Cai *et al. EURASIP Journal on Image and Video Processing* (2017) 2017:19
DOI 10.1186/s13640-017-0165-6

EURASIP Journal on Image and Video Processing

RESEARCH Open Access

Querying geo-tagged videos for vision applications using spatial metadata

Yinghao Cai[1*], Ying Lu[2], Seon Ho Kim[2], Luciano Nocera[2] and Cyrus Shahabi[2]

Abstract

In this paper, we propose a novel geospatial image and video filtering tool (GIFT) to select the most relevant input images and videos for computer vision applications with geo-tagged mobile videos. GIFT tightly couples mobile media content and their geospatial metadata for fine granularity video manipulation in the spatial and temporal domain and intelligently indexes field of views (FOVs) to deal with large volumes of data. To demonstrate the effectiveness of GIFT, we introduce an end-to-end application that utilizes mobile videos to achieve persistent target tracking over large space and time. Our experimental results show promising performance of vision applications with GIFT in terms of lower communication load, improved efficiency, accuracy, and scalability when compared with baseline approaches which do not fully utilize geospatial metadata.

1 Introduction

In the past decade, we have witnessed rapid advances in multimedia data collection and management technologies as well as computer vision applications for the analysis of the collected data. However, there exists an obvious gap between the two well-known fields. Computer vision algorithms mainly focus on the analysis of a given set of input images and videos without much considering what would be the most effective input dataset for the analysis. Data management techniques concentrate on image and video management without providing the search and query capabilities that would be the most useful for vision applications. The result is the inability to utilize the full potentials of underlying techniques in both fields, especially in the era of an explosive increase of image/video data from various sources such as ubiquitous sensors and open-source social media. Due to the unstructured nature of image/video data, vision algorithms may not scale well as the number of input images/videos increases.

A number of trends have recently emerged around imagery data collection. First, we are experiencing enormous growth in the amount of images/videos collected from various sources such as CCTV cameras, unmanned aerial vehicles (UAVs), and drones and especially open sources like ubiquitous mobile devices. It is reported that smartphones become the predominant way people consume video. Second, the continuous fusion of geospatial metadata (e.g., camera location, viewing direction) with videos at a fine granular level has become feasible and transparent for users, leading to the concept of geo-tagged videos or sensor-rich videos [14, 29]. These geospatial metadata can be obtained either from built-in sensors [14, 29] or from image/video analysis [27, 35]. Third, it is observed that the geographical properties of images/videos provide meaningful context information for humans to better understand the media content. For example, a panoramic view which gives a wide-angle view of the specific physical space provides rich immersive user experiences.

However, even though these complementary geospatial metadata are increasingly available, they have been largely underutilized, which in some cases has led to sub-optimal or ad hoc data management solutions to be incorporated in computer vision applications. Here, we identify the needs for technology advances to be incorporated with vision applications: (1) methods to acquire fine granular geospatial metadata for dynamically changing environments and (2) effective management of geospatial metadata to enable efficient search and query of large image/video database.

To address these needs, in this paper, we propose a novel geospatial image and video filtering tool (GIFT), which provides a general and systematic means to select the most

*Correspondence: caiyinghao@gmail.com
[1] Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China
Full list of author information is available at the end of the article

relevant input images/videos for vision algorithms with geo-tagged videos. GIFT harnesses and manages geospatial metadata acquired during recording time and intelligently indexes field of views (FOVs) to deal with large volumes of data. GIFT efficiently filters out unrelated image/video data using geospatial metadata in interacting with the vision applications. GIFT is expected to improve the overall performance of vision algorithms with geo-tagged videos in terms of (1) lower communication cost, (2) faster image processing, and (3) scalability by selecting and transmitting only the most relevant data for a given vision task. The innovative claims of GIFT include (1) maximizing the utility of existing geospatial metadata acquisition and extraction technologies, (2) efficiently managing media content with the associated geospatial metadata collected from mobile devices for indexing and searching, and (3) supporting various vision technologies to enable scalable image/video analytics.

In order to demonstrate the effectiveness of GIFT, we apply GIFT to a computer vision application aimed at achieving persistent target tracking using videos captured from smartphones. Figure 1 illustrates how GIFT is applied to the persistent tracking problem. A target of interest (e.g., a person) is initially tagged in a video frame, and a tracker is then used to automatically follow the target. When the target is lost by the tracker, one needs to search for the target in videos close in space and time to the last known target location. GIFT is used to select the most relevant video segments to allow automatic re-identification and subsequent tracking of the target. This process of re-identification and tracking is repeated so that the target is persistently tracked—in an automated fashion—across multiple video segments from different videos. Experimental results using both synthetic and real dataset demonstrate that with GIFT, the overall performance of persistent tracking is improved in terms of:

- Efficiency and lower communication cost: GIFT effectively selects a small number of the most relevant video segments as the input to the tracking task. Therefore, the running time efficiency of the tracking system is improved while the amount of video data needed to be transferred over the network is dramatically reduced.
- Re-identification and tracking accuracy: it is known that re-identifying targets in a large video repository is error-prone. The more number of times the target needs to be matched with, the more likely the false correspondences. GIFT makes the tracking system more accurate by effectively reducing the number of unnecessary target matching.

The remainder of the paper is organized as follows. An overview of the related work is presented in Section 2. The main concept of GIFT is introduced in Section 3. We introduce a use case of GIFT, persistent tracking, in Section 4. Experimental results and conclusions are given in Section 5 and Section 6, respectively.

