Adaptive Cross-Camera Video Analytics at the Edge

Kaiyang Chen*, Yifei Zhu*, Zhu Han† and Xudong Wang*
*UM-SJTU Joint Institute, Shanghai Jiao Tong University, China
†Department of Electrical and Computer Engineering, University of Houston, USA
Email: kaiyang-chen@sjtu.edu.cn, yifei.zhu@sjtu.edu.cn, zhan2@uh.edu, wxudong@ieee.org

***Abstract*—Cross-camera video analytics is a major video analytic task that associates and analyzes information across multiple cameras. However, the searching cost for existing cross-camera tracking tasks grows linearly with the number of cameras, leading to substantial cost in large-scale camera systems. Although correlation among cameras can greatly reduce the searching cost, our empirical analysis reveals that the correlation actually changes over time, leading to sub-optimal performance for schemes leveraging rigid correlation information. Furthermore, adjusting the correlations to dynamically guide the searching process is extremely challenging due to the high construction cost. In this paper, we propose an adaptive cross-camera video analytics framework under the guidance of fine-grained estimated correlation information. Specifically, we propose a mean-field game approach to estimate the dynamic correlation with only the initial correlation and the destination correlation. We first carefully craft the cost functions and constraint functions to model the dynamics of the users in the camera systems, and formulate the correlation estimation problem as a tracking-cost minimization problem. Considering the enormous number of interactions embedded in the problem, we further reformulate the proposed problem by introducing the correlation as the mean-field term. Given the complexity to solve the equilibrium, we adopt a G-prox primal-dual hybrid gradient algorithm to solve our problem efficiently. Consequently, the correlation from the initial to the destination can also be inferred over time. Extensive experiments on a real-world dataset demonstrate that our adaptive cross-camera video analytics framework based on fine-grained correlation can reduce the overall workload by 36% in general. For queries with a large searching space, the overall workload reduction can even be reduced by 40 times with 6% precision improvement.**

I. Introduction

With the advancement of computer vision techniques and the wide deployment of cameras, video analytics has been applied in a wide range of industrial verticals. It is estimated that about 130 million surveillance cameras are deployed in US, generating over 10 billion hours of video data per week; even one camera can generate hundreds of gigabytes of data [1]. Cross-camera video analytics refers to associating and tracking targeted objects across multiple cameras. Cross-camera tracking tasks can be widely found in applications like transportation, retailing, public security, etc [2], [3]. For example, in AMBER alerts, law enforcement has to search through video footage from thousands of cameras to locate the suspected vehicles or person [4].

This work is supported by SJTU Explore-X grant. Zhu Han's work is partially supported by NSF CNS-2128368, CNS-2107216. Corresponding author: Yifei Zhu

Existing studies on video analytics mainly focus on analysis of a single stream [5], [6], [7], or handling the resource management problems in the analytic systems [3], [8]. They mostly ignore the content correlation among networked cameras, and either treat each stream as an independent entity or focus on the exploitation of inter-frame correlation via frame sampling. For cross-camera video analytics, as the objects move across cameras, knowing the temporal and spatial correlation between cameras is extremely helpful to perform quick and cost-effective searching, since objects of interest in real world only appear in the view of certain cameras at certain given time [9].

Given the objects of interests in a multi-camera system, the spatial correlation is commonly defined as the probability distribution of these objects' appearance in the field of the destination cameras from the source cameras. Correspondingly, the temporal correlation approximates the objects' travelling time distribution from the source camera to the destination camera. Such correlation information has mostly been investigated on improving object tracking accuracy in computer vision field but without detailed study on its implications of workload reduction [10], [11], [12]. Recent efforts in the distributed system field reveal the benefit of these spatial and temporal correlations in reducing the searching cost given the growing need for cross-camera video analytics [9]. In these studies, the correlation is first generated from a short offline profiling period, and is then used to guide the searching in all the following videos in the system. Frames from uncorrelated video sequences with respect to both spatial and temporal dimensions, like distant cameras in short periods, are filtered, and only those highly correlated cameras in the most likely time period are searched.

However, prior work all assumes the correlation is stable over time, which has never been verified yet. In fact, intuitively, the spatial correlations can be dynamic since the direction of traffic flow in real-life is time-dependent. For example, for cameras watching over an urban-rural combination, in the morning rush hour, most of the vehicles passing by will be driving in the direction of the city, while in the evening rush hour, more vehicles are heading out of the city for home. The possible changes in the objects' spatial direction over time also imply the changes of spatial correlation over time.

Given such an observation, when performing searching tasks on past periods, one primitive approach to keep track of the drift in the correlation between cameras is to keep dividing video footage into small periods, and use a portion of

historical data in every period to generate the corresponding models. With the help of fine-grained correlations, fewer searching needs be triggered on edge cameras and overall searching cost can be reduced. However, naively generating extra correlations in this way introduces substantial expenses. Frequent profiling makes the model construction cost difficult to be amortized in the following limited time. Without frequent profiling, resorting to data-driven approaches to approximate the correlation changes also becomes infeasible due to the limited training samples.

In this paper, we propose an adaptive cross-camera video analytics framework that leverages the dynamic correlation estimation to accelerate cross-camera searching. We first conduct an empirical analysis on real-world dataset to confirm the dynamics of the spatial correlation model and the potential benefit in searching cost reduction if an adaptive correlation model is adopted. We then carefully craft the cost model and system constraint models to describe the real-world dynamics. The enormous number of interactions among users in the camera systems make the ideal object tracking problem extremely challenging and costly to be solved. To handle this complexity, a novel MFG framework is leveraged to transform this cross-camera tracking problem to the cost minimization problem subject to the system dynamics. Our desired correlation elegantly emerges as an approximation to the target camera's density field in the MFG problem. We then propose a G-prox primal-dual hybrid gradient algorithm (PDHG) to solve the problem efficiently. The resulting algorithm calculates the fine-grained correlation based on the mere knowledge of the starting correlation and the destination correlation, greatly reducing the profiling cost. To our best knowledge, this is the first work to study the adaptive cross-camera video analytics problem, and the first effort to answer the correlation estimation with the help of MFG theory. Our contribution can be summarized as follows:

- We reveal that the spatial correlation changes over time and identify the significant potential in leveraging such a dynamic correlation based on a thorough data analysis.
- A novel MFG framework is applied to transform the ideal tracking problem into a mean-field game, with the desired correlation information automatically revealed as the mean-field term.
- Given the complexity of solving the problem, we propose an efficient G-prox primal-dual hybrid gradient algorithm with the linear computation complexity and grid-independent convergence rate.
- The extensive experiments on a real-life dataset demonstrate that our adaptive correlation can reduce 36% workload in general, and achieve a 40 times workload reduction and a 6% increase in precision for larger queries.

