation of distributed tracking systems, it is also essential to perform a correct association of trackers across the different cameras in the network. Our proposed approach to address this problem involves performing the same process as the one described in Section 4.2 for the local data association but with the set of measurements replaced by the set of other camera's predictions $\bar{\mathcal{X}}_j = \{\bar{\mathbf{x}}_j\}$, and using the Euclidean distance instead of the Mahalanobis distance. These modifications are based on the information exchanged in the communication message, which is subject to the data required in the DKF and does not include the covariance matrices $\mathbf{P}_j$. In case no local tracker is associated with those received from neighboring cameras, the current camera initializes a new tracker based on the tracker information received.

Since we limit sharing appearance exclusively to newly initialized trackers for saving bandwidth, the tracker consensus process across cameras occurs only when a new tracker is initialized in any of them. To ensure the robustness of mobile cameras in dynamic communication scenarios, where the cameras they exchange information with may change over time, we include the cross-camera trackers association in the communication message. This cross-camera trackers association consists of a look-up table where each tracker locally stores the unique identifier, i, assigned to the same target by the rest of the cameras in the network C_j. Consequently, once the message has traversed the entire network, the cameras achieve a global consensus on the association of trackers across all the cameras in the network.

5. Active perception

In addition to collaborating in the distributed tracking of multiple targets, mobile cameras tackle the task of active perception to gain additional knowledge about the people presented in the scene. They leverage shared information to efficiently position themselves for effectively visualizing each target's attribute class. In this work, mobile cameras are allowed to communicate between them in order to gather global knowledge of the visualization process's status.

5.1. *Target class observations and belief updates*

Every time step τ_h, the drone uses a black-box perception algorithm (e.g., a pre-trained CNN classifier) to compute the class probability distribution for each target visualized from the correct viewpoint. Let $\mathcal{P} = h(\mathcal{X}) = \{\mathbf{p}_i\}_{i=1}^{I}$ be the class probability distribution, where $\mathbf{p}_i$ represents the likelihood of target i belonging to each one of the G classes in the class set $\mathcal{G}$. To simplify the notation, in this complete Section 5, t will denote times periods of τ_h.

The probability distribution over time is modulated by belief vectors $\mathbf{b}_i^t$ for each target i. These vectors contain G belief values b_{ig}^t representing the aggregate likelihood of target i belonging to a class $g \in \mathcal{G}$ up to time t, i.e., combines the historical class probabilities distributions up to time t. The process of aggregating the drone's observations to derive class beliefs for each target is a crucial consideration. Standard Bayesian recursive estimation is not recommended in this case due to the unavailability of the measurement likelihood model, $\mathbb{P}(\mathbf{p}_i^t|\mathbf{b}^{t-1})$, from the black-box sensor. Building a precise pose-dependent likelihood model requires the construction of a dense dataset and considering all targets and occlusions for optimal viewpoint search. This process is expensive and does not scale well due to its computational demands.

Instead, we propose the use of the conflation operator $\zeta(\mathbf{p}_i^{1:t})$, a mathematical method introduced by Hill and Miller (2011). Conflation enables the aggregation of probability distributions obtained from measurements of the same phenomena under different conditions. It possesses the remarkable property of minimizing the loss of Shannon information when combining multiple independent probability distributions into a single distribution, specifically when computing $\mathbf{b}_i^t$ based on the measurements $\mathbf{p}_i^{0:t}$. The conflation is defined by

$$\mathbf{b}_i^t = \zeta(\mathbf{p}_i^{1:t}) \equiv \zeta(\mathbf{b}_i^{t-1}, \mathbf{p}_i^t) = \frac{\mathbf{b}_i^{t-1} \odot \mathbf{p}_i^t}{(\mathbf{b}_i^{t-1})^{\top} \mathbf{p}_i^t}, \tag{10}$$

where the Hadamard product $\odot$ in the numerator is taken componentwise, whereas the dot product is the normalization factor. Conflation's commutative and associative properties enable efficient recursive computation, making it suitable for onboard and decentralized belief updates in the presence of multiple communicating drones. The beliefs are initialized at $t = 0$ with a uniform prior probability distribution over all possible target classes, formally $b_{ig}^0 = 1/G \quad \forall g \in \mathcal{G}$.

5.2. *Viewpoint control policy*

The lack of an observation model that maps target relative poses to a probability distribution, i.e., the h function that maps $\mathcal{P} = h(\mathcal{X})$, hinders the direct solution of the active perception for class recognition problem. Therefore, we leverage Reinforcement Learning to train a viewpoint control policy, π_ϕ, that learns to recommend viewpoints $\mathbf{a}_t$ that minimize the accumulated entropy of all targets' beliefs over a given time horizon. The policy is parameterized by ϕ and operates at the perception low-frequency, $\frac{1}{\tau_h}$.