2 Related work

With the increasing amount of images and videos and the availability of geospatial metadata, the combination of multimedia content with geospatial metadata has received a lot of attention. In this section, we briefly review existing work related to geo-tagged videos and persistent tracking applications.

In the field of spatial databases, some studies [14, 29] focused on modeling and representing the viewable space of a scene. Ay et al. [29] modeled the geo-referenced videos as a set of spatial objects, e.g., FOV objects. Kim et al. [13] represented the geo-videos as a vector model. Other studies [16, 18, 21, 25, 31, 34, 36] mainly focused on geo-video indexing and query processing. Navarrete et al. [25] utilized R-trees [7] and grid files to index the camera locations of videos. Toyama et al. [34] used grid files to index the camera location and temporal information of videos. These two studies [25, 34] treated videos as points. These four studies [16, 18, 21, 31] treated videos as FOV objects. Ay et al. [31] indexed FOV objects with R-tree. Ma et al. [21] presented a grid-based index for FOV objects . Kim et al. [16] presented an R-tree-based index

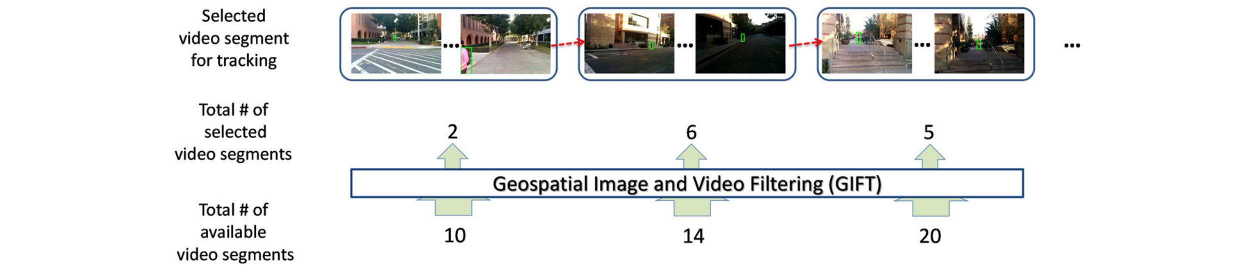


Fig. 1 Illustration of GIFT for persistent tracking. If the target exits the FOV of one video, GIFT effectively selects a small number of the most relevant video frames which may cover this target

named GeoTree for FOV objects. The difference between GeoTree and R-tree is that GeoTree stores the minimum bounding *tilted* rectangle (MB*T*R) in the leaf nodes. An MBTR is a long tilted rectangle paralleling with the moving direction of the FOV stream enclosed in the MBTR. A GeoTree is constructed based on the assumption that the camera moves at a constant speed. Trying to relax the assumption, Lee et al. [17] proposed another index named GeoVideoIndex by constructing an MBTR based on the linear change of moving direction of the camera to more properly fit the trajectory of moving scenes. Both GeoTree and GeoVideoIndex focus on dashcam videos recorded by cars driving on road networks, assuming camera shooting and moving directions does not change frequently. However, in real life, the moving directions and moving trajectories of user-generated mobile videos change frequently, e.g., with Google Glass or a mobile phone. Flora et al. [6] mainly dealt with how to use the existing spatial databases (e.g., Informix) to answer range queries more efficiently. Flora et al. [6] used polygons to represent the coverage of video segments. Lu et al. [18, 19] proposed an orientated R-tree (OR-tree) which considers both camera locations and shooting directions during building the index. While these work provide the ability to filter media content based on geospatial information, it does not take into account the specific requirements of vision applications.

Several vision applications are proposed which obtain or make use of the available metadata in various ways [2, 20, 24]. Tsai [35] proposed an approach to estimate the timestamp of photo capture by establishing relationship between the time and the measurable astronomical factors from a given outdoor photo, i.e., the sun position in the sky and the camera viewing direction in the photo-taken location. Crandall et al. [2] estimated the location of a photo by combining content analysis with structural analysis based on geospatial data. The spatial distribution of where people take pictures is used to define a relational structure which can reflect social processes influencing where people take pictures. Lu et al. [20] exploited geo-tagged photos for trip planning. The geo-location information embedded in the photos makes it possible to discover a tourist's travel route and thus recommend to other users. Mildner et al. [24] combined videos with location data in interactive video tours. While these previous work made use of geospatial metadata in vision-related applications and provide important motivation for our work, our work focuses on providing vision applications with the most relevant input dataset to facilitate image/video analytics, which is different from the focus of [2, 20, 24].

The general problem of persistent tracking is to continuously track a target, e.g., a car or pedestrian, over a large space and time. Since the FOV of a single camera is limited, most of the previous work addresses the problem of achieving persistent tracking using multiple stationary cameras [5, 10, 11]. It is usually assumed that these video feeds are synchronized and their camera network topology is known. The camera network topology is used to predict in which camera the target will reappear so that to reduce unnecessary target matching. Several unsupervised techniques have been developed [3, 22, 33] for the estimation of the camera network topology. Detmold et al. [3] estimated the camera network topology over a hundred camera network using a background subtraction technique to infer the mutual exclusion between pairwise cameras while [22] relied on the statistical consistency of observations to infer the connections among the cameras. Although Detmold et al. [3] method can be scaled to multi-thousand camera networks, both methods [3, 22, 33] relied on statistical observations over a period of time, which could not handle mobile videos captured in a casual way with various shooting directions and moving trajectories. In this paper, we show successful application of GIFT to the application of persistent target tracking which could handle videos captured from stationary cameras as well as videos captured with smartphones.