The rest of the paper is organized as follows. In Section II, we introduce the general background for cross-camera video analytics and correlation model. In Section III, we conduct data analysis to confirm that the correlation is dynamic in practice. In Section IV, we formally present the cost models and formal formulation. In Section V, we reformulate the problem using the framework of the mean-field game and solve it using a primal-dual approach. In Section VI, performance evaluation is conducted. Related work is reviewed in Section VII, followed by the conclusion in Section VIII.

II. Background and Preliminaries

In this section, we first introduce the general background for cross-camera video analytics. We then present the definition, construction, and usage of both spatial correlation and temporal correlation models in cross-camera video analytics.

A. Cross-camera video analytics

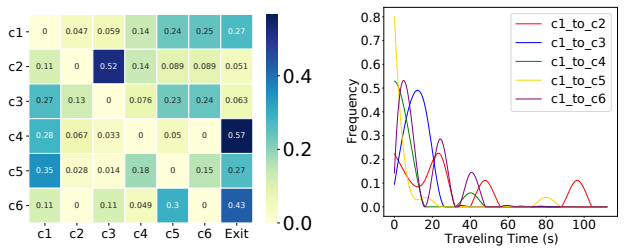
A general camera system in practice consists of a large number of smart cameras responsible for video capturing and processing, as well as a central server coordinating the cross-camera tracking and searching tasks. Previously, when the server wants to retrieve the path of a target from a given camera, the central server will trigger searching on all edge devices simultaneously once the target disappeared from source camera. While searching for all cameras brings substantial computation cost, recent studies and systems start to leverage the correlation among different cameras to greatly reduce the searching space, and thus reduce the searching cost. Specifically, the central server will maintain a correlation model capturing the objective movement correlation among different cameras. When a target disappears from the current known camera, matching tasks will only be triggered on those correlated cameras indicated by the correlation model.

B. Correlation model: definition, construction, and usage

Definition. Correlation models can defined and constructed from two perspectives: *spatial* and *temporal*. A spatial correlation model captures the historical movements of objects in a camera system and describes the possibilities of an object going directly to all other destination cameras from a certain source camera. If an object goes to B from A via C, this trajectory is counted as the movement from A to C, and from C to B, separately. The spatial correlation model $S(c_s, c_t)$ is defined as the ratio of the number of objects that appear on the target camera (c_t) first after appearing from the source camera (c_s) to the total number of people who appear on the source camera. In other words, $S(c_s, c_t)$ approximates the probability of objects that leaves c_s entering c_t directly.

A temporal correlation model describes the relationship among different cameras with respect to traveling time. The temporal correlation model $T(c_s, c_t)$ is quantified as the number of people from source camera to target camera in a given time period ($[t_1, t_2]$) divided by the total number of people in the same trajectory. Namely, it approximates the traveling time distribution in the camera system from a source camera to a destination camera.

As an example, Fig.1 presents the temporal and spatial correlation model constructed from a 6-camera system [13]. In the spatial correlation model, each row in the model represents the source camera while each column is a destination camera.



(a) Spatial correlation (b) Temporal correlation (camera 1)

Fig. 1. Overall spatial correlation and a specific camera's temporal correlation from a 6-camera system

The "Exit" entry here represents the probability of the object leaving the current monitoring area and never showing up again. Fig. 1(b) is a temporal correlation model constructed for camera 1 using the same portion of data. As can be seen, once an object leaves from camera 1 to camera 2, it will most likely appear at 4 different time intervals.

Construction. During the initial correlation construction phase, edge cameras are responsible for objects detection and the corresponding feature extraction. Every time a new feature, representing an object, is extracted in an edge camera, it will be matched with a local feature gallery to test whether this object has been shown before. If not, the camera will send a message tuple in form of (f_i, t_i, c_i) to the central server, where f_i is the feature of a locally unknown object, t_i is its detected time, and c_i is the detected camera-id. In practice, we reduce the communication cost and preserve privacy by only uploading the feature extracted by a local neural network running on edges. The central server maintains a global gallery to store the features of shown objects in the camera system; each object has its list of their shown-up time and camera. Every message tuple sent from edge to the center is compared with features in the gallery. If matched, the detection time and camera will be added to the corresponding object list accordingly. If not, a new object will be added to the global gallery. With all the data pairs detected from the edge devices in the model generation phase, the central server is able to extract the trajectory of objects moving through the camera system, and then construct the spatial and temporal correlation in the past periods based on the definitions we described before.

Usage Given a query request for an object, with the help of the spatial correlation model, when an object of interest leaves its current camera c_s at t_c, we can just trigger searching tasks on those target cameras c_t with spatial correlation $S(c_s, c_t)$ greater than a threshold S_{th}, and omit those frames on the uncorrelated cameras. This avoids extra searching on all edge devices and greatly saves the searching cost with respect to the camera system. Furthermore, by utilizing the temporal correlation of camera pairs, we can narrow down the searching time range in target cameras. With the temporal correlation $T(c_s, c_t)$, we can obtain a $(1-T_{th})$ percent confidence interval for the passage time $[t_1, t_2]$ between source (c_s) and target camera (c_t) and only search frames and object on c_t in duration $[t_c + t_1, t_c + t_2]$. If the system failed to find an object of interest at first correlated filtering tracking, you may choose to trigger a search that relaxes the threshold T_{th} and S_{th} or a global search for the remaining frames depending on your system's application and needs.