Each drone uses a copy of the same learned viewpoint control policy that solves the viewpoint recommendation problem. The viewpoint recommendation problem is formulated as a Partial Observable Markov Decision Process (POMDP), denoted by $\langle S, A, \mathcal{T}, \Omega, \mathcal{O}, R\rangle$. The state S includes the state of the drones, the targets' pose, their beliefs, and their visualization status (visualized or not). Actions A represent recommended viewpoints within a constrained neighborhood and transitions $\mathcal{T}$ assume timely movement to the next viewpoint. The drone receives partial information Ω about the environment through the observation function $\mathcal{O}$. The observation of each target is defined by $\mathbf{o}_{q,i}^t = [\bar{\mathbf{o}}_{q,i}^p, \bar{\mathbf{o}}_{q,i}^c] \in \Omega$ where $\bar{\mathbf{o}}_{q,i}^p$ is the observation of each target physical attributes (poses and velocities). Each target's attribute information is represented by $\bar{\mathbf{o}}_{q,i}^c$ which includes the entropy of the local class estimates from the drone q and the entropy of the global class beliefs. We define the joint target observation vector as $\mathbf{o}_q^t = \{\mathbf{o}_{q,i}^t\}_{i=1}^{I} = [\bar{\mathbf{o}}_q^p, \bar{\mathbf{o}}_q^c]$.

The reward function in this work is based on the formulation of Serra-Gómez et al. (2023). It provides rewards to the agent for successfully classifying each and all targets and reducing the entropy of target class beliefs. Additionally, it penalizes the agent for movement and for each time step in which the task remains incomplete. For more detailed information, we refer the interested reader to Serra-Gómez et al. (2023).

5.2.1. *Architecture*

The generalization ability of the learned policy $\pi_\phi(\mathbf{a}|\mathbf{o}_q)$ depends on the neural network architecture chosen. The main challenge lies in the size and dynamical changes over time of the set $\mathbf{o}_t^q = \{\mathbf{o}_{q,i}^t\}_{i=1}^{I}$.

Inspired by Relational Graph Convolutional Networks (Schlichtkrull et al., 2018) and self-attention mechanisms (Vaswani et al., 2017) used in static knowledge graphs, we employ a self-attention block (SAB) to capture the relationships among all targets at time t. Note that the focus in this first layer is on spatial features such as poses and velocities $\bar{\mathbf{o}}_p^q$, since the purpose is to encode important information including target visibility, observation perspective, occlusions, and potential simultaneous observations. Therefore, the initial layer is,

$$\begin{aligned}
\tilde{\mathbf{e}}_{i,p}^{1,h} &= F(\bar{\mathbf{o}}_{q,i}^p; \mathbf{W}_{q,h}^1) + \sum_{j\in\mathcal{J}} \lambda_{i,j}^h F(\bar{\mathbf{o}}_{q,j}^p; \mathbf{W}_{v,h}^1), \\
\mathbf{e}_{i,p}^1 &= LN(Res^1(LN(concat(\{\tilde{\mathbf{e}}_{i,p}^{1,h}\}_{h=1\ldots H})))), \\
\lambda_{i,j}^h &= \operatorname{softmax}\Big(\frac{1}{\sqrt{d_h}} F(\bar{\mathbf{o}}_{q,i}^p; \mathbf{W}_{q,h}^1)^{\top} F(\bar{\mathbf{o}}_q^p; \mathbf{W}_{k,h}^1)\Big)_j,
\end{aligned} \tag{11}$$

where $i \in \mathcal{J}$, $Res^l(x) = x + \sigma(F(x; \mathbf{W}^l))$, with σ being a ReLU activation function and F a parametric affine transformation. LN stands for Layer Normalization. $\mathbf{W}^1 \in \mathbb{R}^{d_{enc}\times(d_h H+1)}$ and $\mathbf{W}_{w,h}^1 \in \mathbb{R}^{d_h\times(d_{in}+1)}, w \in \{v, q, k\}$, are learnable parameters. d_{in}, d_h, d_{enc} are the dimensionality of the input, each head h, and the first layer. Note that each head h encodes a different relation λ^h between targets. To incorporate the information acquired about each target's class, we concatenate it with the latent representation of each target from the previous layer. Then, we map it back to a latent space of dimension d_{enc} using a learned linear layer. The process can be expressed as $\mathbf{e}_i^1 = F([\mathbf{e}_i^1, \bar{\mathbf{o}}_{q,i}^c]; \mathbf{W}_c)]$, where $\mathbf{e}_i^1$ represents the updated latent representation, $\bar{\mathbf{o}}_{q,i}^c$ is the class information of target i observed by drone q, and $\mathbf{W}_c$ is the learned weight matrix.