In the remainder of the paper, we present our novel approach, GIFT, which bridges the gap between the aforementioned video data management and vision applications. We further illustrate and quantify the benefits of GIFT by applying it to the problem of persistent tracking.

3 Geospatial image filtering tool

3.1 Overview

The framework of GIFT is shown in Fig. 2. The framework has three components: vision applications, GIFT, and the video database. The video database (e.g., MySQL) stores the geospatial metadata of videos. Video contents are stored as files. When the vision application requires a set of images/videos to process (e.g., generating a panoramic image), it sends a query request to GIFT. GIFT performs the query and returns a set of images or video segments as the query result. The image/video results are transferred to the application over the network.

3.2 Video database

More and more media contents are being geo-tagged through various methods such as built-in sensors [14, 29] or image analysis [27]. In this paper, each video frame is tagged with fine granular geospatial metadata (e.g., FOV) [14]. Figure 3 shows an example of the geo-tagged video being displayed on a map with a graphical representation of the geospatial metadata. The blue pie shape represents the FOV of the currently playing frame. The camera position, viewing direction, and its trajectory are also shown. As we can see from Fig. 3, the FOVs of the

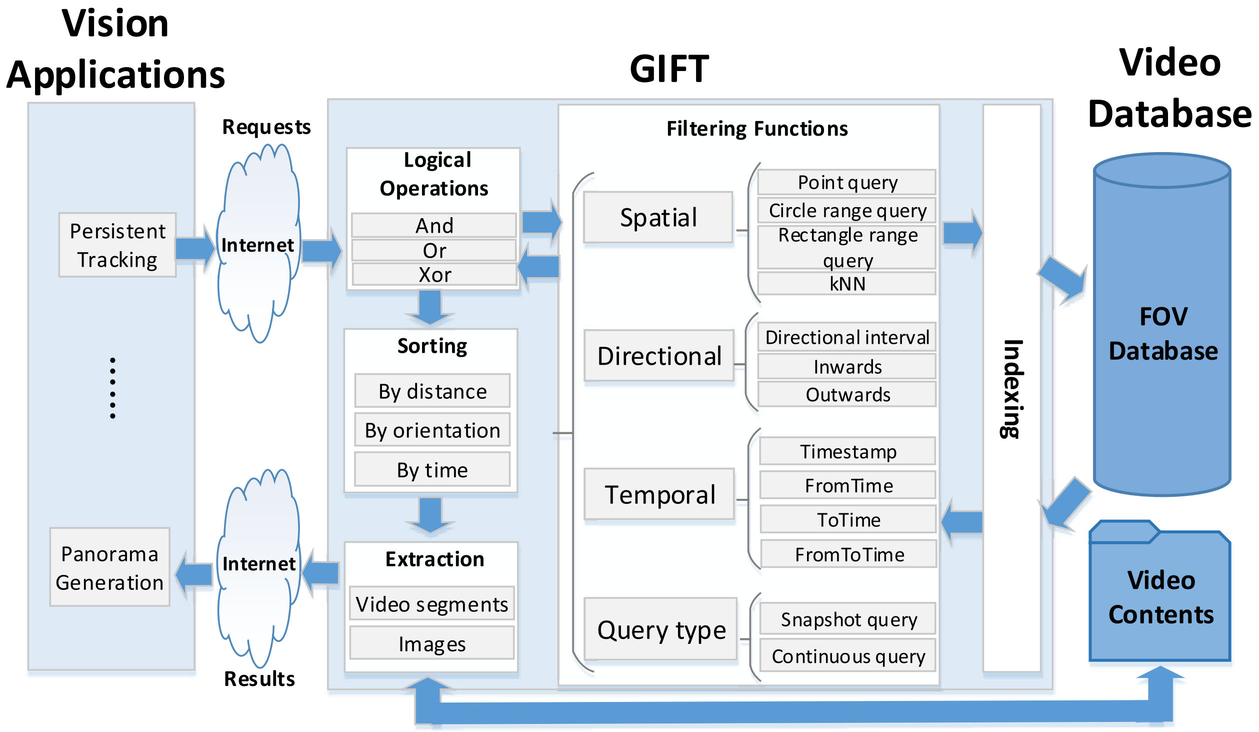


Fig. 2 The framework of GIFT

videos contain rich information about the scene which can be exploited in searching for video frames which overlap with a user-specified point such as a landmark or any location in the case of tracking a target. To this end, we implement spatial queries on the geo-tagged videos to provide computer vision applications with the most relevant videos for a given request. This capability is most important when working with a large video repository. GIFT tightly couples geospatial metadata with video contents and maximizes the use of geo-metadata.