For example, given the spatial-temporal correlation model in Fig.1, the spatial threshold $S_{th} = 0.1$, and the temporal threshold $T_{th} = 0.05$, for a query object first shown up in camera 1, we can just trigger searching tasks on camera 4, 5 and 6, but not 2, 3. Because $S(c_1, c_2)$ and $S(c_1, c_3)$ are below the threshold and thus being regarded as uncorrelated cameras. Based on the temporal correlation model, we can further narrow down the searching time from the total time span to a specific time range. Under the given threshold, we only need to examine time intervals 0 to 11 on camera 5 since 95% of objects' traveling time from camera 1 to camera 5 are covered by these periods.

III. Data Analysis and Motivation

A. Spatial correlation model evolves over time

Existing studies on correlation-supported cross-camera analytic systems usually calculate the correlation model offline and use the resulting model through the entire online analytic part. Since the objects moving speeds usually are within a certain range, the temporal correlation over a representative period should captures the temporal pattern of the objects intuitively. However, how representative the spatial correlation model is remains unknown and has not been studied before. Instinctively, such correlation can also be affected by traffic tides associated with time and the environment. For example, suppose that camera A is in the dormitory building, camera B is in the cafeteria, and camera C is in the academic building. During class time, there may be more people following the trajectory of A to C to attend classes. But during dinner time, more people will go from A to B for their meals. In order to verify this assumption, we use the Market-1501 dataset[1] and evenly divide the time range of the dataset into 4 consecutive periods. The resulting spatial correlation models for these four periods are presented in Fig.2. As can be seen, the overall correlation model indeed changes over time. Take camera 5 as an example, the most correlated camera changes from camera 1 in period 1 to camera 3 in period 4. However, camera 3 is not even being regarded as correlated with camera 5 in period 1 if the system operator sets the filtering threshold $S_{th} > 0.033$.

We further dive into this multi-camera system and extract the number of visits from camera 1 to all other source cameras for 4 consecutive periods in Fig.3. We can see that the amount of visits from camera 1 to camera 6 is the second-highest in the first period, but nearly no people go through this path in periods 2, 3, and 4. This also demonstrates that the dynamics of correlation in the camera network are significant. Consequently, only using the correlation relationship generated from a specific period can not be representative enough and

[1]Details about this dataset can be found in the evaluation section.

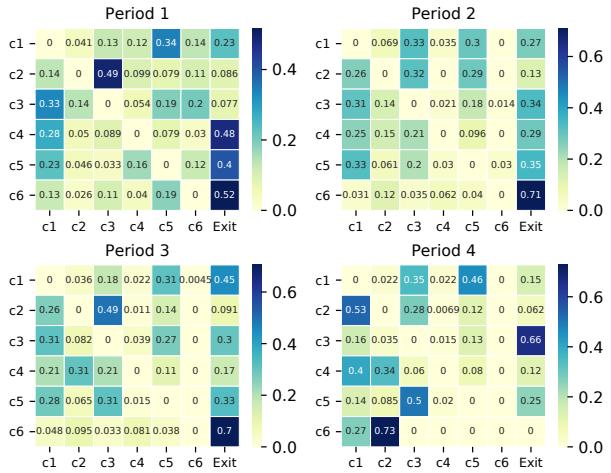


Fig. 2. Spatial correlation in four consecutive periods

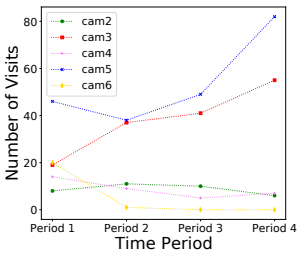


Fig. 3. Dynamics of transportation to camera 3 from sources camera

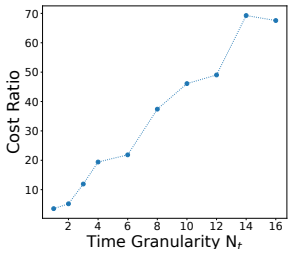


Fig. 4. Benefit of adopting a dynamic model in different time granularity

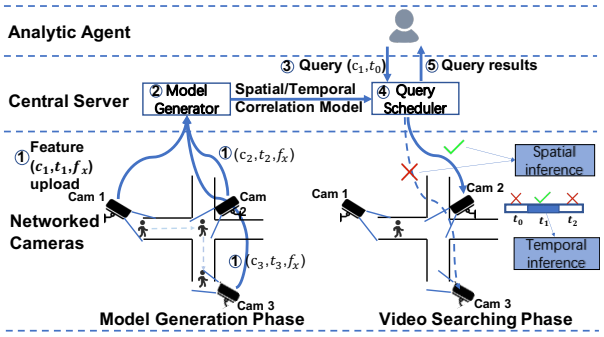


Fig. 5. An overview of our cross-camera video analytics system.

may even be counterproductive. Note that we have not found significant changes in the temporal correlation between cameras because of the fixed distance between camera locations and the similar range of objects' walking speed. Therefore, in the following paper, the correlation model refers to the spatial correlation model, if not specified explicitly. Our empirical analysis thus reveals the temporal property within the spatial correlation model.

B. Potential of a dynamic correlation model

Previous analysis reveals that the spatial correlation model is highly dynamic. If we use the obsolete spatial correlation to guide query search for the later periods, the result can be easily sub-optimal and even becomes worse. Conversely, if we can dynamically update the spatial correlation model, and use the corresponding model to guide the query, the searching space can be fully reduced. In order to study the potential benefit of applying a dynamic model, we conduct a posterior cross-camera searching task based on the correlation models constructed under various time granularity. Namely, we divide the dataset into periods with different lengths and generate the corresponding spatial correlation in each period. We then trigger tracking on the same set of random query objects guided by the corresponding spatial correlation we have for that period. We calculate the cost ratio, defined by the value of the searching cost based on global searching divided by the searching cost aided by the dynamic models. The result is shown in Fig.4. We can find that if we use the whole dataset to generate a general global model, the tracking cost can be reduced by 3×, as indicated by the first point on the left. However, the cost saved can even reach 70× if we had the guidance from the corresponding fine-grained correlation model. The results demonstrate the significant potential of having such an accurate model that captures the correlation. It is also worthwhile to mention that there is a saturation point to the breakdown of time granularity with the cost ratio decreasing from 69.3× to 67.6× when the time granularity changes from 14 to 16. This is due to the intrinsic limitation of such a statistical model construction. If the profiling period is too short, and the number of objects that emerged in a period is too small, the generated correlation lacks its statistical representativeness, leading to more uncertain results.