Next, we use a pooling multi-head attention mechanism (PMA) that incorporates a learned seed vector per head $\mathbf{v}_s^h \in \mathbb{R}^{d_h}$ to calculate the attention weights for a single query,

$$\begin{aligned}
\tilde{\mathbf{e}}^{2,h} &= \mathbf{v}_s^h + \sum_{j\in\mathcal{J}} \lambda_j^h F(\mathbf{e}_j^1; \mathbf{W}_{v,h}^2), \\
\mathbf{e}^2 &= LN(Res^2(LN(concat(\{\tilde{\mathbf{e}}^{2,h}\}_{h=1\ldots H})))), \\
\lambda_j^h &= \operatorname{softmax}\Big(\Big\{\frac{1}{\sqrt{d_h}} \mathbf{v}_s^{h,\top} F(\mathbf{e}_j^1; \mathbf{W}_{k,h}^2)\Big\}_{j\in\mathcal{J}}\Big)_j.
\end{aligned} \tag{12}$$

The output latent vector $\mathbf{e}^2$ is further processed by a fully connected layer to obtain the parameters $\mu_{\mathbf{a}_t}$ and $\log(\sigma_{\mathbf{a}_t})$ of a diagonal Gaussian distribution $\mathcal{N}(\mu_{\mathbf{a}_t}, \sigma_{\mathbf{a}_t})$ over viewpoints. The learned policy π_ϕ then samples recommended viewpoints $\mathbf{a}_t$ from this distribution. We assume that the drone can reach the recommended viewpoint before the next time step.

For training the network, we employ the Proximal Policy Optimization (PPO) algorithm (Schulman et al., 2017; Liang et al., 2018). PPO requires an estimate of the state-value $V^{\pi_\phi}(\mathbf{s}_t)$, which is approximated by a linear layer predicting $V^{\pi_\phi}(\mathbf{s}_t) \approx \mathbf{v}_v^\top \mathbf{e}^2$. This value estimation is used during training to guide the policy. The training process combines the surrogate loss and KL-divergence term to ensure stability. Additionally, an entropy regularization term is included to promote exploration (Haarnoja et al., 2017). For more detailed information and equations regarding the algorithm, we refer the reader to Schulman et al. (2017).

5.2.2. Low-level control MPC

During training, we assume the drone reaches the suggested viewpoint by the next time step. However, at test time we employ a low-level controller operating at a frequency $\frac{1}{\tau_l}$Hz ≫ $\frac{1}{\tau_h}$Hz, to guide it there while accounting for the drone dynamics. The controller solves the following receding-horizon constrained optimization problem:

$$\begin{aligned} \min_{\mathbf{y}_{1:N},\rho_{0:N-1}} \quad & \sum_{k=0}^{N-1} w_\rho \left\|\rho_k\right\| + w_g \frac{\left\|\mathbf{y}_N - \mathbf{a}_t\right\|}{\left\|\mathbf{y}^0 - \mathbf{a}_t\right\|} \\ \text{s.t.} \quad & \mathbf{y}_0 = \mathbf{y}_t, \quad \mathbf{y}_{k+1} = f(\mathbf{y}_k, \rho_k) \\ & \rho_k \in \mathcal{P}, \quad 0 \le k \le N-1 \end{aligned} \tag{13}$$

where ρ_k is the low-level control input sent to the robot, that needs to be inside the possible values $\mathcal{P}$, $f(\mathbf{y}_k,\rho_k)$ the internal dynamics and w_u and w_g are the respective weights of the stage and terminal costs. For more details, we refer the reader to Zhu and Alonso-Mora (2019) and Serra-Gómez et al. (2023). Although our full method accounts for the drone dynamics using this low-level controller, our formulation is flexible to other low-level controllers as long as they track the recommended viewpoint $\mathbf{a}_t$. This is why during simulation we employ both the in-built drone dynamic model and the controller from AirSim, see Shah et al. (2018) for more information.

6. Experiments

6.1. Environment

We use two high-fidelity virtual environments in Unreal Engine to test the presented framework using our previous work (Casao et al., 2023), where we provide the essential tools for creating multi-pedestrian scenarios. Photo-realistic simulators offer several advantages, including obtaining automatically labeled data, easily varying testing conditions, and developing autonomous robotics approaches by filling the gap of using perception information. Previous works have shown that methods developed in such environments, which are increasingly prevalent, can generalize to real-world scenes with augmentation techniques (Zhong et al., 2019; Luo et al., 2019).