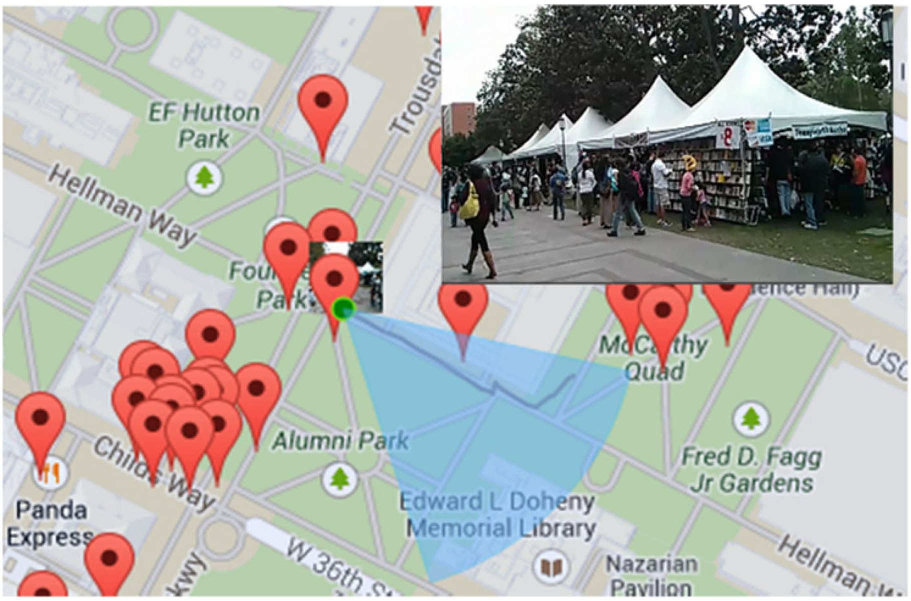


Fig. 3 An example of geo-tagged videos. Video frames are tagged with camera locations and viewing directions at the frame level. The *blue* pie shape represents the FOV of the currently playing frame of the video

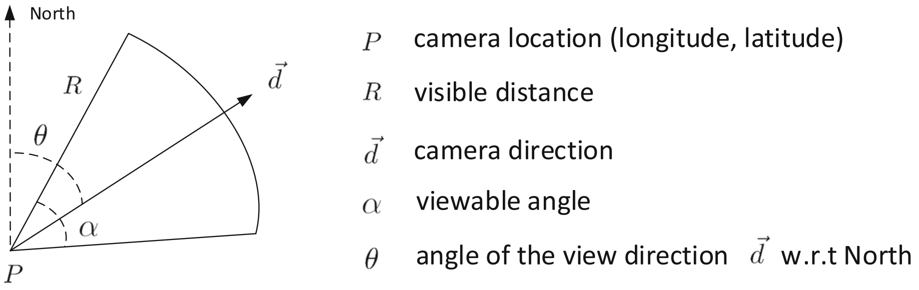

North
R
$\vec{d}$
θ
α
P
P camera location (longitude, latitude)
R visible distance
$\vec{d}$ camera direction
α viewable angle
θ angle of the view direction $\vec{d}$ w.r.t North

Fig. 4 Illustration of the FOV model

We use an existing prototype system, MediaQ [14], to collect images or videos as well as geospatial metadata. MediaQ is a mobile media management system which collects, organizes, and searches video contents using geospatial metadata. Each video frame is geo-tagged and represented as an FOV model as shown in Fig. 4. Camera location P is the latitude and longitude coordinates read from GPS sensor of the smartphone. Camera viewing direction $\vec{d}$ is the orientation angle provided by the digital compass. Camera viewable angle α describes the angular extent of the scene viewed by the camera. Visible distance R is defined as the maximum distance which can be viewed by the camera. Based on camera direction $\vec{d}$, we can determine θ which is the angle of the view direction with respect to the North direction. Therefore, the FOV model of each frame is represented as $f(P, \theta, \alpha, R)$. The video repository $\mathcal{V}$ is then represented as an FOV database $\mathcal{F} = \{f_{v_i} | \forall v_i \in \mathcal{V}\}$, where v_i is the video frame. The timestamp t and camera tilt of each video frame are captured as well in the metadata. It should be noted that the camera FOV is generally a 3D object. For a 3D representation of the FOV model, the altitude of the camera and camera tilt should be considered. In this paper, without loss of generality, we use 2D representation of FOV for simplicity.

3.3 Modules of GIFT

GIFT contains three main modules: logical operations, filtering functions, and sorting and extraction, which are elaborated individually below.

3.3.1 Logical operation module

The logical operation module (1) receives query requests from vision applications, where a *query request* is a logical combination of a set of filtering functions, (2) converts the specific query request to a set of *query functions* (e.g., filtering functions) to be executed in the filtering function module, (3) gets the FOV result sets of the filtering functions and combines the FOV result sets into a result set, and (4) sends the result set to the sorting module.

GIFT supports three basic logical operations, AND, OR, and XOR, for combining the filtering functions. Query requests are logical expressions of specified filtering functions, e.g., (f_1 *AND* f_2) or ((f_1 *OR* f_2) *AND* f_3), where f_1, f_2 and f_3 are filtering functions defined in Section 3.3.2.

3.3.2 Filtering function module

The filtering function module provides query functions with different filtering options based on the geo-metadata of videos, FOVs stored in the database. We provide three types of query functions depending on the type of filtering options: spatial queries, directional queries, temporal queries, and two types of queries in terms of query frequency (snapshot or continuous). These five types of queries are orthogonal and can be combined for different purposes.

Let $|pq|$ denotes the Euclidean distance between two points p and q. $\overrightarrow{\theta_b}$ (resp. $\overrightarrow{\theta_e}$) is the beginning (resp. ending) orientation of an FOV $f(p, \theta, \alpha, R)$ which can be calculated as $\theta_b = \left(\theta - \frac{\alpha}{2} + 360\right)$ mod360 (resp. $\theta_e = (\theta + \frac{\alpha}{2} + 360)$ mod360). $\overrightarrow{\Theta}$ is the orientation of the FOV in form of a tuple$< \overrightarrow{\theta_b}\ \overrightarrow{\theta_e} >$. Furthermore, we define $\overrightarrow{\theta_b}\ \overrightarrow{\theta_e}$ as the angle from $\overrightarrow{\theta_b}$ to $\overrightarrow{\theta_e}$ in clockwise direction.

