Graph-Based Topic-Focused Retrieval in Distributed Camera Network

Jiejun Xu, *Member, IEEE*, Vignesh Jagadeesh, *Member, IEEE*, Zefeng Ni, *Member, IEEE*,
Santhoshkumar Sunderrajan, *Member, IEEE*, and B. S. Manjunath, *Fellow, IEEE*

***Abstract*—Wide-area wireless camera networks are being increasingly deployed in many urban scenarios. The large amount of data generated from these cameras pose significant information processing challenges. In this work, we focus on representation, search and retrieval of moving objects in the scene, with emphasis on local camera node video analysis. We develop a graph model that captures the relationships among objects without the need to identify global trajectories. Specifically, two types of edges are defined in the graph: object edges linking the same object across the whole network and context edges linking different objects within a spatial-temporal proximity. We propose a manifold ranking method with a greedy diversification step to order the relevant items based on similarity as well as diversity within the database. Detailed experimental results using video data from a 10-camera network covering bike paths are presented.**

***Index Terms*—Distributed camera network, diverse and relevant ranking, graph-based modeling, information search and retrieval.**

I. INTRODUCTION

WITH the rapid development of sensor technologies, distributed cameras are becoming prevalent as part of the urban infrastructures covering large areas for a variety of applications. However, managing and analyzing the massive amount of data from distributed cameras is a very challenging task. Traditional approaches usually follow a centralized model which streams all videos to a central server for further processing. This approach is not scalable as it puts significant demand on communication bandwidth and power. On the other hand, cameras are becoming powerful in terms of local processing power and storage capacity. This leads to the growing trend of decentralized processing for camera sensor networks. In such a set up, raw videos are processed at individual sensor nodes, and only information extracted from the observed scene is transmitted to the central server. Though this distributed paradigm provides scalability and robustness, it also poses challenges on global understanding of the scene. In particular, the problem of searching and retrieving objects of interest across distributed camera nodes assumes significance. In such a distributed setting, individual cameras do not have an aggregated view of the entire scene being observed, and the central node does not have direct access to original videos. Thus the full potential of a wide-area camera network data can only be realized by striking an appropriate balance between local, distributed processing of the large volumes of video data coupled with a centralized aggregation of information.

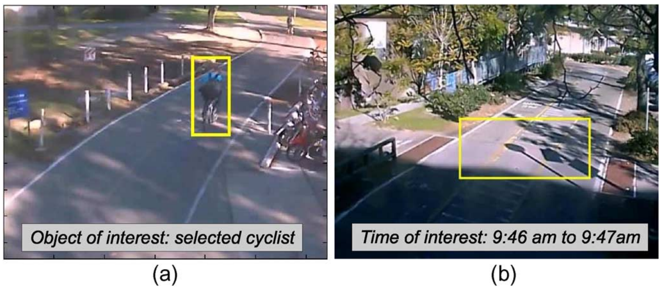

Object of interest: selected cyclist
Time of interest: 9:46 am to 9:47am
(a)
(b)

Fig. 1. Sample queries. Yellow rectangle on the left image indicates an Object Query; Yellow rectangle on the right image indicates a Spatial-Temporal Query.

Our experimental setup consists of a wide-area camera network with a wireless communication backend (see Fig. 2 and Fig. 8). We assume raw videos are archived at remote camera nodes and each node has limited processing power for simple video analysis such as motion detection and tracking. We envision the following two scenarios:

- **Object Query** (see Fig. 1(a)): A user initiates query of the form, "FIND all objects *related* to the object instance at region C FROM camera 1 at time 9:32:41.3am". A special case for this query scenario resembles to the classic problem of object re-identification, i.e. find the instances of the same object in all camera views.
- **Spatial-Temporal Query** (see Fig. 1(b)): A user initiates spatio-temporally constrained query, for example, "FIND object instances *related* to region A FROM camera 1 OR region B FROM camera 4 between time 9:30am and 9:35am".

Prior research has focused on methods that can track all observed objects across the entire camera network [1]–[3]. Such methods generally require solving a data-association problem based on some matching criterion. However, matching can be unreliable, especially in real world scenarios due to illumination changes, variation in poses, and possible occlusion, and it is difficult to assure that the inferred match is the right match. In this work, instead of precise identification of the global trajectories for every object visible in the network, we propose a global graph to model the underlying relationships of all local camera observations without explicit association.

Manuscript received December 29, 2012; revised April 12, 2013; accepted June 01, 2013. Date of publication September 05, 2013; date of current version November 13, 2013. This work was supported in part by the following awards: ONR #N00014-10-1-0478 and #N00014-12-1-0503. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Yen-Kuang Chen.

J. Xu is with the University of California, Santa Barbara, Computer Science Department, Santa Barbara, CA 93106-9650 USA.

V. Jagadeesh, Z. Ni, S. Sunderrajan, and B. S. Manjunath are with University of California, Santa Barbara, Electrical and Computer Engineering, Santa Barbara, CA 93106-9650 USA.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TMM.2013.2281019

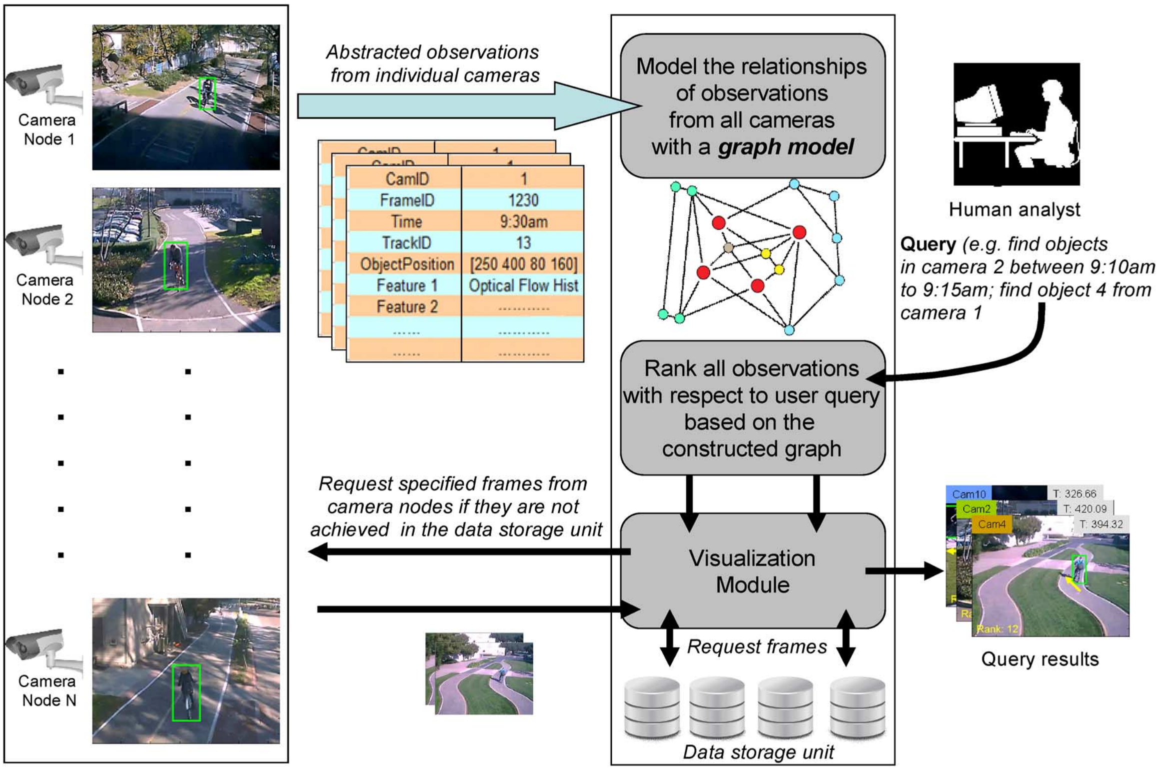

Camera Node 1
Camera Node 2
Camera Node N
Abstracted observations from individual cameras
CamID 1
FrameID 1230
Time 9:30am
TrackID 13
ObjectPosition [250 400 80 160]
Feature 1 Optical Flow Hist
Feature 2 ....
Model the relationships of observations from all cameras with a ***graph model***
Human analyst
Query *(e.g. find objects in camera 2 between 9:10am to 9:15am; find object 4 from camera 1*
Rank all observations with respect to user query based on the constructed graph
Request specified frames from camera nodes if they are not achieved in the data storage unit
Visualization Module
Request frames
Data storage unit
Cam10
Cam2
Cam4
T: 326.66
T: 420.09
T: 394.32
Rank: 12
Query results

Fig. 2. A conceptual architecture for information processing in a wide-area wireless camera network.