The designed scenes are presented in Fig. 3. The first scene is a commercial street (Street), while the second is a green open area (Font). Their respective dimensions are 97 × 27 m and 97 × 50 m. In both scenes, we place three static cameras with overlapping views for global area monitoring and define distant starting points for the drones. The size of the images captured by the camera network is set to $1440x900$ and the field of view to 90 degrees.[3] Regarding communications, to be as faithful as possible to a real-world scenario, we set the drones to share information with each other as well as with the closest camera to them at the time. Communication between static cameras is limited to their direct neighbor, as shown in Fig. 3. Finally, the number of pedestrians present on the scene varies between episodes, and their trajectories are randomized.

Regarding the task of correctly visualizing people's attributes, we devise a marketing study on clothing brands as a use case. Specifically, we create different pedestrians with a logo on the front of their T-shirts which can be visualized exclusively from the frontal view of the person.

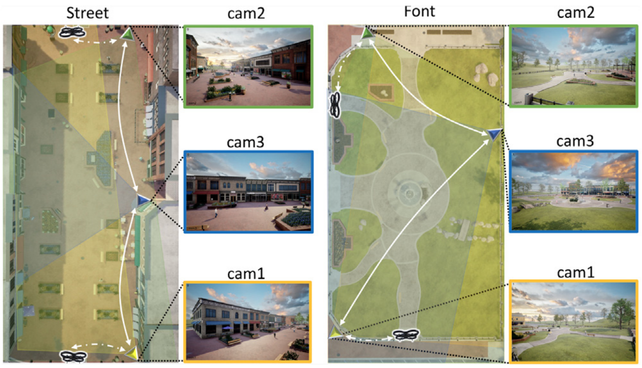

Fig. 3. Experimental environments used to evaluate the proposed framework. On the left, we show the setup for the experiments performed in the commercial street, *Street*, and on the right the setup for the font area, *Font*. The starting points of the two drones used as mobile cameras are also shown and they always communicate with each other.

6.2. Evaluation metrics

To comprehensively evaluate the proposed approach for distributed **multi-target tracking**, the common CLEAR MOT metrics (Bernardin and Stiefelhagen, 2008; Ristani et al., 2016) are adopted for evaluation:

- Multiple Object Tracking Accuracy (**MOTA**): measures failures during the tracking taking into account the number of misses, false positives, and mismatches.
- Identity F1 Score (**IDF1**): evaluates the capability of the system for preserving the identities over time.
- Multiple Object Tracking Precision (**MOTP**): shows the ability of the tracker to estimate precise object positions through the error in estimated position.

The above evaluation is performed in the image plane where metrics require setting a threshold between the ground truth and the resulting trackers in order to consider a tracker valid. We evaluate the resulting bounding boxes in the image plane using a minimum intersection over union (IoU) of 0.3 as the threshold to validate the trackers. The final tracking results are those obtained as output of the Distributed Tracker Manager. The final result is the median of the cameras in the network.

Regarding the acquisition of the correct people's viewpoint obtained from the **active perception** approach, the evaluation is performed using a black-box clothing brands detector. Thus, we employ two metrics:

- Trackers Classified (**TC**): measures the percentage of trackers whose beliefs are higher than 95%.
- Precision (**P**): evaluates the percentage of trackers for which their beliefs exceed 95% and correctly identifies their brand (attribute).

To associate each tracker with a ground truth brand class, we perform a linear sum assignment problem between the trackers and ground truth bounding boxes. The ground truth bounding boxes obtained from the simulator contain the person's attribute class.

6.3. Sequence evaluated

Several sequences are evaluated in each one of the environments with their corresponding ground truth being automatically obtained from the simulator. Specifically, we varied the number of pedestrians to assess the performance for 5, 10, and 15 pedestrians. Thus, the conducted experiments are named as sparse, medium, and busy for 5, 10, and 15 pedestrians respectively, resulting in the following sequences: *Street Sparse*, *Street Medium*, *Street Busy* for *Street* environment, and *Font Sparse*, *Font Medium*, *Font Busy* for *Font* environment. All of them have the same length of 500 frames. The Unreal project will be released upon

[3] The rest of the camera parameters are those set by default in Unreal Engine and AirSim.

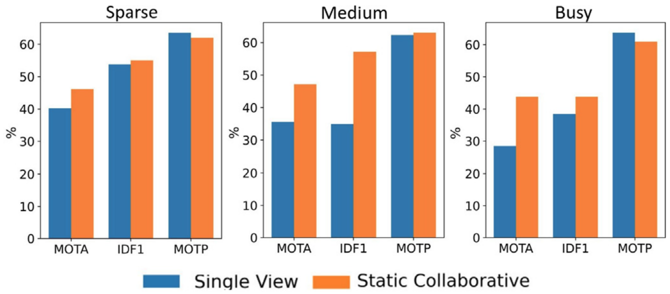

Fig. 4. Comparison of the cameras responsible for distributed multi-target tracking collaborating with each other with a chain graph of communications (Static Collaborative) and a single view tracking with isolated cameras (Single View).