Spatial queries The spatial queries filter FOVs with the spatial information. Spatial queries include point queries, circle or rectangle range queries, and kNN queries.

1. *Point queries* are defined as finding FOVs that contain the user-specified query point q. As shown in Fig. 5, the FOV $f(p, \theta, \alpha, R)$ contains the query point q if it satisfies: (1) the Euclidean distance between the query point q and the camera location p is smaller than the viewable distance R, and (2) the orientation $\overrightarrow{pq}$ is within the beginning orientation $\overrightarrow{\theta_b}$ and the ending orientation $\overrightarrow{\theta_e}$, e.g., $|pq| < R$ and $\overrightarrow{\theta_b}\ \overrightarrow{pq} + \overrightarrow{pq}\ \overrightarrow{\theta_e} = \overrightarrow{\theta_b}\ \overrightarrow{\theta_e}$.
2. *Circle or rectangle range queries* are defined as finding FOVs that overlap with the user-specified query circle or rectangle q_r. Take the circle range

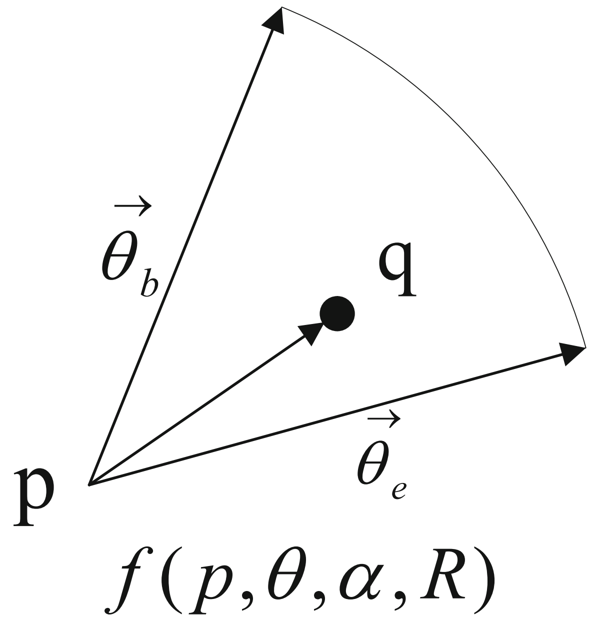


Fig. 5 Point query. Point query finds FOVs that contain the user-specified query point *q*

query for example, as shown in Fig. 6, there are four overlapping cases: (1) Camera location p is within the query Q_r. (2) p is outside of Q_r, and the ray $\overrightarrow{pq}$ is within the FOV view orientation of $f.\Theta$. (3) p is outside of Q_r, and the ray $\overrightarrow{\theta_b}$ is between the ray $\overrightarrow{pq}$ and ray $\overrightarrow{pg}$. (4) p is outside of Q_r, and the ray $\overrightarrow{\theta_e}$ is between the ray $\overrightarrow{pf}$ and ray $\overrightarrow{pq}$.

3. *kNN queries* aim to find k closest FOVs that show the user-specified query point q, where the closeness from an FOV f to the query point q is measured by the Euclidean distance between the camera location of f to the query point q.

Directional queries find FOVs which the orientations overlap with the user-specified direction requests. The direction requests can be directional internal (e.g., the Northeast), inwards (e.g., FOVs that pointing toward the query point) and outwards (e.g., FOVs that pointing outward the query point).

Temporal queries aim to find FOVs that are recorded within the user-specified query time. The query time can be specified as a specific timestamp or time interval.

Snapshot queries are evaluated/processed once assuming that the query results will not change as time goes on.

Continuous queries require continuous evaluation/ processing as the query results become invalid with the change of information (e.g., the change of query locations).

3.3.3 Indexing module

To accelerate the query processing for the filtering functions, we support FOV indexing in GIFT. Since FOVs are spatial objects in the shape of pie slice, in GIFT, we use R-tree [7] to enclose the area of each FOV with its minimum bounding rectangle (MBR). With R-trees, we use the pruning techniques supported by MySQL for our spatial filtering functions, such as point queries, range queries, and kNN queries.

3.3.4 Sorting and extraction module

For many vision tasks, it is often the case that only a small subset of all available input images or video segments is necessary. For example, for the application of persistent target tracking, FOVs which locations are closer to the target are more likely to be useful. To this end, GIFT provides three basic sorting functions: (1) sorting by distance, where the distance is the Euclidean distance from the FOV's camera location to the query point; (2) sorting by orientation, which is the angle between the FOV's orientation and the query direction; and (3) sorting by time, e.g., the closeness of the FOV's timestamp to the query time. The three sorting functions can be combined according to the requirements of the applications.