This global graph is used to find a set of "informative" and "diverse" snapshots with respect to user query. Fig. 2 gives an overview of the proposed system. Assuming a network of N distributed static cameras with embedded storage and computing power, each camera node independently detects and tracks moving objects within its field of view. For each frame with detected objects, the camera sends an abstracted record, comprising of an object's spatial, temporal and visual feature information, to a central server node. At the central node, a graph is built to model the relationships among the camera observations based on the received records. In particular, vertices in the graph represent fixed length tracklets of detected objects, which can be easily constructed by aggregating records. There are two types of linkages in the graph, object links and context links. The object links connect vertices (i.e. tracklets) which are from the same object. The context links connect different objects but appearing in nearby spatial or temporal context. Fig. 7 shows an example of how objects can be connected in a graph through assorted links.

Given a user query, the central node first performs ranking on the graph to identify a representative and concise set of tracklets. Subsequently the corresponding frames (snapshot) are retrieved to visualize the ranking results.

The key contributions of this work are:

- Design and implementation of an interactive search and retrieval system for a large distributed camera network.
- Modeling of distributed camera observations using a global graph, and provide a ranking mechanism to generate an informative and diverse set of results based on user queries.
- Relating observations across cameras with a set of linear regression functions modeling the time-delays of objects moving between cameras.

The graph-based retrieval approach for camera network was first introduced in our previous work [4]. The current work substantially differs and improves over the previous work in the following aspects: 1) a more expressive graph model to capture the spatial-temporal relationship among objects, 2) a more effective motion-based feature to characterize objects, and 3) an extensive evaluation on a large and densely labeled dataset. The rest of the paper is organized as follows. Section II describes related works on object re-identification and graph-based method for visual retrieval. Section III describes the details of our retrieval system, which includes local processing in distributed cameras nodes, and graph modeling and ranking of camera observations in central nodes. Finally, Section V presents both quantitative and qualitative evaluation of the system using real world data collected from our 10-node camera network. Section VI concludes this paper with some discussions.

II. RELATED WORK

This work is related to the problem of object reacquisition or re-identification in multiple cameras. In [5], [6], similar systems with distributed cameras are proposed, with a server collecting camera observations and assigning unique global object ID based on object's estimated location and/or color appearance. To deal with appearance variations across views, much work has been done on finding the best matching criterion. Examples of such works include the joint motion and appearance model [1], low-dimension subspace learning of brightness transfer functions [2], symmetry-driven accumulation of local features [7], statistic of spatial-temporal optical flow [8], probabilistic relative distance comparison [9], the shared set of haar-features [10], and the discriminative human appearance signature based on mean Riemannian Covariance Grid in [11]. All these methods use the pair-wise comparison of measurements from different camera views. This way of direct comparison might suffer when measurements (object detection and tracking) from individual cameras are noisy. A more effective way of relating observations from different cameras is to treat them collectively, instead of pair-wise similarity comparison. In [3], likely paths of objects are assembled using Bayesian estimates over all camera observations. In [12], a distributed system for supporting spatio-temporal analysis on large-scale camera networks is proposed. In particular, signatures (face features) generated at local camera nodes are matched to a signature database. These matching results are then used to guide the state update for different person. However, all these methods require perfect detection and tracking from individual cameras. In contrast, we propose to utilize a graph model to represent the underlying relationships among camera observations *without explicit association of objects across camera views*. With the graph representation, we then propose to cast the retrieval problem as a graph ranking problem by identifying *representative* snap shots that could contain the observations requested by the user.

This work is also closely related to graph-based image retrieval and ranking [13], [14]. The basic idea of graph-based ranking is to first construct a weighted graph, where nodes are the data and edges reflect their similarities. Then the query node will be selected and assigned an initial ranking score. This score will be spread to the rest of the graph nodes based on the intrinsic structure of the weighted graph. The final score associated with each node indicate the relevance of corresponding node to the query. Graph-based ranking has been applied to many problems [15]–[17], and was first adopted to the domain of image retrieval in [18]. Several follow up works have been developed to further improve the performance in terms of both efficiency [19] and scalability [20]. In addition, graph-based ranking can also be modeled with a more explicit random walk theory [21]. Given a graph and a query node, the ranking scores of the other nodes can be computed as the likelihood that they will be visited by a random walker starting from the query node. Image retrieval is then formulated as a random walk process. For example, visual ranking algorithms are proposed in [22], [23] to apply PageRank [24] as a solution to large scale image search. By introducing multiple types of edges, graph-based image retrieval can be improved by fusing multiple visual feature modalities and exploiting their mutual reinforcement [25].

Our proposed system aims to extract a concise visual summary according to a given topic (e.g. spatial-temporal region of interest), specified by a user query. In [26], an automatic procedure to construct a compact synthesized summary from a video sequence is proposed. The method extracts salient regions of interest (ROIs) from selected representative images and seamlessly arrange ROIs on a given canvas with the video's temporal structure preserved. Similar methods are also proposed in [27]–[30]. However, the visual summary generated from these methods are not constrained by any query topic. Instead they are highlights of the video created with some visual saliency criteria. In addition, these methods focus on extracting a summary from a single video sequence, while we are aiming to extract a summary from multiple (possibly) correlated video sequences from different views of the camera network. Recently, there are a few methods proposed to deal with multi-view videos for visual summary [31], [32]. Both methods try to find generic optimization criteria to remove redundancy across views. For example, if a person appears in several camera views, the algorithms will attempt to compress the information by showing the person in only one view and assume the rest of the information can be inferred.

III. A GRAPH MODEL FOR CAMERA NETWORK DATA

We propose to abstract the entire camera network as a graph $G=(V,E)$. The nodes $v_i \in V, 1 \leq i \leq N$ comprise a total of N tracklets, extracted across multiple cameras. Tracklets are represented by feature descriptors extracted from temporally adjacent bounding boxes obtained by tracking an object. The edge set $e \in E$ comprise edges introduced between graph nodes v_i and v_j, which signify their relationship, i.e. tracklets that belong to the same object or tracklets that belong to different objects but co-occur in a spatial-temporal proximity. Additional tracklets can be appended to the graph incrementally, and the graph can grow over time. For simplicity, we focus on a portion of the time-evolving graph with a fixed number of tracklets. We begin by motivating the design of node attributes obtained by distributed processing. Subsequently, we discuss the procedure by which a centralized server aggregates the information transmitted by all network cameras and induces graph edges based on spatio-temporal and semantic constraints.

A. Computing Node Attributes by Distributed Processing

1) Detection and Tracking: Each camera node detects moving objects and tracks them on the image plane. In a static camera network, objects can be detected by modeling background and identifying moving foreground. Connected foreground pixels are combined together to form foreground blobs, which are then tracked by a mean shift algorithm [33]. In our set up, each object is represented with a rectangular blob. To address the problem of scale variations, we utilize the general mean-shift blob tracking algorithm proposed in [34].

For each tracked object, a unique track ID is assigned. For each frame processed by the camera, a record is generated for each detected/tracked object and sent to the central node. Each observation record includes information such as camera ID, time, object blob position on the image plane and more importantly, an image feature characterizing the tracked object, see Section III-A-2.

Tracklets form the basic unit of representation and are generated by dividing a track into uniform non-overlapping intervals. In our implementation, each tracklet is fixed to contain 4 frames. Thus if a track extends for 80 frames, it is broken into 20 tracklets, each of size 4 frames. In addition, a parent track gives rise to child tracklets. We use the notation $\mathcal{T}(i)$ to denote the mapping from a child tracklet i to its parent track.

2) Object Feature Extraction: In [4], we utilized a simple appearance based feature, a 16-bin Hue histogram, to represent a detected object. However, this feature is not robust to variations such as illumination changes, object pose and possible occlusions, and all these are common issues in a outdoor camera network. On the other hand, the motion based cue described below is less sensitive to these above imaging conditions and is used in our experiments.

Speed and direction of motion derived from motion cues can serve as important indicators of time-delay estimation (i.e., time an object takes to go from one camera to another) in our fixed camera setting, as topology of cameras are known *a priori*. Based on the estimated time-delay, one can make a reasonable inference on whether or not they are the same object. To utilize motion cues, we compute dense optical flow fields between adjacent frames using a standard variational optical flow method followed by temporal smoothing. Given a pair of images from adjacent frames I_t and I_{t+1}, the corresponding optical flow is given by a vector field $Flow_{t,t+1} = \{m_r^t, \theta_r^t\}$, where $\{m_r^t, \theta_r^t\}$ are magnitude and direction of flow vectors at pixel r in frame t. After the temporal smoothing, $\hat{Flow}_{t,t+1} = \sum_{-2\le k\le 2}(Flow_{t-k,t-k+1}/5) = \{\hat{m}_r^t, \hat{\theta}_r^t\}$ to remove artifacts, a d-dimensional histogram of optical flow h_t^f is computed within a bounding box containing the tracked object in frame t utilizing the flow field $\hat{Flow}_{t,t+1}$. The entries of the flow histogram are defined as follows,

$$h_t^f(j) = \sum_r \hat{m_r^t}\mathcal{I}\left(\frac{360(j-1)}{d} \le \hat{\theta}_r^t \le \frac{360j}{d}\right),\ 1 \le j \le d, \tag{1}$$

where $\mathcal{I}$ is an indicator function. The dimension d of the histogram is fixed to 8 in our experiments (see Fig. 3).