Table 1
Percentage of trackers classified (TC) and percentage of trackers correctly identified (P) in the *Street* sequences. Results for the baseline static camera network (SC) and our hybrid system with two mobile cameras (MC).

Method	Classification Process (%)					
	Street Sparse		*Street Medium*		*Street Busy*	
	↑TC	↑P	↑TC	↑P	↑TC	↑P
SC	75	75	80	60	70.83	58.33
MC	71.5	64.3	76.2	66.7	81.5	74.1

acceptance together with the recorded sequence and the extrinsic of all the cameras to facilitate the comparison with the proposed tracking approach.

6.4. Results and settings

In the following, we explain the baselines selected to compare the proposed method in the *Street* sequences and perform a detailed analysis of the obtained results. To conclude the experiments, we also present the performance of our approach in the metrics described above for both environments, *Street* and *Font*. We set the parameters defined in the method for all the experiments to $\tau_{d\,LDA} = 1$, $\alpha = 700$, $\tau_{a\,LDA} = 0.55$, $\tau_{d\,DTM} = 2$, $\tau_{a\,LDA} = 1$, $\tau_l = 0.05$, $\tau_h = 0.25$ and the size of the targets' gallery is set to 10.

Collaborative behavior analysis. To demonstrate the benefits of collaborative behavior between nodes in a multi-target tracking network, we gather the three cameras from our system responsible for tracking and assessed their performance with and without communication. The first case (Static Collaborative) follows the initial setup where cameras communicate exclusively with their direct neighbor in a chain graph (Fig. 3). In the second setup, the different cameras perform individual tracking without any communication between nodes (Single View). The obtained results, shown in Fig. 4, demonstrate the benefits of sharing information once per iteration with minimum communications so that no node in the network is isolated. The Static collaborative setup achieves up to 21% and 15% of improvement in the IDF1 and MOTA metric, respectively, in comparison with the tracking in Single View. Therefore, we can conclude that in large scenarios, the use of collaborative cameras with overlapping perspectives enhances tracking performance in comparison to the use of independent cameras.

Mobile cameras analysis. Furthermore, we evaluate the efficiency of our mobile cameras (MC) to correctly visualize the desired people's viewpoint against a baseline of static cameras (SC). The static setup is composed of five cameras, the three already existing in the system and two more located on the other side of the street for more visual coverage of the scene. Communications among the five cameras are defined as a ring graph, i.e., each camera shares information with its two nearest neighbors. As a consequence of the distributed nature of the system, the static cameras collaborate to gain knowledge of the overall scene, and the evaluation of the correct viewpoint visualization is performed individually. The final results of the baseline are the median of all the cameras in the system.

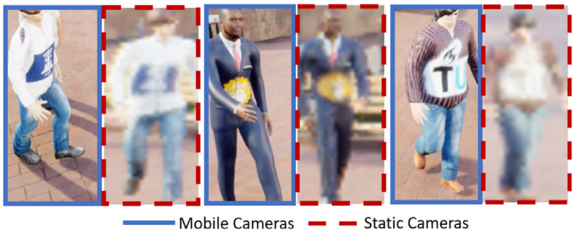

Fig. 5. Examples of people images captured from the correct viewpoint: mobile cameras (blue box) and static cameras (red dashed box) in the *Street Busy* sequence. Every pair of columns displays images of the same person.

The results obtained of the percentage of trackers classified (TC) and correctly identified their brands (P) with beliefs higher than 95% are presented in Table 1. In the sparse scenario, where the occlusions between targets are not frequent, the static camera setup gets better results than the mobile cameras. However, in more crowded scenarios, static cameras struggle to avoid occlusions for obtaining a view with high confidence from the pedestrian. In contrast, mobile cameras can be actively positioned to capture the desired viewpoint, achieving a coverage (TC) of 81.5%, against the 70.83% obtained from the static setup, in the most challenging scenario (*Street Busy*). In addition, the quality of the people data captured by each one of the systems is unmatched. Fig. 5 shows examples of the same pedestrian captured with the mobile cameras (blue box) and with the static cameras (red dashes box). Every two columns correspond to the same person and we can notice the great difference in quality. The images from mobile cameras revealed much more clear details than the static ones, whose images are of low quality and blurry. These result in better identification of the person's brand in most of the sequences evaluated (P).