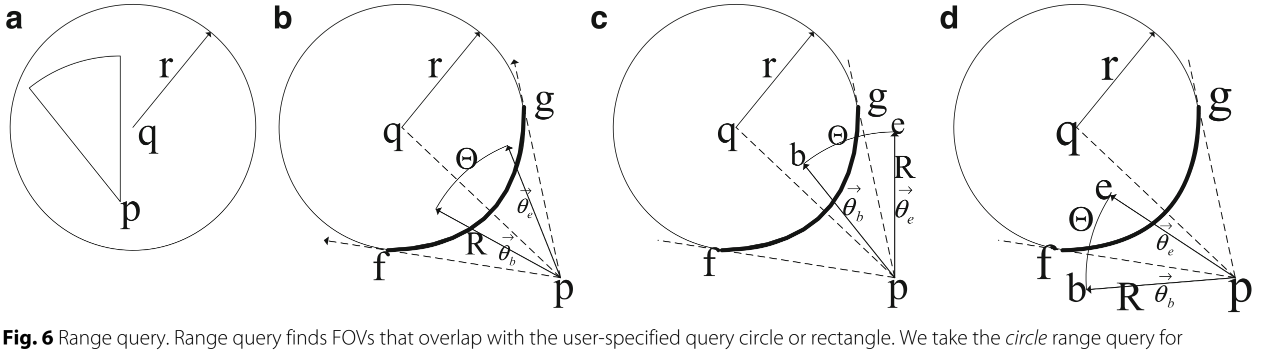


Fig. 6 Range query. Range query finds FOVs that overlap with the user-specified query circle or rectangle. We take the *circle* range query for example here. Four overlapping cases are introduced. **a** Case 1. **b** Case 2. **c** Case 3. **d** Case 4

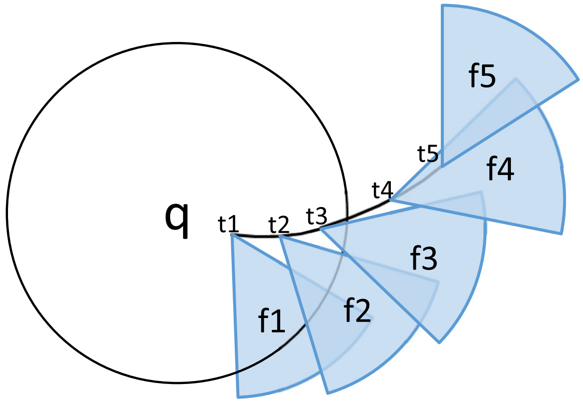


Fig. 7 Continuous returned FOVs that are in the same original video are combined into video segments using FFMPEG library

Note that the query results of the filtering functions obtained from the database consist of a set of FOVs. To return video frames or video segments to the application side, we use FFMPEG library to extract video frames or video segments based on the query results from the video contents stored in the file system.

To extract video segments based on the query results, we combine the continuous FOVs that are in the same original video into a segment in form of $Seg(vid, starttime, endtime)$, which can be used to extract video segments from videos through FFMPEG. For example, as shown in Fig. 7, circle q is the query circle and $f_1, \ldots f_5$ are the FOVs in a recorded video. In this example, FOVs f_1, f_2, and f_3 are the results for the range query q. Since f_1, f_2, and f_3 are continuous and they are in the same original video, we combine them together to generate a single video segment. The video ID, the starting time t_1, and the ending time t_3 are used to extract video segments from the video content using FFMPEG library.

3.4 Example of GIFT application

To illustrate how GIFT improves the efficiency of vision applications, we show an application of GIFT in panorama generation from our previous study [15]. Generally speaking, there are two types of panorama.

Point Panorama Point panorama generation is to generate a panoramic image as seen from a user-specified query point. Point panorama generation is illustrated in Fig. 8a. The main idea of GIFT here is to select video frames which the camera locations are within a predefined threshold radius r from the query point q. The radius r is defined as 10 m here which is a typical GPS error margin. Then, as shown in Fig. 9a, we divide 360° into n groups around the query point q based on the directions of FOVs. For each group, the best matching FOV is selected. The "best" metric is measured by a linear combination of the distance from the FOV's camera location to the point q and the direction difference between the FOV's orientation and the group direction.

Using GIFT, we apply circle range query first. Sorting function based on the linear combination of the distance and orientation difference for each group is then applied.

Route panorama Route panorama generation is to generate a panorama image as seen from a user-specified route on a specified side as shown in Fig. 8b. Here, the main idea of key frame selection is to select video frames which the camera locations with the route se are no larger than a predefined threshold r (e.g., 10 m). Then, as shown in Fig. 9b, we divide the specified route into a set of line segments based on the FOVs' camera locations. For each line segment, GIFT selects the most relevant FOVs where the relevance are measured as a linear combination of the distance from the FOV's camera location to the route and the orientation difference between the FOV's orientation and the specified orientation.

Using GIFT, we apply rectangle range query first. Then, we use sorting function based on the linear combination of distance and orientation difference for each line segment.

In experiments, videos are recorded from multiple people to generate panorama images. As shown in Fig. 10, without using GIFT for filtering, the stitching algorithm has to use all 228 video frames. However, GIFT only selects 13 frames without loss of image quality by filtering out redundant frames. It demonstrates that GIFT effectively eliminates redundant video frames for efficient panorama generation.

4 Use case: persistent tracking

In this section, we focus on a case study that can showcase the effectiveness of GIFT in vision applications. One of the challenging vision applications is persistent tracking which is to continuously track a target, e.g., car or pedestrian, over large space and time. In general, persistent tracking needs to analyze a series of videos to identify the target. Its processing time heavily depends on the amount of input data that needed to be processed. Thus, persistent tracking can be a good case study for the evaluation of GIFT.