Though our presented empirical analysis so far has demonstrated the significant benefit of adopting a dynamic model, the model construction cost is extremely expensive in practice. Naively dividing the examined period into finer granularity, and constructing fine-grained models based on each offline sub-period can lead to significant profiling costs. Furthermore, the construction of the correlation model relies on actually conducting object tracking tasks, which is exactly what we want to avoid as much as possible under the help of the correlation model. In other words, the approach towards model construction is in nature conflicts with the goal of searching for fewer frames. The more frequently we conduct offline profiling, the less gain it helps to our systems. *Therefore, we need a more efficient approach to construct the dynamic correlation models in order to fully unleash the power of correlations.*

IV. System Models and Problem Formulation

A. System model

We consider a multi-camera system responsible for video processing, as well as a central server coordinating cross-camera object searching tasks. We have an n-camera system deployed in a two-dimensional monitoring area S, with K users/agents traversing within the area. T is the time range

of the whole tracking task. The set of agents appeared in the camera system is denoted as $\mathbb{A} = \{a_1, ..., a_k, ..., a_K\}$ where a_k denotes the kth agent. We denote the physical location for all agents at time t as $\mathbb{X} = \{x_1(t), ..., x_k(t), ..., x_K(t)\}$ where $x_k(t) \in \mathbb{R}^2$. Let $\mathbb{V} = \{v_1(t), ..., v_k(t), ..., v_K(t)\}$ where $v_k(t)$ is the traveling speed for agent a_k at time t. The cross-camera objects tracking aims at identifying the next frames containing a queried object and its location based on a tracker. Our tracker outputs a predicted position for all agents at time t as $\mathbb{Y} = \{y_1(t), ..., y_k(t), ..., y_K(t)\}$, with the corresponding speed $\hat{\mathbb{V}} = \{\hat{v}_1(t), ..., \hat{v}_k(t), ..., \hat{v}_K(t)\}$. The overall system workflow is described in Fig.5: (1) As describe in Construction part of section II-B, edge cameras uploading message tuples containing detected feature to center server; (2) With all information from the edge, center server extracts the movement track of each person and then construct the temporal and spatial correlations; (3) An analytic agent is interested in querying the movement of a target disappear from c_1 at t_0; (4) The central server conducts spatial and temporal inferences aided by correlations generated at step 2, and triggers tasks to edge cameras accordingly; (5) The edge camera which identifies the target report back to the central server and then to the analytic agent.

B. Tracking cost modeling

We model the tracking cost with respect to the individual agent from two aspects: the speed deviation cost and the destination prediction cost.

Speed deviation cost. In practice, the traveling speed of agents usually falls within a certain range. The speed deviation cost aims at quantifying the fluctuation of agents' speed in the system. The more deviated an agent from the normal speed, the harder it is for us to construct a stable correlation model. To model the degree of difference from our predicted speed for agent and the system average speed, we use the l2-norm function to simulate the cost across all agents,

$$J_1 = \alpha_1 \frac{1}{K} \sum_{k=1}^{K} \int_0^T ||\bar{v}_k(t)||^2 dt, \tag{1}$$

$$\bar{v}_k(t) = \hat{v}_k(t) - \bar{v}, \tag{2}$$

where $\bar{v}_k(t)$ is the deviation of predicted speed from system actual average speed accordingly. α_1 is the coefficient to weigh the speed deviation cost and facilitate key parameters comparison.

Destination prediction cost. The destination prediction cost is the sum of the distance of agents between their actual target location and a predicted one at time t, which measures how well our predictions of agent dynamics simulate the real situation, i.e.,

$$J_2 = \alpha_2 \frac{1}{K} \sum_{k=1}^{K} dis(x_k(t), y_k(t)), \tag{3}$$

$$dis(x, y) = \begin{cases} 0, & x = y \\ 1, & x \neq y, \end{cases} \tag{4}$$

where `dis` is a boolean function return whether two inputs are identical, $x_k(t)$ is the real position of agent k at time t and $y_k(t)$ is the predict position of agent k at time t. Similarly, α_2 is a coefficient to weight the destination prediction cost.

System constraint. We introduce the kinematic modeling for a single user as an epitome for constraint at system scale, which is given by

$$dx_k(t) = v_k(t)dt \tag{5}$$

for $k = 1, 2, ..., K$. In addition, we have the information for the initial position for every agent, that is

$$x_k(0) = x_{k0} \tag{6}$$

for $k = 1, 2, ..., K$.

C. Problem formulation

With the system cost functions and constraint functions defined above, we intend to find a tracker with minimized tracking cost subject to the user dynamics. Formally, our problem becomes

$$\min J = J_1 + J_2 \tag{7}$$

$$s.t. \begin{cases} C_1 : dx_k(t) = v_k(t)dt, \\ C_2 : x_k(0) = x_{k0}, \end{cases}$$

where C_1 describes the kinematic constraint for all agents, and C_2 is the the initial condition regarding agents' positions.

V. Mean-Field Game Approach

A. MFG framework

With the enormous number of interactions among agents in real-world systems, solving the previous problem becomes extremely challenging and costly. The mean-field theory specifies a very efficient way to deal with a wide variety of situations where there are too many particles contributes to the dynamics or equilibrium by modeling the interactions between all particles through constructing a good approximation of the situation by introducing a "mean-field term" m as a mediator for describing inter-particle interactions.

Mean field games study the interaction of individual agents with others by modeling the interaction and system dynamics through two coupled partial differential equations: the Hamilton-Jacobi-Bellman (HJB) equation and the Fokker-Plank-Kolmogorov (FPK) equation, where the HJB equation is the optimal condition for the system and FPK equation studies the system dynamics. The goal of MFG is to find out the Nash equilibria in games modeled by controlled stochastic dynamical systems that involve a great number of asymptotically negligible players.