Final evaluation. As a summary, we present the performance of the proposed framework in both photo-realistic environments, *Street* and *Font*. The results obtained are shown in Table 2, from which we can conclude that the method is consistent under various conditions, including different numbers of people, size of the space, and type of environments. Specifically, the experiments focus on evaluating sparse, medium, and busy scenarios, with 5, 10, and 15 pedestrians, respectively. Moreover, the *Font* environment is larger than the *Street* environment with static cameras located further away from the path where people walk, making it more challenging for monitoring. Finally, we also perform a measurement of the mean time required by each of the modules comprising the proposed framework: detection 0.0198 s, local data association per tracker 0.038 s, distributed Kalman filter per tracker 0.002 s, distributed tracker manager 0.0015 s, class information fusion per tracker 0.00004 s, viewpoint control policy 0.005 s. The complete evaluation is conducted in one computer with an Intel® Core™ i7-9700 CPU @ 3.00 GHz × 8 and a Nvidia GeForce GTX 1070. Both tracking modules, with mobile and static cameras, and classification modules, with mobile cameras, work in parallel. Provided that the poses of new targets are estimated and relayed to the mobile cameras within τ_h, our framework operates in real time. This is not a strict constraint, as there is allowable latency; however, it is crucial that tracked targets remain within the recommended viewpoint FOV during any such delays.

In addition to the numerical results, Fig. 6 displays examples of images captured by the hybrid system at a specific time. The first row corresponds to images from the *Street* environment and the second row from the *Font* scenario. The overall understanding of the scene is mainly performed by the static cameras although the drones also

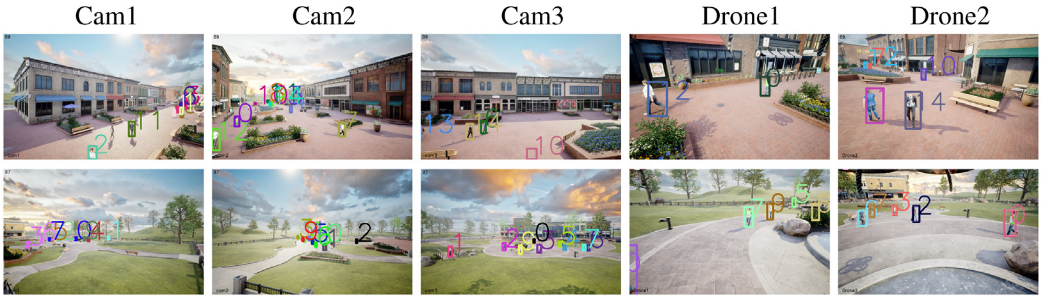

Fig. 6. Example of images captured by the hybrid system. First row *Street Busy* sequence and second row *Font Medium* sequence. Static cameras are mainly responsible for the global understanding of the scene while mobile cameras (drones) capture pedestrian images from the desired viewpoint.

Table 2
Results of the evaluated metrics in the *Street* and the *Font* sequences where sparse, medium, and busy environments are analyzed.

Sequence	Multi-target Tracking			Classification	
	↑MOTA%	↑IDF1%	↑MOTP%	↑TC%	↑P%
Street Sparse	54.18	48.34	61.43	71.5	64.3
Street Medium	43.62	42.57	60.45	76.2	66.7
Street Busy	38.83	42.52	60.1	81.5	74.1
Font Sparse	38.24	41	58.1	100	75
Font Medium	47.22	55.52	59	93.3	53.35
Font Busy	40.34	47.96	63.63	72.73	63.63

assist in the distributed tracking, while the close-up person images are gathered from the mobile cameras. For example, in the first row, Drone2 correctly captures the viewpoint of the target with local identity 14, and in the second scenario, Drone1 accomplishes its goal with local identity 7. In the supplementary material, we include more examples of the complete framework working on both scenarios.

7. Conclusions

In this work, we have presented a collaborative hybrid system comprised of static and mobile cameras where all of them cooperate for pedestrian monitoring and high-resolution visualization of certain people's attributes. The proposed framework performs multi-camera distributed tracking providing a global understanding of the scene for which the static cameras are mainly responsible. We demonstrate that by allowing collaboration between cameras through sharing information once per cycle with the closest nodes, the multi-target tracking improves up to 21 points in the IDF1 metric and up to 15 points in MOTA. Global scene awareness and the current state of drones are used by the viewpoint control policy to provide a new position and orientation for mobile cameras whose goal is capturing a desired viewpoint of the people as quickly as possible. In comparison with a static multi-camera system, mobile cameras are able to capture the required viewpoint with higher precision in most of the scenes evaluated.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Ana Cristina Murillo reports financial support was provided by Ministerio de Ciencia, Innovación y Universidades. Ana Cristina Murillo reports financial support was provided by Dirección General de Aragón.

Data availability

The data used in this work is available at: https://sites.google.com/unizar.es/poc-team/research/hlunderstanding/collaborativecameras.

Appendix A. Supplementary material

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.cviu.2023.103876.