In order to reduce computational load, each tracklet is only associated with one motion histogram, which is obtained by averaging the flow histograms between the beginning (t_{start}) and terminating frames (t_{end}) of a tracklet. Recall that a tracklet is of fixed length of 4 frames, thus the motion histogram for tracklet i is defined as: $h_i^f = (1/4)\sum_{t_{start}\le l\le t_{end}} h_l^f$. Subsequently the direction θ_i and speed α_i (i.e. optical flow magnitude) of the tracklet i is estimated as:

$$\begin{aligned}\theta_i &= \arg\max_{j\in 1,2,\dots d} h_i^f(j)\\ \alpha_i &= h_t^f(\theta_i).\end{aligned} \tag{2}$$

As stated earlier, tracklets serve as the basic units in our work, and they are nodes in the global graph. The speed feature α_i will be used for relating tracklets across cameras through time-delay regression, as shown in Section III-B-3.

B. Graph Model for Aggregating Camera Observations

Once feature is extracted from a detected object as in (2), it will be encapsulated in an abstracted record and transmitted to the central node along with other meta data such as camera ID and time. In particular, we create a representation at the central node that emphasize the following two properties, **centrality** (i.e., one which cluster closely related observations) and **diversity** (i.e., covering as many distinct groups of observations as possible). For instance, when retrieving a particular object, it would be informative to retrieve instances when the object interacts with others. Similarly, it would be interesting to show the object with diverse context, e.g. different camera views.

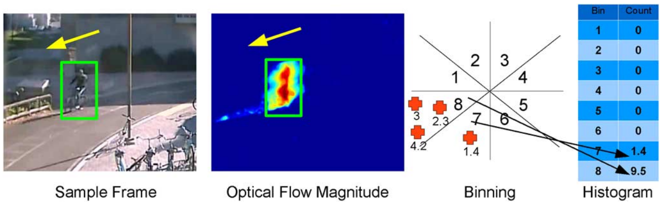


Fig. 3. Histogram of dense optical flow: In this example, the object is moving towards the left direction, thus only the bins 7,8 contain non-zero magnitude responses.

With this in mind, a global graph is constructed by introducing edges between tracklets to model the overall relationships between camera observations. Note that tracklets are extracted from multiple camera views. Let T_c be the set of N_c tracklets extracted from camera $c \in C$, where C is the set of all cameras. Let $i = p(c, j)$ be a lookup function that takes in a camera index c and tracklet index $j : 1 \le j \le N_c$ and maps to the global tracklet index $i : 1 \le i \le N$. Then,

$$\begin{aligned}N &= \sum_{c\in C} |T_c|,\\ T_c &= \left\{v_{p(c,1)}, v_{p(c,2)} \dots v_{p(c,N_c)}\right\}, \quad |T_c| = N_c.\end{aligned} \tag{3}$$

Let W denotes the $N \times N$ adjacency matrix of the global graph. The entries of W are constructed as described below. We consider two types of intra camera links—object links and contextual links.

1) Intra Camera Object Links: Consider $\{i, j\} \in T_c, c \in C$. Then, an object link is introduced between tracklets i and j if:

$$W^1_{p(c,i),p(c,j)} = \boldsymbol{\delta}_{\mathcal{T}(p(c,i)),\mathcal{T}(p(c,j))}, \tag{4}$$

where δ is a Kronecker delta function, and p is a function that maps the local tracklet index to the global index. $\mathcal{T}$ is a function to retrieve parent track. Basically if both tracklets share the same parent track, then a link is introduced.

2) Intra Camera Context Links: Here we consider capture contextual information around an object of interest. Edges are introduced between tracklet nodes that co-ccour in time within the same camera, but do not belong to the same object. Thus,

$$\begin{aligned}W^2_{p(c,i),p(c,j)} = \mathbf{U}&\left(th_s - \left|t_{p(c,i)} - t_{p(c,j)}\right|\right) \wedge\\ &\left(1 - \boldsymbol{\delta}_{\mathcal{T}(p(c,i)),\mathcal{T}(p(c,j))}\right).\end{aligned} \tag{5}$$

Here $\mathbf{U}$ is a step function which returns 1 if its input is greater than zero, and returns 0 otherwise. $t_{p(c,i)}$ and $t_{p(c,j)}$ correspond to the times at which two tracklets are extracted; th_s is a constant threshold controlling the temporal proximity of two tracklets.

3) Inter Camera Object Links: A key challenge in searching in distributed camera network is to relate objects in different camera view points. As mentioned earlier, methods based on matching appearance features are not very reliable for outdoor settings. Instead, we focus on estimating the time-delay between cameras using object reappearance. Robust object association

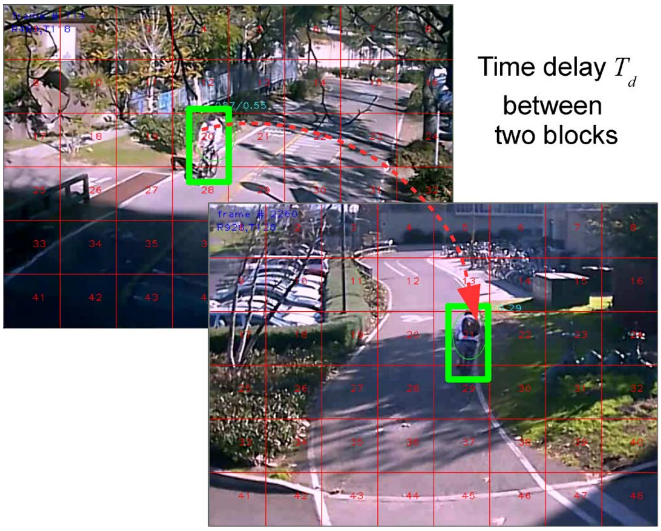


Fig. 4. Implicit linkage between camera blocks.

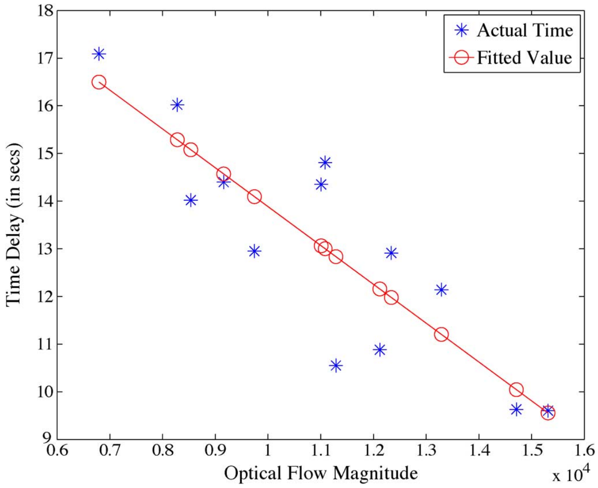


Fig. 5. Standard error of regression for time-delay estimation.

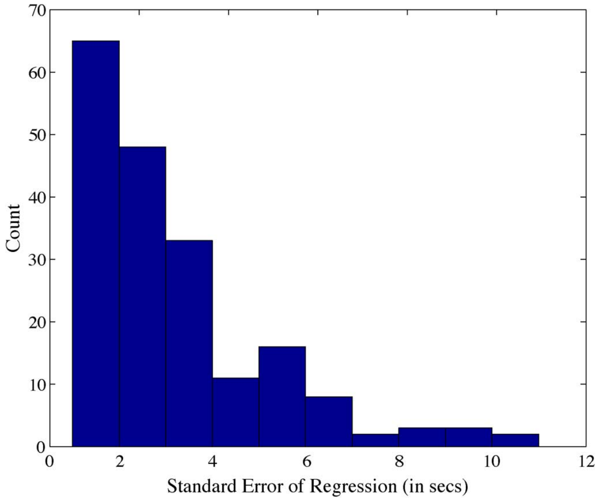


Fig. 6. Distribution of standard error of regression.

can be achieved by comparing the actual time stamps of the detected objects against the estimated time-delay in their corresponding blocks.