Figure 11 shows an overview of the proposed persistent tracking system. We first tag a target which is then automatically tracked in videos. If the tracker ends, e.g., the tracker reaches the last frame of the video or the tracker loses the target, the persistent tracking system issues a request to GIFT to receive videos which may cover the target to allow for subsequent reacquisition and tracking. Video selection, re-identification, and tracking are repeated, and the target is persistently tracked—in an automatic fashion—across multiple video segments from different videos.

4.1 GIFT in persistent tracking

If the tracker ends at timestamp t_o as shown in Fig. 12, based on the last observed location P_o and the moving direction of the target, it is possible to predict the next possible locations of the target in the next consecutive timestamps t'_o using a constant velocity model. GIFT is applied to actively select video frames which may cover this target. Suppose the predicted location of the target is P'_o , we combine spatial queries with temporal queries in GIFT:

$$\begin{aligned}&\text{PointQuery}(P_o, t_o) \longleftarrow \\ &\qquad \{F_{v_i} \in \mathcal{F} \mid F_{v_i} \,\cap\, P'_o \;\neq\; \emptyset,\;\; 0 \le t'_o - t_o < \tau\} \\ &\text{RangeQuery}(P_o, t_o) \longleftarrow \\ &\qquad \{F_{v_i} \in \mathcal{F} \mid F_{v_i} \,\cap\, Q_r(P'_o, r) \;\neq\; \emptyset,\;\; 0 \le t'_o - t_o < \tau\}\end{aligned} \tag{1}$$

where $Q_r(P'_o, r)$ is a query circle with the center point P'_o and the radius r. $\cap$ is the geographical overlap. Point query finds all FOVs which overlap with the predicted location of the target. Range query aims at finding all FOVs which overlap with a specified region, e.g., a circle $Q_r(P'_o, r)$. Since long-term prediction of the target location may not be reliable, we limit the temporal range within a bound τ. An illustration of the point query and range query in persistent tracking is shown in Fig. 12.

As we can see from Fig. 12, point query and range query return FOVs which overlap with the predicted location of the target. Range query generally returns more FOVs compared with point query due to the relaxed constraints. Consecutive FOVs are grouped into video segments for tracking in the extraction module of GIFT. Short video segments less than 5 s are eliminated.

The target reacquisition module (Section 4.2) has to examine the reappearance of this target in all video segments returned by the queries. However, viewpoint-invariant person re-identification still remains a challenging problem [12]. For example, the side view and the frontal view of a person may look significantly different. We leverage the observation that it is advantageous to have a consistent viewpoint to observe the target for persistent tracking. Here, "viewpoint" refers to the camera's azimuth with respect to the target. To this end, we further combine direction query in GIFT to find FOVs whose camera orientations overlap with the target's moving direction within a range:

$$\begin{aligned}&\text{DirectionQuery}(\theta_o) \longleftarrow \\ &\qquad \{F_{v_i} \in \mathcal{F} \mid \theta_o - \delta \le 180 + F_{v_i}(\theta) \le \theta_o + \delta\}\end{aligned} \tag{2}$$

where $F_{v_i}(\theta)$ is the direction of the FOV with respect to the north direction. θ_o is the moving direction of the target, and δ is a direction margin since a precise moving direction of the target may not be easy to obtain. By applying direction query on the top of the results of point query and range query, we get a consistent frontal view of the target. Finally, query results are sorted in the sorting module of GIFT according to the distance between the camera location and the predicted location of the target.

Different combinations of queries are defined in GIFT. Without further specification, range query combined with temporal query is used in GIFT for persistent tracking. In GIFT, indexing FOVs (Section 3.3.3) is performed to speed up the queries. Since the timestamp information of each video frame is stored in the metadata, it naturally resolves the problem of video synchronization.

4.2 Target tracking and reacquisition

The target is first tagged at the first frame of one video and is then tracked persistently over large space and time.

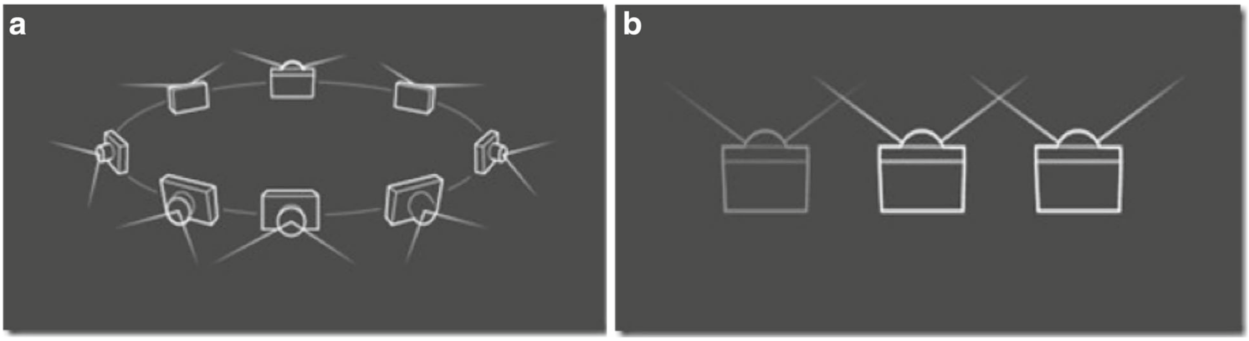


Fig. 8 Point panorama generation and route panorama generation. **a** Point panorama. **b** Route panorama