References

Alcántara, A., Capitán, J., Cunha, R., Ollero, A., 2021. Optimal trajectory planning for cinematography with multiple unmanned aerial vehicles. Robot. Auton. Syst. 140, 103778.

Atanasov, N., Sankaran, B., Le Ny, J., Pappas, G.J., Daniilidis, K., 2014. Nonmyopic view planning for active object classification and pose estimation. IEEE Trans. Rob. 30 (5), 1078–1090.

Bernardin, K., Stiefelhagen, R., 2008. Evaluating multiple object tracking performance: CLEAR MOT metrics. J Image Video Process..

Bisagno, N., Conci, N., Rinner, B., 2018. Dynamic camera network reconfiguration for crowd surveillance. In: International Conference on Distributed Smart Cameras.

Byeon, M., Yoo, H., Kim, K., Oh, S., Choi, J.Y., 2018. Unified optimization framework for localization and tracking of multiple targets with multiple cameras. Comput. Vis. Image Underst. 166, 51–65.

Casao, S., Naya, A., Murillo, A.C., Montijano, E., 2021. Distributed multi-target tracking in camera networks. In: International Conference on Robotics and Automation. IEEE, pp. 1903–1909.

Casao, S., Otero, A., Serra-Gómez, Á., Murillo, A.C., Alonso-Mora, J., Montijano, E., 2023. A framework for fast prototyping of photo-realistic environments with multiple pedestrians. In: International Conference on Robotics and Automation. IEEE.

Ge, X., Han, Q.L., Zhang, X.M., Ding, L., Yang, F., 2019. Distributed event-triggered estimation over sensor networks: A survey. IEEE Trans. Cybern. 50 (3), 1306–1320.

Guo, Y., Liu, Z., Luo, H., Pu, H., Tan, J., 2022. Multi-person multi-camera tracking for live stream videos based on improved motion model and matching cascade. Neurocomputing 492, 561–571.

Haarnoja, T., Tang, H., Abbeel, P., Levine, S., 2017. Reinforcement learning with deep energy-based policies. In: International Conference on Machine Learning.

He, L., Liu, G., Tian, G., Zhang, J., Ji, Z., 2019. Efficient multi-view multi-target tracking using a distributed camera network. IEEE Sens. J..

Hill, T., Miller, J., 2011. How to combine independent data sets for the same quantity. Chaos 21 (3), 033102 (1–8).

Jeon, B.F., Shim, D., Jin Kim, H., 2020. Detection-aware trajectory generation for a drone cinematographer. In: IEEE/RSJ Int. Conf. on Intelligent Robots and Systems. pp. 1450–1457.

Kamal, A.T., Bappy, J.H., Farrell, J.A., Roy-Chowdhury, A.K., 2015. Distributed multi-target tracking and data association in vision networks. IEEE Trans. Pattern Anal. Mach. Intell. 38 (7), 1397–1410.

Kent, D., Chernova, S., 2020. Human-centric active perception for autonomous observation. In: IEEE Int. Conf. on Robotics and Automation. pp. 1785–1791.

Kuhn, H.W., 1955. The hungarian method for the assignment problem. Nav. Res. Logist. Q. 2 (1–2), 83–97.

Li, Z., Liang, Y., Xu, L., Ma, S., 2023. Distributed extended object tracking information filter over sensor networks. Internat. J. Robust Nonlinear Control 33 (2), 1122–1149.

Li, X., Su, Y., Liu, Y., Zhai, S., Wu, Y., 2018. Active target tracking: A simplified view aligning method for binocular camera model. Comput. Vis. Image Underst. 175, 11–23.

Li, J., Xu, J., Zhong, F., Kong, X., Qiao, Y., Wang, Y., 2020. Pose-assisted multi-camera collaboration for active object tracking. In: AAAI Conference on Artificial Intelligence, Vol. 34. (01), pp. 759–766.

Liang, E., et al., 2018. RLlib: Abstractions for distributed reinforcement learning. In: Int. Conf. on Mach. Lear..

Luo, W., Sun, P., Zhong, F., Liu, W., Zhang, T., Wang, Y., 2019. End-to-end active object tracking and its real-world deployment via reinforcement learning. IEEE Trans. Pattern Anal. Mach. Intell. 42 (6), 1317–1332.

Mekonnen, A.A., Lerasle, F., Herbulot, A., 2013. Cooperative passers-by tracking with a mobile robot and external cameras. Comput. Vis. Image Underst. 117 (10), 1229–1244.

Miller, I.D., Cladera, F., Smith, T., Taylor, C.J., Kumar, V., 2022. Stronger together: Air-ground robotic collaboration using semantics. IEEE Robot. Autom. Lett. 7 (4), 9643–9650.