In order to estimate time-delay between camera nodes, we partition an image frame of a camera node into 48 non-overlapping blocks in a 8 × 6 grid as shown in Fig. 4. For a specific pair of blocks, we model the relationship between object speed α_i and time delay y_i using a vector $[\beta_1 \beta_0]$.

$$y_1 = [\alpha_1 \quad 1] \begin{bmatrix} \beta_1 \\ \beta_0 \end{bmatrix}$$

$$\cdots\cdots\cdots\cdot$$

$$\cdots\cdots\cdots\cdot$$

$$y_K = [\alpha_K \quad 1] \begin{bmatrix} \beta_1 \\ \beta_0 \end{bmatrix}$$

$$Y = \begin{bmatrix} y_1 \\ y_2 \\ \cdots \\ y_K \end{bmatrix}, \quad A = \begin{bmatrix} \alpha_1 & 1 \\ \alpha_2 & 1 \\ \cdots\cdots\cdots & \\ \alpha_K & 1 \end{bmatrix}, \quad \beta = \begin{bmatrix} \beta_1 \\ \beta_0 \end{bmatrix}$$

In the following discussion, we abbreviate a variable $X(b_1 = p, b_2 = q; c_1 = k, c_2 = l)$, modeling the pairwise relationships between blocks (p, q) in cameras (k, l) respectively, by $X_{pq;kl}$ to avoid notational clutter. From pre-annotated ground truths, a regressor $\beta_{pq;kl}$ is learnt between sub-block b_1 and b_2 belonging to cameras c_1 and c_2 respectively. We assume a simple linear regression model given by,

$$y_{pq;kl} = A_{pq;kl}\beta_{pq;kl} + \epsilon. \tag{6}$$

The variable A stores the observations, each of whose rows correspond to object speed, approximated by optical flow magnitudes. The target variable y refers to time delays. The following linear regression solution $\hat{\beta}_{pq;kl} = (A^T_{pq;kl}A_{pq;kl})^{-1}A_{pq;kl}y_{pq;kl}$ yields a solution to speed-to-time-delay mapping between blocks b_1 and b_2. The same procedure is repeated for all pairwise blocks among nearby camera nodes, see Fig. 4. This gives us a time-delay function $F_{pq;kl}$ with speed as its input parameter for each block pair. Note that block pairs are limited to local neighboring cameras only, as two far away blocks are likely to include noisy and false correlation. In addition, block pairs are directional, as one can go from one block to the other and vice versa. Thus $F_{pq;kl}$ is not necessary the same as $F_{qp;lk}$.

The difference between actual value y and estimated value $\hat{y}$ can be measured using Standard Error of Regression (SER). It can be used to learn a threshold function that yields a slack to the predictions. Fig. 5 illustrates a plot of regression function computed for the same two blocks as indicated in Fig. 4. As we can see from the spread of the data points, the average “mistake” made by the regression function is quite small.

To further understand the characteristics of regression estimation, we selected a subgroup of local cameras (Cam 1 to Cam 4 as shown in Fig. 8), and computed the time-delay regression for all the block pairs based on pre-annotated data. Distribution of residual errors between predicted time-delays and their actual values is shown in Fig. 6. Note that most of the residual error are quite small (usually less than 3 seconds). In addition, the number of block pairs are fairly well bounded because we only consider cameras in a local neighborhoods. In addition,

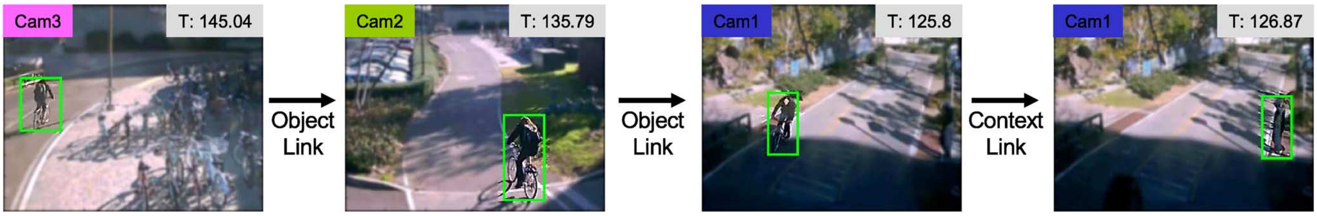


Fig. 7. An illustration on how distant nodes can be related together in the graph through combination of linkages. The first three snapshots highlight the same cyclist, while the last snapshot highlight a nearby pedestrian.

common object paths usually only cover a limited number of blocks in each camera view.

Without loss of generality, let us assume that $t_i < t_j$ and estimate the future time at which tracklet i would reach the camera and block where tracklet j was extracted from. This can be achieved from the learnt predictor as:

$$\hat{y} = F_{pq;kl}(\alpha_i) \tag{7}$$

Now we define the condition for an edge to exist between two tracklet nodes i and j as:

$$W^3_{i,j} = \mathbf{U}\left(th_o - |\hat{y} - (t_j - t_i)|\right). \tag{8}$$

Here i and j are assumed to be in the global tracklet index; th_o is a constant threshold controlling the difference between the estimated time-delay versus the actual time-delay between two tracklets.

The final adjacency matrix W is a union of the matrices constructed as in (4), (5) and (8). Essentially, it is a linear combination of all three graphs with uniform weights.

IV. QUERYING THE CAMERA NETWORK GRAPH

So far we have constructed a graph representation, with globally identified tracklets and connections between them represented with the W entries. Next we will transform a user query, such as the ones in Fig. 1, into nodes in the graph representation, and answer it by ranking the rest of the graph nodes based on their relevance to the query.

A. Query Instantiation

To instantiate a user query, the system first initializes a query vector $r \in R^{N\times 1}$ whose elements correspond to the nodes (i.e. tracklets) in the constructed global graph. The entries of the vector are mostly zeros, except for the ones corresponding to the nodes which reflect the query. For instance, suppose a user is interested in objects or events related to "region B of camera c between time t_1 and t_2" (Spatial-Temporal Query) (see Fig. 1(b)). The system will first identify all tracklets with records directly match this criteria and mark the corresponding m nodes $\{\mathcal{G}_q\}$ as the query nodes. Then, a uniform score is assigned to these nodes in the query set $\{\mathcal{G}_q\}$, i.e., $r_i = 1/m$ if $i \in \{\mathcal{G}_q\}$, and $r_i = 0$ otherwise. For the Object Query (see Fig. 1(a)), it is more straightforward. For example, to search for "object instance appears at time t of camera 3", which corresponds the node j in the graph, the vector is set as $r_j = 1$ with all other entries are set to 0.

B. Ranking on the Graph

Given a set of query nodes, the goal is to rank and retrieve all other nodes based on their degree of relevance to the query nodes with respect to the global intrinsic graph structure. There are many off-the-shelf graph-based ranking methods, such as random walk [24], hubs and authorities [35], elastic springs [21], electric network [36] and many more. We adopt manifold ranking [13] to our work due to its well known performance on visual retrieval applications [19], [20].

Let $f \in R^{N\times 1}$ be the vector containing the final ranking scores of graph nodes. It has been shown in [13] that the solution of f can be obtained with the following closed-form:

$$f = (1-\mu)(I - \mu S)^{-1} r, \tag{9}$$

where $S = D^{-1/2} W D^{-1/2}$, W is the adjacency matrix representing the constructed global graph; D is a diagonal matrix whose (i,i)-entry equal to the sum of the i-th row of W, r is the query vector computed in Section IV-A, and μ is a constant coefficient between 0 and 1. It has also been shown that f obtained from the closed-form equation is equivalent to the following iterative process

$$f^{(t+1)} = \mu S f^{(t)} + (1-\mu) r. \tag{10}$$

This iterative algorithm has been proven to converge, and it is usually more efficient when the size of the graph is large. An intuitive description of the ranking process is to iteratively spread the scores from the query nodes to the rest of the nodes by simulating a "gradient walk" on the graph. Based on how likely other nodes can be reached, ranking scores will be assigned. Fig. 7 gives an visualization on how a walk starts from one node, and ends at a distant node through assorted links. The parameter μ in (10) essentially controls "how far away" the walk can go starting from the query nodes. The value of μ is fixed at 0.8 in this work. Note that the final rank for each node obtained in this process is a balance on both its relevance level to the query and its importance (i.e. centrality) level among all other nodes in the graph.

C. Diversification

Manifold ranking take into account centrality (relevance and importance of the data), however it does not consider diversity. For example, if a node is very close to a high ranked node, it will share a similar high rank. In order to take into consideration of diversity, recently methods such as decayed DivRank [37], absorbing random walks [38], and manifold ranking with stop points [39] have been proposed. The basic idea is to let a high ranked node to transform into an "absorbing" state (i.e. a sink point) during the ranking process on graphs. This node will then "drag down" the importance value of other connected unranked nodes, thus encouraging diversity. Similar to this, we develop a greedy algorithm which explicitly penalize redundancies and promote diversity (Algorithm 1). This penalty, for example, minimizes the possibility of retrieving a redundant list of tracklets of the same person from the same camera view.

Algorithm 1 Diversification

```
Input: Set $A = \{x_i | i = 1, 2, \ldots, n\}$ containing the list of
nodes (i.e. tracklets), and each element is associated with a
score computed from manifold ranking, i.e., $Score(x_i) = f_i^*$,
$i = 1, 2, \ldots, n$.
Output: Set $B$ containing the nodes with updated ranks
1: Initialization $B = \emptyset$
2: Sort the tracklets in $A$ by their current overall ranking
   scores in descending order
3: Identify the tracklets with the highest ranked value, i.e.
   first element in the sorted list. Suppose sub-track $x_i$ is the
   highest, first move it from set $A$ to $B$, and then a penalty
   is imposed to all other sub-tracks which are linked with
   $x_i$ and also from the same scene (i.e. camera view). For
   each tracklets $x_j$ satisfy the criteria, we update its score
   as follows:
       $Score(x_j) = Score(x_j) - \omega S_{ij} f_i^*$,
   where $\omega$ is a penalty coefficient greater than 0. A larger
   $omega$ value imposes greater penalty to the overall
   ranking score of a node.
4: Repeat from step 2 again until $A = \emptyset$ or the pre-defined
   iteration count has been reached
```

The central idea of the diversification algorithm is to decrease the overall ranking scores of nodes which have information already conveyed by the more informative ones. After the final scores for all the nodes are computed, the ones with the highest overall scores will be selected to answer the user query. The combination of manifold ranking and greedy diversification significantly increases the probability that a relevant yet diverse set of results to appear in the top-ranked list.