A mean field game consist of large number of homogeneous agents $\mathbb{A} = \{a_1, a_2, ..., a_k\}$ and want to derive an optimal control $\mathbb{V} = \{\nu_1, \nu_2, ..., \nu_k\}$. The cost for agent a_k can be formulated as

$$J_k(\nu(t)) = \mathbb{E}\Big[\int_{t_0}^{T} C(\nu(t), \rho(t))dt + U(\rho_T, \rho(T)) \Big], \tag{8}$$

where $\rho(t)$ is the mean-field term at time t, $C(\nu(t), \rho(t))$ measures the cost of moving at a given input and $U(\rho_T, \rho(T))$ determines the terminal cost.

MFG first defines the value function

$$\phi(t,x) = \inf_{\nu(t)} \left\{ \mathbb{E}[\int_{t_0}^{T} C(\nu(t), \rho(t))dt + U(\rho_T, \rho(T))] \right\}. \quad (9)$$

The optimal control $\nu(t)$ can be derived by solving the following partial differential equations (PDEs).

$$-\partial_t \phi - \beta \Delta \phi + H(\nabla_x \phi) = 0, \quad (10)$$

$$-\partial_t \rho + \beta \Delta \rho + \nabla_x \cdot (\rho H'(\nabla_x \phi)) = 0, \quad (11)$$

$$\rho(0,x) = \rho_0(x), \rho(T,x) = \rho_T(x), \quad (12)$$

where $\beta := \frac{1}{2}\sigma^2$ is the viscosity term, and the Hamiltonian function H is a convex function with respect to ϕ.

B. Problem reformulation based on the MFG framework

As we try to formulate our system cost in an individual perspective, we find that accurately tracking the state of every object in a large-scale system is expensive and not practical. Thus, we try to simulate the system dynamics by applying MFG theory. The key technique of MFG that approximate the states of agents with the density field of the agent groups comes across our definition of spatial correlation model. We can model each user as an agent, and regard the target camera distribution for a certain camera as the approximation to the mentioned density field. The transformation is valid because the correlation indeed is the outcome of inter-user interactions and the probability of the user leaving one camera and entering any other cameras (exit included) also sums up to one.

Since in practice, users with location x are captured by the cameras and become an object in the cameras' frames. Each camera also covers a wide region, rather than a single location point. We thus define the agent states $\hat{\mathbb{X}} = \{\hat{x}_1, \hat{x}_2, ..., \hat{x}_n\}$, adapted from $\mathbb{X}$, where n is the number of cameras in the system. Every $\hat{x}_i \in \hat{\mathbb{X}}$ represents the cover range of camera i's sensing area. Our problem related to tracking the users' location thus can be transformed to tracking of shown camera $\hat{x}_i$.

With the introduced framework, we can define the running cost and termination cost in our MFG as follows:

Running cost. We formulate the running cost similar to (1). However, when the amount of agents is too large, checking all of their destinations and measuring their speed is expensive in cost, so we can approximate their next shown-up camera by the spatial correlation $\rho(t, \hat{x})$ that served as the mean-field term here. Also, the speed of the agents at time t can be approximate by current location $\hat{x_i}$ as $v(t, \hat{x}_i)$, i.e.,

$$\begin{aligned} \bar{J_3} &= \alpha_1 \int_0^T \mathbb{E}_{\hat{x}(t) \backsim \rho(t)} ||v(t,\hat{x})||_2^2 dt \\ &= \alpha_1 \int_0^T \int_\Omega \rho(t,\hat{x}) ||v(t,\hat{x})||_2^2 dt d\hat{x} \\ &= \alpha_1 \int_0^T \int_\Omega \frac{||m(t,\hat{x})||_2^2}{\rho(t,\hat{x})} dt d\hat{x}, \end{aligned} \quad (13)$$

$$m\,(t,\hat{x}) = \rho(t,\hat{x}) \times v(t,\hat{x}). \quad (14)$$

Here the $m(t,\hat{x})$ is the momentum of agents flow at camera $\hat{x}$ at time t, Ω is the cover range of camera sensing area. And T is the length of the inspection time range.

Termination cost. The termination cost becomes the sum of distance of agents between their actual target camera and a predicted one at time T.

$$J_4 = \alpha_2 \frac{1}{K} \sum_{k=1}^{K} dis(\hat{x}_k(T), y_k(T)), \quad (15)$$

where $y_k(t)$ is the real destination camera for agent k.

The deficiencies emerge when the number of agents is enormous and costly to keep track of individually. Instead, we can use the mean-field term $\rho(t,x)$, which is the distribution of target cameras for camera x. Thus, the cost can be conveniently determined by the distance between our prediction distribution at time T and the actual distribution. In order to make our model accurate and instructive, our terminal cost function imposes more severe punishment as the deviation is larger. We use the Kullback-Leibler (KL) divergence to measure the distance between distributions.

$$\bar{J_4} = \alpha_2 \int_\Omega \rho(T,\hat{x}) log(\frac{\rho(T,\hat{x})}{\rho_{T,\hat{x}}}) d\hat{x}, \quad (16)$$

where α_2 is the significance coefficient, $\rho(T,x)$ is the predicted correlation at time T and ρ_T is the true correlation. n is the number of cameras in this system.

With the above-defined running cost and termination cost, we have our overall combined form cost function $\bar{J}$

$$\bar{J} = \bar{J_3} + \bar{J_4}. \quad (17)$$

Constraint function. Here, we define the constraint function in our MFG setting with the help of the FPK equation. The system constraints mainly consider about the conservation problems because the distribution of the agents must sum to one. Assume we know the velocity field v here and the initial distribution $\rho(0,\hat{x})$ is given, we can model the process of the distribution evolution as time goes forward by solving the FPK and end up with the advection equation which is the PDEs that governs the motion of a conserved scalar field as it is advected by a known velocity vector field.

$$\partial_t \rho(t,\hat{x}) + \nabla \cdot (\rho v)(t,\hat{x}) = 0, \quad (18)$$

$$\partial_t \rho(t,\hat{x}) + \nabla \cdot m(t,\hat{x}) = 0, \quad (19)$$

where $\nabla \cdot m(t,\hat{x})$ can be seem as the the dynamics moving out of position $\hat{x}$ at time t.