Patten, T., Martens, W., Fitch, R., 2018. Monte Carlo planning for active object classification. Auton. Rob. 42 (02), 391–421.

Patten, T., Zillich, M., Fitch, R.C., Vincze, M., Sukkarieh, S., 2016. Viewpoint evaluation for online 3-D active object classification. IEEE Robot. Autom. Lett. 1 (1), 73–81.

Popović, M., Hitz, G., Nieto, J., Sa, I., Siegwart, R., Galceran, E., 2017. Online informative path planning for active classification using UAVs. In: IEEE Int. Conf. on Robotics and Automation. pp. 5753–5758.

Quach, K.G., Nguyen, P., Le, H., Truong, T.D., Duong, C.N., Tran, M.T., Luu, K., 2021. Dyglip: A dynamic graph model with link prediction for accurate multi-camera multiple object tracking. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 13784–13793.

Ristani, E., Solera, F., Zou, R., Cucchiara, R., Tomasi, C., 2016. Performance measures and a data set for multi-target, multi-camera tracking. In: European Conference on Computer Vision. pp. 17–35.

Schlichtkrull, M., Kipf, T., Bloem, P., Berg, R., Titov, I., Welling, M., 2018. Modeling relational data with graph convolutional networks. In: Extended Semantic Web Conference. pp. 593–607.

Schranz, M., Andre, T., 2018. Towards resource-aware hybrid camera systems. In: International Conference on Distributed Smart Cameras.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., Klimov, O., 2017. Proximal policy optimization algorithms. ArXiv arXiv:1707.06347.

Serra-Gómez, Á., Montijano, E., Böhmer, W., Alonso-Mora, J., 2023. Active classification of moving targets with learned control policies. IEEE Robot. Autom. Lett. 8 (6), 3717–3724.

Shah, S., Dey, D., Lovett, C., Kapoor, A., 2018. Airsim: High-fidelity visual and physical simulation for autonomous vehicles. In: Field and Service Robotics. Springer, pp. 621–635.

Sock, J., Garcia-Hernando, G., Kim, T.-K., 2020. Active 6D multi-object pose estimation in cluttered scenarios with deep reinforcement learning. In: IEEE/RSJ Int. Conf. on Intel. Rob. and Syst.. pp. 10564–10571.

Soto, C., Song, B., Roy-Chowdhury, A.K., 2009. Distributed multi-target tracking in a self-configuring camera network. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition. IEEE, pp. 1486–1493.

Trujillo, J.C., Munguía, R., Ruiz-Velázquez, E., Castillo-Toledo, B., 2019. A cooperative aerial robotic approach for tracking and estimating the 3D position of a moving object by using pseudo-stereo vision. J. Intell. Robot. Syst. 96, 297–313.

Vaswani, A., et al., 2017. Attention is all you need. In: Adv. in Neur. Inform. Processing Systems, Vol. 30. pp. 1–11.

Wei, L., Zhang, S., Gao, W., Tian, Q., 2018. Person transfer gan to bridge domain gap for person re-identification. In: IEEE Conf. on Computer Vision and Pattern Recognition. pp. 79–88.

Xompero, A., Cavallaro, A., 2022. Cross-camera view-overlap recognition. In: European Conference on Computer Vision. Springer, pp. 253–269.

Xu, Q., et al., 2021. Towards efficient multiview object detection with adaptive action prediction. In: IEEE Int. Conf. on Robotics and Automation. pp. 13423–13429.

Yu, J., Vincent, J.A., Schwager, M., 2022. Dinno: Distributed neural network optimization for multi-robot collaborative learning. IEEE Robot. Autom. Lett. 7 (2), 1896–1903.

Zhang, R., Wu, L., Yang, Y., Wu, W., Chen, Y., Xu, M., 2020. Multi-camera multi-player tracking with deep player identification in sports video. Pattern Recognit. 102, 107260.

Zhong, F., Sun, P., Luo, W., Yan, T., Wang, Y., 2019. Ad-vat+: An asymmetric dueling mechanism for learning and understanding visual active tracking. IEEE Trans. Pattern Anal. Mach. Intell. 43 (5), 1467–1482.

Zhou, Y., Xiao, J., Zhou, Y., Loianno, G., 2022. Multi-robot collaborative perception with graph neural networks. IEEE Robot. Autom. Lett. 7 (2), 2289–2296.

Zhou, K., Yang, Y., Cavallaro, A., Xiang, T., 2021. Learning generalisable omni-scale representations for person re-identification. Trans. Pattern Anal. Mach. Intell..

Zhu, H., Alonso-Mora, J., 2019. Chance-constrained collision avoidance for MAVs in dynamic environments. IEEE Robot. Autom. Lett. 4 (2), 776–783.