V. EXPERIMENTS

A. Dataset

Our data was collected from a recently deployed camera network at the UCSB campus. The network consists of over 40 stationary cameras covering wide area of the campus, including bicycle paths, popular walkways, and a vehicle traffic circle. In our experiments, a subset of 10 cameras are used as shown in Fig. 8. These fixed cameras are located near the bike paths monitoring non-motorized (pedestrian and cyclist) traffic. Our camera node consists of two parts, a Cisco WVC2300 wireless-G Internet video camera and a nearby dedicated computer for archiving and processing the video. The computer and the wireless camera together simulate a distributed "smart camera node" in a camera network. Communication from these nodes to the central server is also done through a wireless channel. A two hour video is recorded from each camera between 9:30am to 11am on a weekday during regular school quarter. Overlapping 15-minute duration video sequences are used from each camera for testing, and the remaining video data is used for learning the spatial-temporal regression functions as well as other parameters. In the 15-minute test data, there are on average 38 distinct people appearing in each camera view.

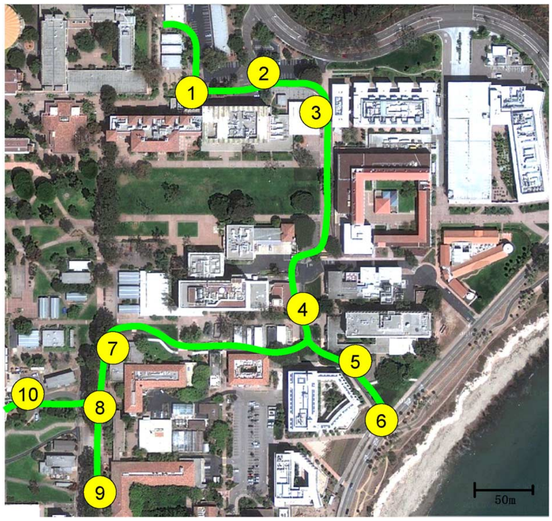


Fig. 8. Aerial map of the camera locations.

B. Quantitative Object Retrieval Evaluation

Object query-retrieval is evaluated in a way similar to traditional information retrieval setting where the top-K best results are returned based on the ranking algorithm. To measure the performance, we compute the standard precision v.s. scope curve, which is to evaluate the precision at different K values as defined below

$$Precision@K = \frac{\#\text{ of relevant items}}{\text{K}} \tag{11}$$

Note that an item is only counted as "relevant" if it is the same individual as the query in this case. To perform a systematic evaluation, we first identified 10 persons who have appeared in at least four different camera views in our network. For each person, we instantiated 5 queries using tracklets from randomly selected camera blocks. In other words, we have a total of 50 queries to the system. Precision values reported for this and subsequent experiments are based on the average of the 50 queries. There are two thresholds control the linkages in the graph, th_o and th_s as shown in Section III-B. In the experiment, we fix th_s to 0.5 seconds, and vary th_o among [1, 3, 5, 7] seconds. As we can see from Fig. 9, performance are slightly worse when th_o is 1. This is probably due to a strong assumption imposed on the time-delay regression. Performance for the rest are quiet similar, however, $th_o = 3$ seems to be best if we only retrieve the top-20 or fewer results.

In the second experiment, we investigate coverage of the results given a query. Assuming an object has been observed by a number of different camera views in our network, coverage

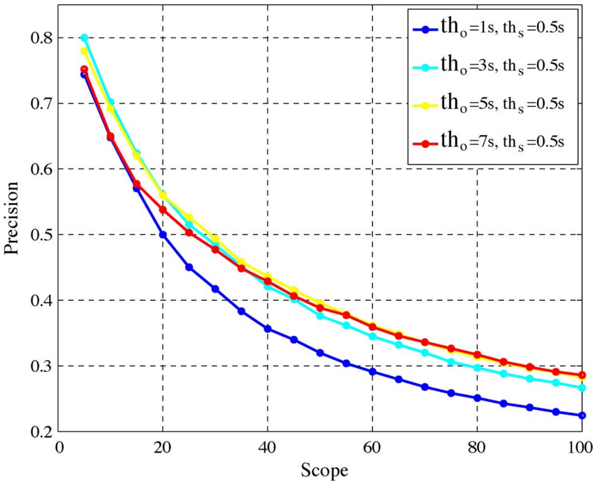


Fig. 9. Precision v.s. Scope curves showing the effects of different th_o thresholds for graph construction.

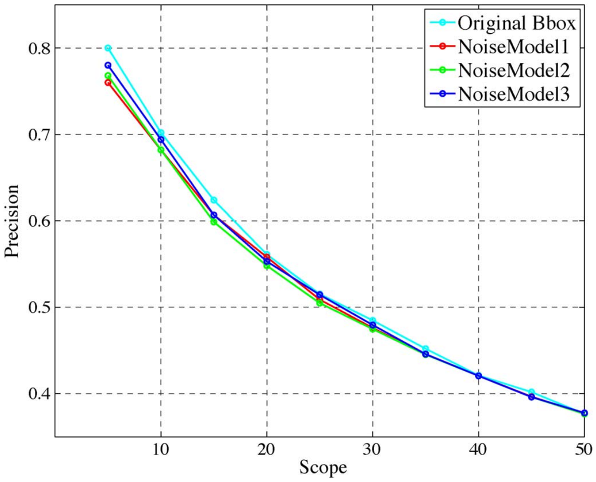


Fig. 11. Effects of adding noise to the original tracks.

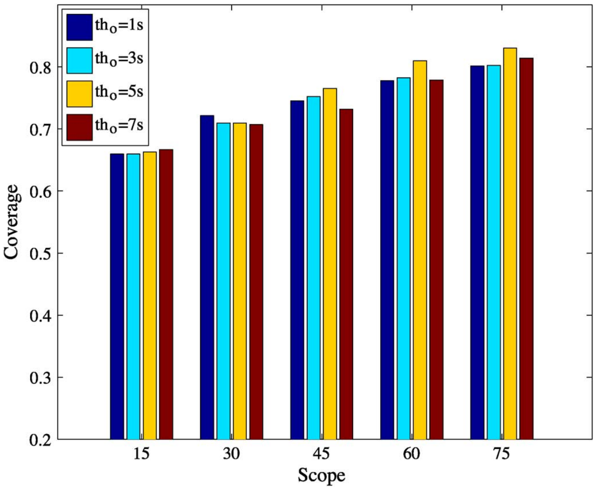


Fig. 10. Coverage v.s. Scope curves showing the effects of different th_o thresholds for graph construction.

simply measure the percentage of distinct views our results cover. We continue to fix threshold th_s at 0.5 sec, and vary th_o among [1, 3, 5, 7] sec. As can be seen in Fig. 10, more than 65% coverage rate is achieved by taking only the top-15 results, and the percentage gradually increases as we increase the number of retrieved results.

In the third experiment, we test the robustness of our proposed method against tracking failures. To do that, we perturbed the tracking bounding boxes using various degrees of severity. Fig. 11 shows the Precision v.s. Scope for the proposed methodology with various degrees of tracking failures. We disturbed the tracking bounding boxes with Gaussian noise with various parameters for $[xPos, yPos, scaleX, scaleY]$. In the experiments we used the following noise variances $\sigma_{NoiseModel1} = [4, 4, 0.1, 0.1]$, $\sigma_{NoiseModel2} = [6, 6, 0.2, 0.2]$ and $\sigma_{NoiseModel3} = [10, 10, 0.3, 0.3]$ with mean values centered at the original tracking bounding boxes. As can be seen in Fig. 11, there is very little drop in precision values as tracking noise increases.

C. *Qualitative Human Evaluation*

In our previous experiment, we evaluated the system as in the people re-identification paradigm. However, this evaluation alone may not be sufficient to justify the effectiveness of our system. This is because our system is designed to return not just the same object, but also objects which are *related* to the query in nearby spatial-temporal proximity. Thus it is important to understand how informative our results are and how well a human viewer can interpret based on the results. We conducted subjective evaluations using the Amazon Mechanical Turk (AMT)[1] to evaluate the usefulness of the proposed system for object browsing and retrieval.