Correspondingly, we present our cost minimization problem with the help of mean-field term as follows:

$$\begin{aligned} &\min \bar{J} = \bar{J_3} + \bar{J_4} \\ &s.t. \quad \partial_t \rho(t,\hat{x}) + \nabla \cdot m(t,\hat{x}) = 0, \end{aligned} \quad (20)$$

where the boundary condition for distribution ρ is known as ρ_0 and ρ_T. The objective means that in order to gain the optimal transport from initial spatial correlation to terminal one, the

system needs to minimize the running cost and termination cost as possible. The constraint means that the changing rate of target camera distribution is equal to dynamics moving out of the current camera at any position.

C. A G-prox Primal-Dual Solution

The transformed tracking cost minimization problem consists of two coupled PDE equations constrained with a partial differential equation and boundary conditions, which is very complex to solve. There is no closed-form solution to such a complex problem, and only numerical algorithms [14], [15] or learning-based algorithms, e.g., the Generative Adversarial Network (GAN), exist. However, the convergence of the existing numerical algorithms is usually $\mathcal{O}(n^2)$ or $\mathcal{O}(n^3)$ (n is the grid size). The training time of learning-based algorithms can be long(usually several hours) and requires a substantial amount of data. Therefore, we adopts a G-prox PDHG algorithm (PDHG) introduced in [16], whose complexity is linear with the number of grids, and convergence rate is independent of grid sizes. PDHG converts a minimization problem into a saddle point problem so that we can find the optimal solution easier. In the implementation, we discretize the coverage of camera system Ω into a $N_x \times N_y$ grids and overall video time span T into N_T points. Our PDHG thus is an iterative algorithm that updates $(\rho^n_{i,j})^{k+1}$, $(m^n_{i,j})^{k+1}$, $\rho(T,x)^{k+1}_{i,j}$, $(\Phi^n_{i,j})^{k+1}$ and $(\lambda_{i,j})^{k+1}$ in every iteration k for all t, i and j according to the defined updating policy. Thus, the computational complexity of every iteration is $\mathcal{O}(N_x \times N_y \times N_t)$, which is linear to the total number of grids in our MFG problems and independent to the number of agents in the system. Furthermore, according to [16], the step sizes of our master problem and dual problem are set to satisfy the convergence condition. Namely, if there exist an optimal solution $(\rho^n_{i,j})^*$ in our problem, then the numerical solution we get will converge to that solution.

VI. Performance Evaluation

A. Experiment setup

Dataset. We use Market-1501 dataset [17] to simulate a real-world camera system in our experiment. This dataset was captured in front of a supermarket on the campus of Tsinghua University by 5 high definition cameras and 1 standard definition camera. In total, 1501 pedestrians, 32,668 detected pedestrians are captured. We randomly select 750 objects as the queries for the testing set.

Parameter setting. The default time granularity $N_t = 14$. We set $T_{th} = 0.1$ and $S_{th} = 0.1$ in our experiments.

Benchmarks. We compare our MFG approach with four other benchmarks.

- Global search (Global): When the query instance leaves the current camera, searching jobs are triggered on all other edge cameras from now on until the target is found or the searching time exceeds the limits.
- Static correlation (Static): When the query instance leaves the current camera, searching jobs are triggered on those spatial-correlated cameras based on the static correlation model.

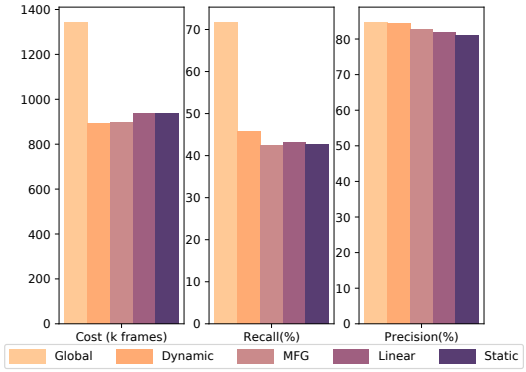


Fig. 6. Results of MFG generated model vs. other benchmarks

- Linear adapted correlation(Linear): Linear adapted correlation approach enhanced the static correlation approach by assuming the linear growth rate from the correlation of first period to the correlation of last period.
- Ground truth dynamic correlation (Dynamic): The correlation model is generated from all objects of each period, which reflects the actual movement of objects. This hindsight-based approach sets an upper limit for all approaches.

Metrics. We define the following three metrics to examine the effectiveness of different searching algorithms.

- Computational cost: Summation of all the frames being processed at the edge cameras for all queries.
- Precision(%): Ratio of number of correctly matched objects to number of queries that retrieved objects.
- Recall(%): Ratio of number of correctly matched objects to number of total queries.

B. Evaluation results

Fig. 6 shows the results of comparison among our method and all other bench marks.

Computational cost. The global searching method examined 1,343,345 frames to complete the cost. In comparison, our MFG methods took only 896,273 frames, which is 36%, 4.2%, and 4.4% less than the global method, linear, and static method, respectively. Compared with the posterior dynamic method that builds upon extensive profiling, our method consumes almost the same amount of frames with drastically less generation cost. This indicates that the cost functions and constraint functions defined under our setting for MFG are valid and practical; Our model successfully captures the system dynamics with only the initial and terminal correlation.