We created four sets of experimental groups. For each group, we presented the users two sample queries: one for Object query (Fig. 1(a)), and one for Spatial-temporal query (Fig. 1(b)). Along with each query, we present a list of top-15 results containing the corresponding tracklets returned from the system (see Fig. 16 and Fig. 17). An aerial map of the camera locations along with a brief set of instructions for answering the questions is provided. Note that the content in each of the four experiment sets are all different, however, a common set of 8 multiple choice questions were asked to the human subjects. About half of these questions are based on the queried objects and the other half are related to the spatio-temporal activities. The idea is to evaluate the quality of our results, and test whether they are helpful to the human users. The 8 questions are as follows:

1) "How many camera views has the target cyclist passed through based on the results?"
2) "How many people was the target cyclist riding along with?"
3) "How many pedestrians stop for the target cyclist at the intersections, if there is any?"
4) "What is the cardinal direction (N,E,S,W) of the target based on the map?"
5) "What is the dominant travel pattern in the camera region at the specified time period?"

---

[1] https://www.mturk.com/mturk/welcome.

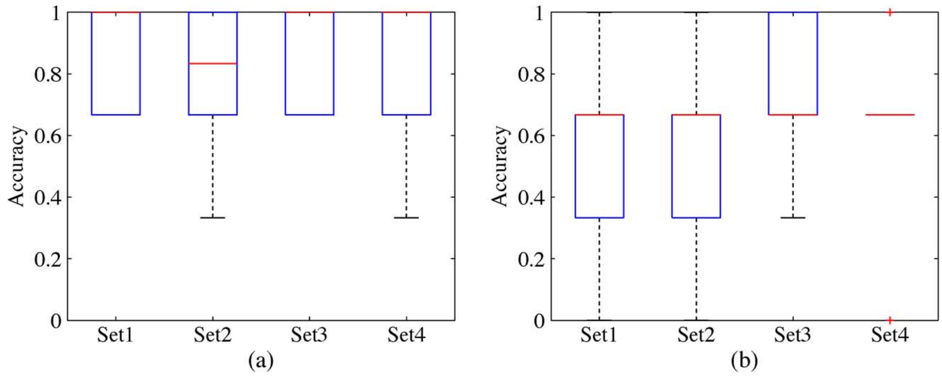


Fig. 12. % of correct answers from AMT users (a) object related questions (b) spatial-temporal related questions.

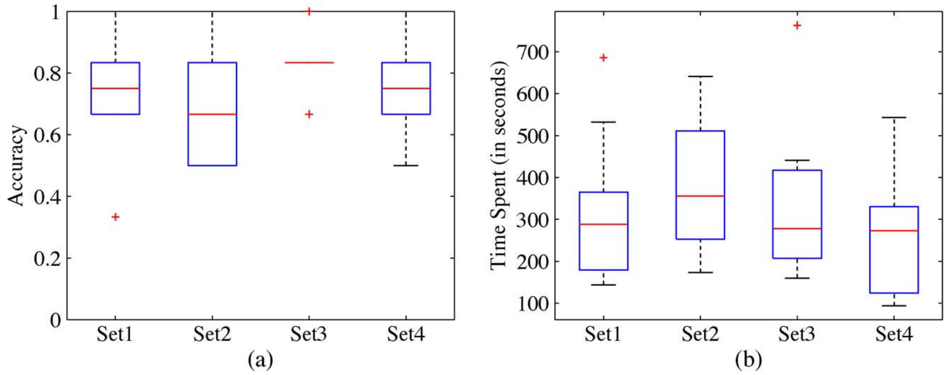


Fig. 13. (a) % of correct from AMT users for both type of questions (b) total time spent to answer all the questions.

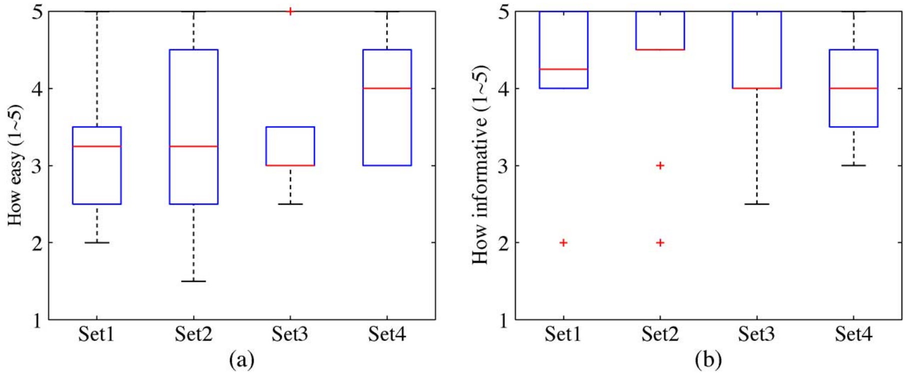


Fig. 14. Subjective evaluation from AMT users (a) how easy is to answer the questions based on the returned results? (b) How informative are the provided results for high-level scene understanding?

6) “How many people have passed through the query region in the specified time?”
7) “What is the more probable traffic direction across cameras at this time?”
8) “What is the estimated time-delay between cam X and cam Y for a cyclist?”

A total of 40 human subjects participated in the experiment, and they are evenly distributed to each of the four experimental groups. Fig. 12 shows box plots for the average accuracy of a human user over questions on for the two types of queries. As we can see, the median accuracy to answer object related questions are all above 80%, and the highest average accuracy are 1 across all experimental sets. This indicates the results provided for Object Query are highly relevant, and they are diverse enough to cover additional contextual information to the

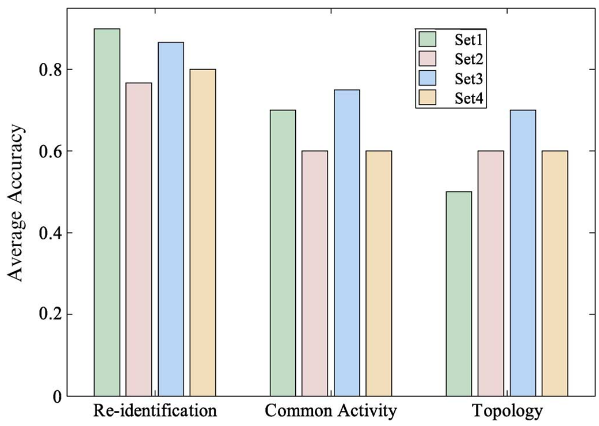


Fig. 15. % of correct for questions of different categories.

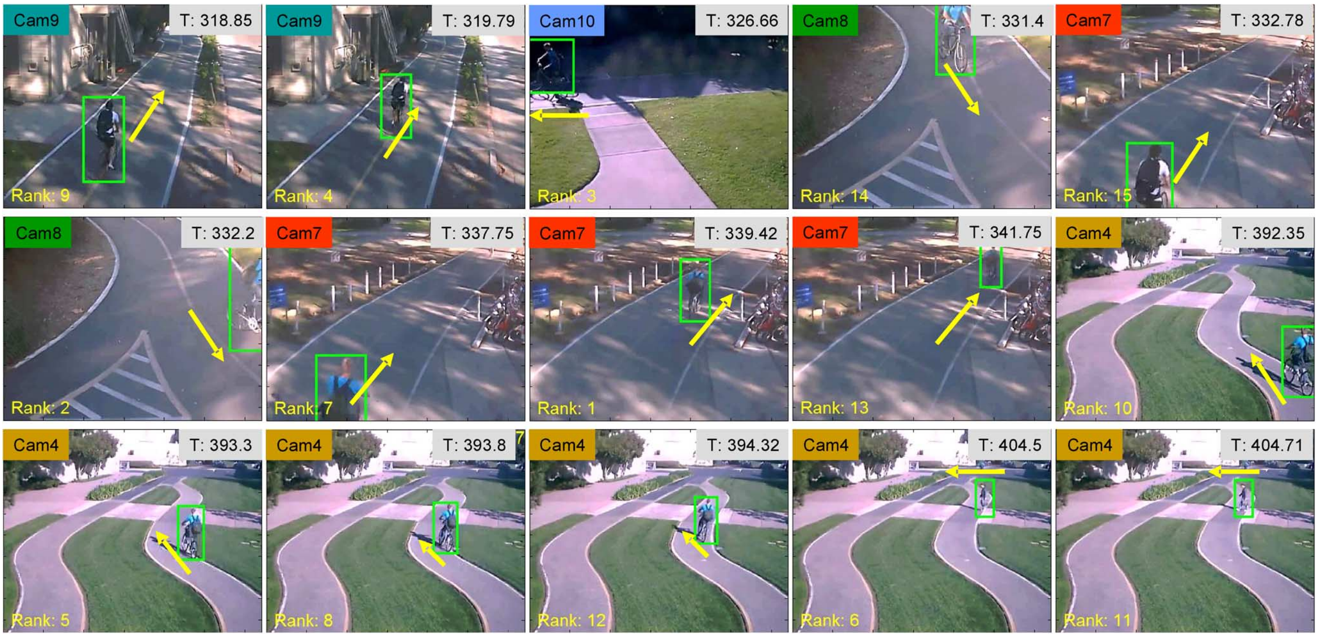


Fig. 16. Visual results of the Object Query shown in Fig. 1(a). Note that results are re-ordered (sorted in increasing temporal order) for viewing convenience. Additional metadata such as camera id, time stamp and object direction are extracted from actual record and imprinted on the frame for visualization purpose.

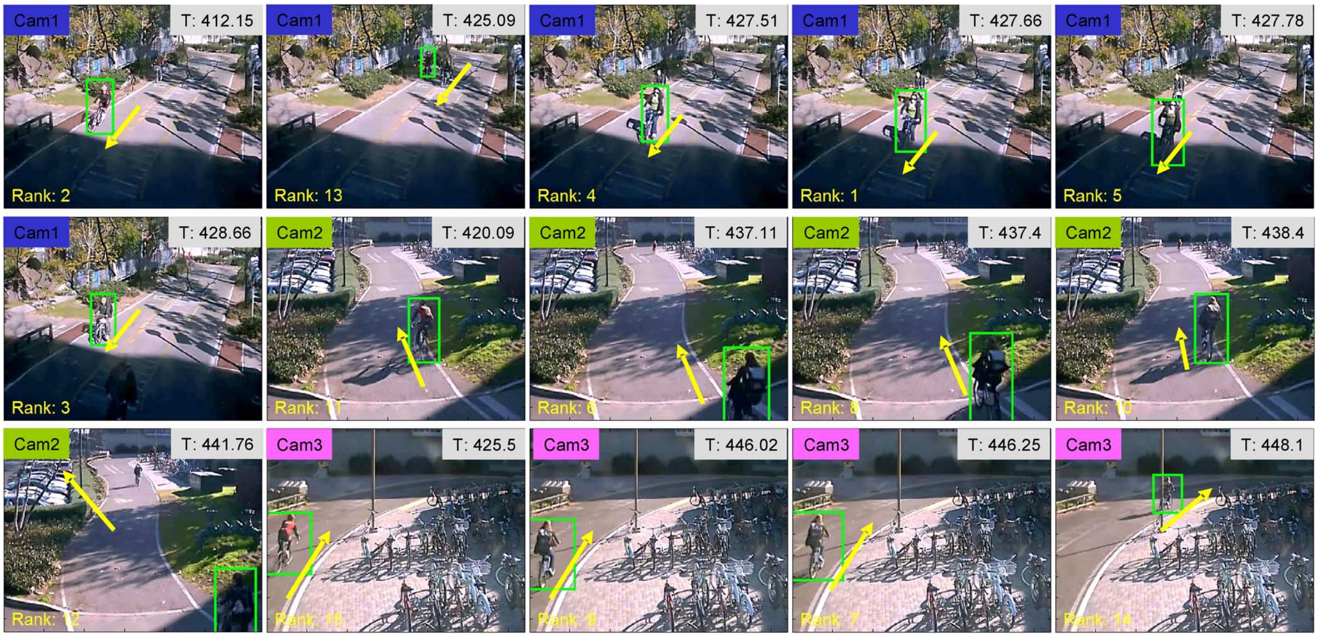


Fig. 17. Visual results of the spatial-temporal query shown in Fig. 1(b). Note that results are re-ordered (grouped by cameras and sorted in temporal order) for viewing convenience. Additional metadata such as camera id, time stamp and object direction are extracted from actual record and imprinted on the frame for visualization purpose.

human users. The median accuracy for Spatial-temporal related questions are a little lower (at 65%). This is probably because more human effort is required to identify different individuals and extract commonality from all of them. In addition, our display scheme may not be best suited for this type of tasks, and techniques such as visual collage [26] should be considered to be included in the visualization steps. Fig. 13(a) summarizes the overall accuracy for users to answer all 8 questions. Fig. 13(b) shows the box plot of total time spent on the experiment, which includes reading instruction, and answering all the questions. The time spent are fairly consistent (about 5 minutes) across all experimental set.

The original 8 multiple choice questions can also be classified into 3 categories, re-identification, common activity, and topology. A bar graph showing the average accuracy for each types of questions are shown in Fig. 15. As we can see, the accuracy on "topology" is the lowest overall. Example questions for this category include questions 7 and 8 listed above. We suspect this is another indication that better visualization mechanisms are needed to guide user navigate through results from different cameras.

In addition to the 8 multiple choice questions, we also ask 2 subjective questions to all users: “How easy is to answer the questions based on the returned results?” and “How informative are the provided results for scene understanding?”. Users are asked to answer the questions in a scale from 1 to 5, where 5 being easiest and most informative, and 1 being hardest and least informative. Box plots illustrating the results are shown in Fig. 14. Overall ratings show that results provided from our system is quite satisfactory.

D. Discussion on Graph Size

Recall that additional tracklets are appended to the graph incrementally, thus the graph can potentially grow out of control as time goes by. However, by design each tracklet is only connected to its spatial-temporal neighbors. The intuition is that detected objects from distant cameras or time separations are not supposed to be linked together as they cannot indicate any reliable relationships. In practice, one can construct a new graph for each time window. Furthermore, due to the spatial-temporal constraints, the constructed graph is very sparse. For instance, in our first experiment with the 10-camera network and 15 minutes of video from each of the cameras ($th_o = 3$ s, $th_s = 0.5$ s), we constructed a global graph with 7262 nodes (tracklets) and 607120 edges. In other words, the number of edges in the constructed graph is less than 2% compared to the number of edges in a fully-connected graph. Subsequent computation on the graph can be carried out very efficiently. In addition, recent work [19] proposed the use of “anchor graphs” to address scalability in graph-based algorithms, and their method is applicable to our scenario. Overall we believe there are many viable solutions to accommodate the analysis on large graphs.

VI. CONCLUSIONS

We presented the search and retrieval system for a distributed camera network. Raw videos are processed locally at the remote camera nodes, and only the computed information is passed to the central server. Based on the collected information, a graph is built to model the network observations to facilitate search and retrieval. A central philosophy of our work is to generate an informative and diverse set of visual results based on a user query. This is achieved by a manifold ranking on the constructed graph, followed by a greedy diversification algorithm. Experimental results using real world data has demonstrated the effectiveness of the proposed system.

REFERENCES

[1] O. Javed, Z. Rasheed, K. Shafique, and M. Shah, “Tracking across multiple cameras with disjoint views,” in *Proc. Int. Conf. Computer Vision*, 2003, pp. 952–960, IEEE Computer Society.

[2] O. Javed, K. Shafique, and M. Shah, “Appearance modeling for tracking in multiple non-overlapping cameras,” in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2005, pp. 26–33.

[3] V. Kettnaker and R. Zabih, “Bayesian multi-camera surveillance,” in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 1999, pp. 252–259.

[4] Z. Ni, J. Xu, and B. S. Manjunath, “Object browsing and searching in a camera network using graph models,” in *Proc. CVPR Workshops*, 2012, pp. 7–14.

[5] R. Collins, A. Lipton, H. Fujiyoshi, and T. Kanade, “Algorithms for cooperative multisensor surveillance,” *Proc. IEEE* vol. 89, no. 10, pp. 1456–1477, 2001. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=959341.

[6] O. Javed, Z. Rasheed, O. Alatas, and M. Shah, “Knight m: A real time surveillance system for multiple overlapping and non-overlapping cameras,” in *Proc. IEEE Conf. Multimedia and Expo*, 2003, pp. 6–9.

[7] M. Farenzena, L. Bazzani, A. Perina, V. Murino, and M. Cristani, “Person re-identification by symmetry-driven accumulation of local features,” in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2010, pp. 2360–2367.

[8] V. Jagadeesh, S. Karthikeyan, and B. Manjunath, “Spatio-temporal optical flow statistics (stofs) for activity classification,” in *Proc. 7th Indian Conf. Computer Vision, Graphics and Image Processing*, 2010, pp. 178–182, ACM.

[9] W.-S. Zheng, S. Gong, and T. Xiang, “Person re-identification by probabilistic relative distance comparison,” in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2011, pp. 649–656.

[10] R. Rios Cabrera, T. Tuytelaars, and L. Van Gool, “Efficient multicamera detection, tracking, and identification using a shared set of haar-features,” in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2011, pp. 65–71.

[11] E. Corvee, S. Bak, and F. Bremond, “People detection and re-identification for multi surveillance cameras,” in *Proc. VISAPP—Int. Conf. Computer Vision Theory and Applications-2012*, Rome, Italy, Feb. 2012. [Online]. Available: http://hal.inria.fr/hal-00656108, biometrics Groups at TELECOM SudParis, Multimedia Image processing Group of Eurecom and T3S (Thales Security Systems and Solutions S.A.S.).

[12] K. Hong, M. Voelz, V. Govindaraju, B. Jayaraman, and U. Ramachandran, A Distributed System for Supporting Spatio-Temporal Analysis on Large-Scale Camera Networks. School of Computer Science, Georgia Inst. Technol., 2012, Tech. Rep..

[13] D. Zhou, J. Weston, A. Gretton, O. Bousquet, and B. Schölkopf, “Ranking on data manifolds,” in *Proc. NIPS*, 2003.

[14] S. Agarwal, “Ranking on graph data,” in *Proc. 23rd Int. Conf. Machine Learning, ser. ICML ’06*, New York, NY, USA: ACM, 2006, pp. 25–32. [Online]. Available: http://doi.acm.org/10.1145/1143844.1143848.