Recall and precision. We noticed that all benchmark methods as well as our MFG method have a certain decline in recall rate compared to the global searching method. Such decline is reasonable because as all these methods excluded those frames in uncorrelated cameras and uncorrelated time periods, its ability to retrieve objects from all video sequences weakens. Recall rates for our MFG methods only drop 3%

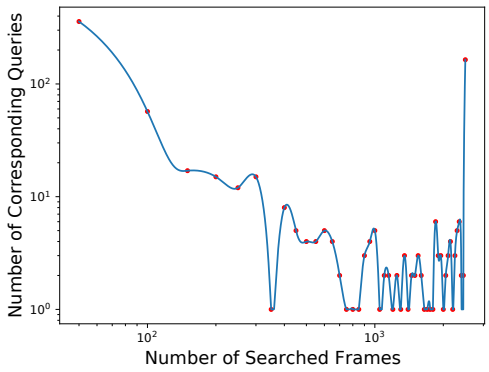


Fig. 7. Distribution of searching frames number for query instances

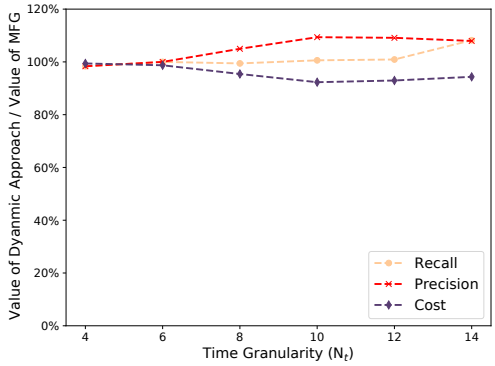


Fig. 9. Comparison of the dynamic approach to the MFG approach

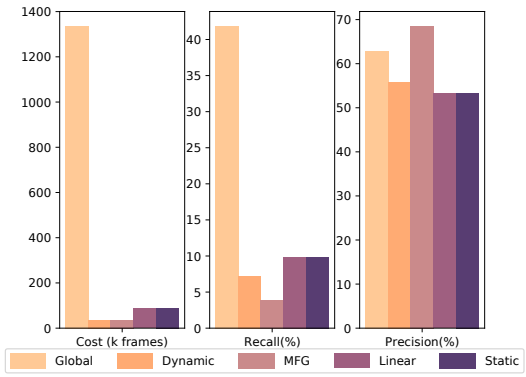


Fig. 8. Results of MFG generated model vs. other benchmarks in queries that need to process more than 100 frames

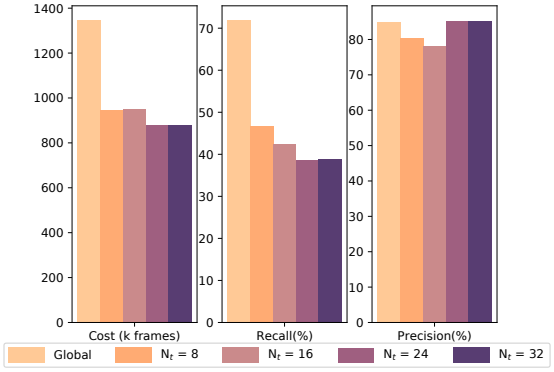


Fig. 10. MFG approaches under different time granularity N_t

compared to the ground truth dynamic method. Our method has less than a 2% drop in precision compared to the ground truth dynamic method and outperforms the linear and static approach by 1% and 2%.

Heavy hitter analysis. While some may argue that the saving in the computational cost of our model is not enough for compensation of drops on recall and precision, our methods show its aggressiveness when the frames needed to be processed in order to match the object is relatively large. We examine the distribution of processed frames for the global searching method among 750 queries in Fig. 7. As can be seen from the figure, more than half of the quires take more than 100 frames to search for the global searching method. Studying these quires with a larger searching range is meaningful since these queries take up 99.3% of all the computational cost. So we perform identify these heavy hitter queries requiring more than 100 frames by the global searching method and reveal the savings of our approach for these heavy hitters in Fig. 8. As can be seen from Fig.8, the global searching method processed nearly 40× more frames than our method. The static and linear adapted method processed almost 3× more frames than our method. Surprisingly, The precision of our method exceeds the global searching by 6%, showing its ability to cross out irrelevant frames.

Sensitivity of time granularity N_t. In Fig.9, we show the comparison results of the dynamic and MFG methods at different time granularities N_t. With the results, we find that, in terms of each metric, the searching tasks guided by the prediction models we obtained through MFG are able to obtain a performance similar to that of the global generated dynamic models. For all three metrics, the ratio for the value of the dynamic approach to the value of the MFG approach is between 92.3% to 109.3%. And the greatest deviation from the dynamic method is 8.6%, 5.9%, 7.3%, for precision, recall, and precision separately. This indicates that the models predicted by our MFG method simulate the real situation well. Obtaining such a realistic model with a significant reduction in model generation cost demonstrates the advantages of our approach.

Another advantage of the MFG model is its ability to generate a more fine-grained model than the global dynamic model. Fig.10 compares the performance of the MFG-generated correlation model of different time granularity. From the results, we find that tracking guided by model of $N_t = 32$ or $N_t = 24$ search around 600k frames less than that of $N_t = 8$ and $N_t = 16$ with 5% and 7% increase in precision separately. We can find that the searching cost is apparently reduced by using the correlation between cameras.

VII. Related Work

Current cross-camera video analytics studies can be categorized into two types: reducing the individual cost at the edge camera level [5] [6] [7] and reducing overall overhead through system coordination [3], [8], [9]. Focus in [5] uses cheap CNN at the edge and searches only top K matches to lower respond latency, which compensates for the lower accuracy. Optasia [3] performs automatic parallelism with respect to multiple tasks from a single camera as well as sources from all cameras. Although the above methods provide remarkable efficient solutions, all of them neglect the inherent correlations between edge cameras. One recent work in [9] exploits cross-cameras correlation with respect to both spatial and temporal to reduce tracking cost by filtering out uncorrelated frames. However, it fails to conduct complete analysis on correlation and just adopts a static model. There are some other works focus on increasing the tracking accuracy in mutli-target multi-camera tracking tasks by integrating correlation information in their system flow [10], [11], [12], but these works focus on precision improving and shed little light on cost reduction in cross-camera video analytics, which is the focus of this paper.

As a relatively new concept in game theory, MFG theory, proposed by Lasry and Lions in [18] and Caines, Huang, and Malhame in [19], has been applied in many applications in communications and networked systems. While conventional optimization models and game theory models struggle with systems of a large number of agents, MFG theory simplifies the problem by restating the problem as an interaction of each agent with the mass of others by introducing a concept of the mean-field term. Multiple applications have been studied, like UAV control [20], power control [21], price determination in the Internet markets [22],etc. To the best of our knowledge, we conduct the first work that investigates cross-camera video analytics using MFG theory.