[15] X. Wan, J. Yang, and J. Xiao, “Manifold-ranking based topic-focused multi-document summarization,” in *Proc. 20th Int. Joint Conf. Artifical Intelligence*, 2007, pp. 2903–2908.

[16] L. Cao, J. Luo, and T. S. Huang, “Annotating photo collections by label propagation according to multiple similarity cues,” in *Proc. 16th ACM Int. Conf. Multimedia, ser. MM ’08*, New York, NY, USA: ACM, 2008, pp. 121–130. [Online]. Available: http://doi.acm.org/10.1145/1459359.1459376.

[17] R. Ohbuchi and T. Shimizu, “Ranking on semantic manifold for shape-based 3d model retrieval,” in *Proc. 1st ACM Int. Conf. Multimedia Information Retrieval, ser. MIR ’08*, New York, NY, USA: ACM, 2008, pp. 411–418. [Online]. Available: http://doi.acm.org/10.1145/1460096.1460163.

[18] J. He, M. Li, H.-J. Zhang, H. Tong, and C. Zhang, “Manifold-ranking based image retrieval,” in *Proc. 12th Annual ACM Int. Conf. Multimedia, ser. MULTIMEDIA ’04*, New York, NY, USA: ACM, 2004, pp. 9–16. [Online]. Available: http://doi.acm.org/10.1145/1027527.1027531.

[19] B. Xu, J. Bu, C. Chen, D. Cai, X. He, W. Liu, and J. Luo, “Efficient manifold ranking for image retrieval,” in *Proc. SIGIR*, 2011, pp. 525–534.

[20] J. He, M. Li, H. Zhang, H. Tong, and C. Zhang, “Generalized manifoldranking-based image retrieval,” *IEEE Trans. Image Process.*, vol. 15, no. 10, pp. 3170–3177, Oct. 2006.

[21] L. Lovász, “Random walks on graphs: A survey,” in *Combinatorics, Paul Erdös is Eighty*, D. Miklós, V. T. Sós, and T. Szönyi, Eds. Budapest, Hungary: János Bolyai Mathematical Society, 1996, vol. 2, pp. 353–398.

[22] Y. Jing and S. Baluja, “Visualrank: Applying pagerank to large-scale image search,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 11, pp. 1877–1890, Nov. 2008.

[23] F. Richter, S. Romberg, E. Hörster, and R. Lienhart, “Multimodal ranking for image search on community databases,” in *Proc. ICMR*, 2010.

[24] L. Page, S. Brin, R. Motwani, and T. Winograd, The Pagerank Citation Ranking: Bringing Order to the Web, Stanford InfoLab, Tech. Rep. 1999-66, Nov. 1999. [Online]. Available: http://ilpubs.stanford.edu:8090/422/.

[25] H.-K. Tan and C.-W. Ngo, "Fusing heterogeneous modalities for video and image re-ranking," in *Proc. 1st ACM Int. Conf. Multimedia Retrieval*, 2011, pp. 1–8. [Online]. Available: http://doi.acm.org/10.1145/1991996.1992011.

[26] T. Mei, B. Yang, S.-Q. Yang, and X.-S. Hua, "Video collage: Presenting a video sequence using a single image," *Vis. Comput.* vol. 25, no. 1, pp. 39–51, Dec. 2008. [Online]. Available: http://dx.doi.org/10.1007/s00371-008-0282-4.

[27] J. Lee, J.-H. Oh, and S. Hwang, "Scenario based dynamic video abstractions using graph matching," in *Proc. ACM Multimedia*, 2005, pp. 810–819.

[28] Y. Gong and X. Liu, "Summarizing video by minimizing visual content redundancies," in *Proc. ICME*, 2001.

[29] Y. Peng and C.-W. Ngo, "Clip-based similarity measure for querydependent clip retrieval and video summarization," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16, no. 5, pp. 612–627, May 2006.

[30] C.-W. Ngo, Y.-F. Ma, and H.-J. Zhang, "Video summarization and scene detection by graph modeling," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 2, pp. 296–305, Feb. 2005.

[31] Y. Fu, Y. Guo, Y. Zhu, F. Liu, C. Song, and Z.-H. Zhou, "Multi-view video summarization," *IEEE Trans. Multimedia*, vol. 12, no. 7, pp. 717–729, Nov. 2010.

[32] C. De Leo and B. S. Manjunath, "Multicamera video summarization from optimal reconstruction," in *Proc. 2010 Int. Conf. Computer Vision—Volume Part I, ser. ACCV'10*, Berlin, Germany: Springer-Verlag, 2011, pp. 94–103. [Online]. Available: http://dl.acm.org/citation.cfm?id=2040690.2040701.

[33] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 4, pp. 603–619, 2002.

[34] R. T. Collins, "Mean-shift blob tracking through scale space," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2003.

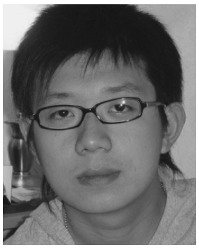
[35] J. M. Kleinberg, "Authoritative sources in a hyperlinked environment," *J. ACM* vol. 46, no. 5, pp. 604–632, Sep. 1999. [Online]. Available: http://doi.acm.org/10.1145/324133.324140.

[36] C. R. Palmer and C. Faloutsos, "Electricity based external similarity of categorical attributes," in *Proc. 7th Pacific-Asia Conf. Advances in Knowledge Discovery and Data Mining*, Berlin, Germany: Springer-Verlag, 2003, ser. PAKDD'03, pp. 486–500. [Online]. Available: http://dl.acm.org/citation.cfm?id=1760894.1760959.

[37] P. Du, J. Guo, and X.-Q. Cheng, "Decayed divrank: Capturing relevance, diversity and prestige in information networks," in *Proc. Int. ACM SIGIR Conf. Research and Development in Information Retrieval*, 2011, pp. 1239–1240.

[38] X. Zhu, A. B. Goldberg, J. Van Gael, and V. G. Andrzejewski, "Improving diversity in ranking using absorbing random walks," in *Proc. Annu. Conf. North American Chapter of the Association for Computational Linguistics*, 2007, pp. 97–104.

[39] X. Zhu, J. Guo, X. Cheng, P. Du, and H.-W. Shen, "A unified framework for recommending diverse and relevant queries," in *Proc. Int. Conf. World Wide Web*, 2011, pp. 37–46.



Jiejun Xu received his M.S. and Ph.D. degree in computer science from University of California Santa Barbara in 2007 and 2013, respectively. His research interests are computer vision and multimedia retrieval. Prior to joining the Ph.D. program, he was with the Center for Bio-Image Informatics, and worked on developing the Bio-Image Semantic Query User Environment (Bisque). He has interned with Telefonia Research in Barcelona and HRL Laboratories in Malibu. He was also a recipient of the NSF IGERT fellowship in 2007.

Vignesh Jagadeesh received his Ph.D. in Electrical Engineering from UCSB in 2013. He holds an M.S. in ECE from UCSB, and a B.S. from Anna University (India). His research interests span computer vision, image analysis and processing. His graduate research work is on interactive and scalable image analysis to handle large scale image mosaics generated by high throughput electron microscopy. He has interned for two summers with True Vision 3D Surgical, one summer each with Mayachitra Inc. and eBay Research. He is a member of the IEEE.

Zefeng Ni received the B.Eng. degree in computer engineering from the Nanyang Technological University (NTU), Singapore, in 2003. He received the M.S. and Ph.D. degree in electrical and computer engineering from the University of California, Santa Barbara, in 2007 and 2012, respectively. His graduate research work focused on object tracking and searching in distributed smart camera network. He worked on scalable video coding and transmission as a research assistant in the Center for Multimedia and Network Technology at NTU from 2004 to 2006. He is now a data scientist at Freelancer.com.

Santhoshkumar Sunderrajan is a PhD student in the ECE Department at UCSB. He holds an M.S. in ECE from UCSB, and a B.S. from Anna University (India). His research interests include computer vision and large scale machine learning. His graduate research work is focussed on multiple camera tracking and activity analysis.

B. S. Manjunath (F'05) received the B.E. degree (with distinction) in electronics from Bangalore University, Bangalore, India, in 1985, the M.E. degree (with distinction) in systems science and automation from the Indian Institute of Science, Bangalore, in 1987, and the Ph.D. degree in electrical engineering from University of Southern California, Los Angeles, in 1991. He is now a Professor of electrical and computer engineering and Director of the Center for Bio-Image Informatics at the University of California, Santa Barbara. His current research interests include image processing, data hiding, multimedia databases, and bio-image informatics. He has published over 250 peer-reviewed articles on these topics and is a co-editor of the book Introduction to MPEG-7 (Wiley, 2002). He was an associate editor of the IEEE TRANSACTIONS ON IMAGE PROCESSING, Pattern Analysis and Machine Intelligence, Multimedia, Information Forensics, the IEEE SIGNAL PROCESSING LETTERS and is currently an AE for the BMC Bioinformatics Journal. He is a co-author of the paper that won the 2013 Trans. Multimedia best paper award and is a fellow of the IEEE.