VIII. Conclusion

Cross-camera video analytics is a key application in large scale camera systems that tracks the association of objects across different cameras. However, searching for correlated objects across cameras is extremely time and computation costly. Existing efforts on leveraging spatial and temporal correlations to help the search relies on the static information, which is inaccurate and suboptimal. In this paper, we identify the existence of the dynamic correlation and reveal the potential benefit in leveraging the dynamic correlation. We then present the first work to apply MFG to the emerging cross-camera video analytics applications. Our MFG approach captures the complex dynamics of the real world system, and can generate fine grained correlation models with a small number of samples. Extensive experiments on real-world dataset reveal that our method significantly reduces the number of frames to be processed with almost equal precision performance.

References

[1] T. Ricker, "The us, like china, has about one surveillance camera for every four people, says report," Dec. 2019. [Online]. Available: https://www.theverge.com/2019/12/9/21002515/surveillance-cameras-globally-us-china-amount-citizens

[2] E. Ristani and C. Tomasi, "Features for multi-target multi-camera tracking and re-identification," in *Proc. IEEE CVPR*, Mar. 2018, pp. 6036–6046.

[3] Y. Lu, A. Chowdhery, and S. Kandula, "Optasia: A relational platform for efficient large-scale video analytics," in *Proc. ACM Symposium on Cloud Computing*, Oct. 2016, p. 57–70.

[4] M. Satyanarayanan, "Mobile computing: the next decade," *Mobile Computing and Communications Review*, vol. 15, pp. 2–10, Jan. 2011.

[5] K. Hsieh, G. Ananthanarayanan, P. Bodik, S. Venkataraman, P. Bahl, M. Philipose, P. B. Gibbons, and O. Mutlu, "Focus: Querying large video datasets with low latency and low cost," in *Proc. USENIX OSDI*, Carlsbad, CA, Oct. 2018, pp. 269–286.

[6] D. Kang, J. Emmons, F. Abuzaid, P. Bailis, and M. Zaharia, "Noscope: Optimizing neural network queries over video at scale," *Proc. VLDB Endowment*, vol. 10, no. 11, Aug. 2017.

[7] J. Jiang, G. Ananthanarayanan, P. Bodík, S. Sen, and I. Stoica, "Chameleon: Scalable adaptation of video analytics," in *Proc. ACM SIGCOMM*, Aug. 2018.

[8] H. Zhang, G. Ananthanarayanan, P. Bodik, M. Philipose, P. Bahl, and M. J. Freedman, "Live video analytics at scale with approximation and Delay-Tolerance," in *Proc. USENIX NSDI*, Boston, MA, Mar. 2017, pp. 377–392.

[9] S. Jain, X. Zhang, Y. Zhou, G. Ananthanarayanan, J. Jiang, Y. Shu, V. Bahl, and J. Gonzalez, "Spatula: Efficient cross-camera video analytics on large camera networks," in *Proc. ACM/IEEE Symposium on Edge Computing (SEC)*, Nov. 2020.

[10] M. Wu, Y. Qian, C. Wang, and M. Yang, "A multi-camera vehicle tracking system based on city-scale vehicle re-id and spatial-temporal information," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2021, pp. 4072–4081.

[11] A. Specker, D. Stadler, L. Florin, and J. Beyerer, "An occlusion-aware multi-target multi-camera tracking system," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2021, pp. 4168–4177.

[12] P. Ren, K. Lu, Y. Yang, Y. Yang, G. Sun, W. Wang, G. Wang, J. Cao, Z. Zhao, and W. Liu, "Multi-camera vehicle tracking system based on spatial-temporal filtering," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2021, pp. 4208–4214.

[13] L. Zheng, L. Shen, L. Tian, S. Wang, J. Wang, and Q. Tian, "Scalable person re-identification: A benchmark," in *Proc. IEEE ICCV*, 2015.

[14] T. E. Duncan and H. Tembine, "Linear–quadratic mean-field-type games: A direct method," *Games*, vol. 9, no. 1, p. 7, Feb. 2018.

[15] A. T. Lin, S. W. Fung, W. Li, L. Nurbekyan, and S. J. Osher, "Apac-net: Alternating the population and agent control via two neural networks to solve high-dimensional stochastic mean field games," *arXiv preprint arXiv:2002.10113*, 2020.

[16] M. Jacobs, F. Léger, W. Li, and S. Osher, "Solving large-scale optimization problems with a convergence rate independent of grid size," *SIAM Journal on Numerical Analysis*, vol. 57, no. 3, pp. 1100–1123, May 2019.

[17] L. Zheng, L. Shen, L. Tian, S. Wang, J. Wang, and Q. Tian, "Scalable person re-identification: A benchmark," in *Proc. IEEE ICCV*, Dec. 2015, pp. 1116–1124.

[18] J.-M. Lasry and P.-L. Lions, "Mean field games," *Japanese journal of mathematics*, vol. 2, no. 1, pp. 229–260, 2007.

[19] P. E. Caines, M. Huang, and R. P. Malhamé, "Mean field games." 2015.

[20] Y. Kang, S. Liu, H. Zhang, Z. Han, S. Osher, and H. V. Poor, "Task selection and collision-free route planning for mobile crowd sensing using multi-population mean-field games," *IEEE Transactions on Green Communications and Networking*, Jan. 2021.

[21] C. Yang, J. Li, P. Semasinghe, E. Hossain, S. M. Perlaza, and Z. Han, "Distributed interference and energy-aware power control for ultra-dense d2d networks: A mean field game," *IEEE Transactions on Wireless Communications*, vol. 16, no. 2, pp. 1205–1217, Dec. 2016.

[22] V. Reddyvari Raja, V. Ramaswamy, S. Shakkottai, and V. Subramanian, "Mean field equilibria of pricing games in internet marketplaces," *SIGMETRICS Perform. Eval. Rev.*, vol. 44, no. 1, p. 387–388, jun 2016.