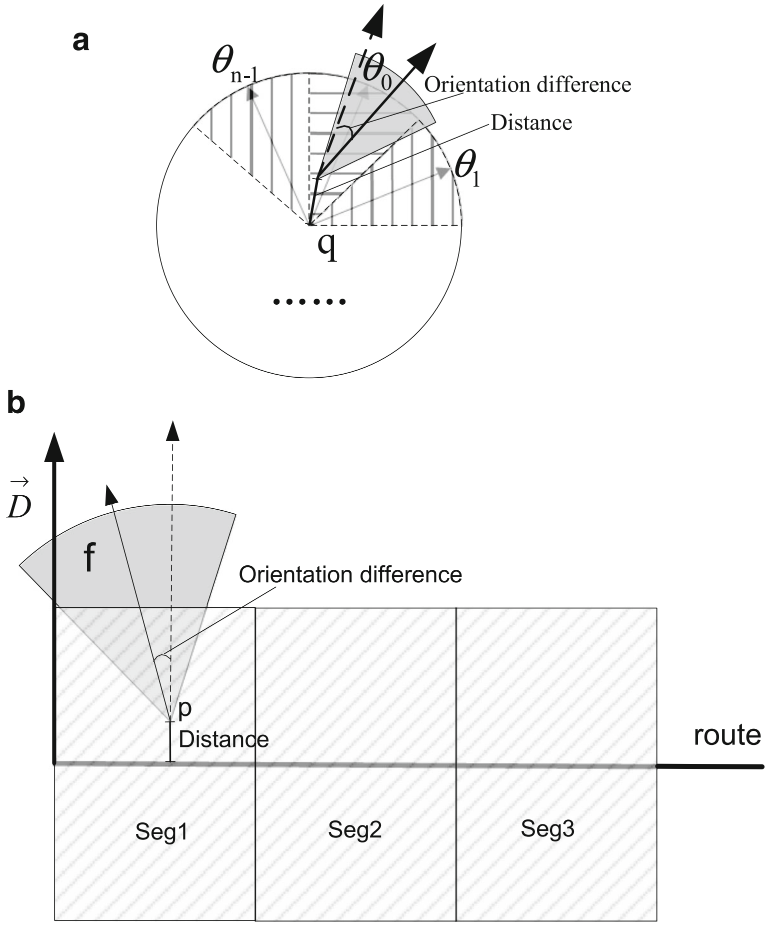


Fig. 9 Key frame selection for point and route panorama. **a** Point panorama. **b** Route panorama

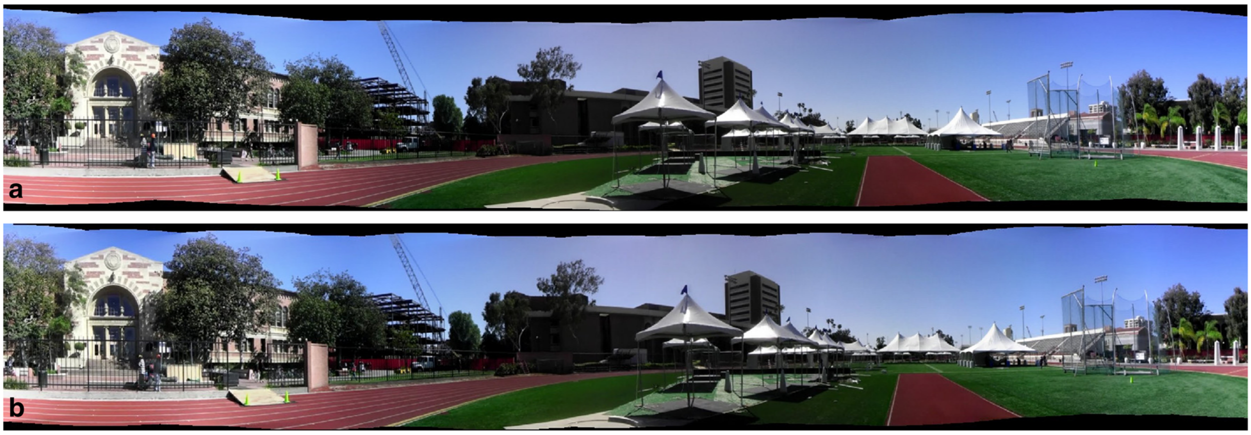


Fig. 10 GIFT in panorama generation. Images for stitching are extracted from 3 videos. **a** Without GIFT, the stitching algorithm has to use all 228 video frames for panorama generation. **b** With GIFT, only 13 FOVs are selected to generate the panorama without loss of image quality

4.2.1 Target tracking within cameras

In target tracking, reliable trajectories which are obtained by incremental linking of detection responses frame by frame are used to build the appearance models for visual tracking. If detection-based tracking fails, e.g., there is no detection response which can be associated with the target at frame t, visual tracking [4] is then triggered to resume tracking.

More specifically, we first apply a color normalization method [28] to reduce the illumination variations across frames. Pedestrian detector [9] is then applied at every frame. To reduce false alarms from the detector, given camera pose information read from the sensors, we further remove false alarms of detections using a prior on human heights. The probability of linking newly detected response to the existing trajectory is defined according to the appearance affinity, position affinity, and size affinity [23]. We only track the target we tagged at the first frame here.

The detection response with the maximum affinity score is used to update the target's trajectory and the visual tracker. Note that the detection-based tracking is only carried out between consecutive frames. When there is no detection response which can be associated with the target due to miss detection or low affinity score, we switch to the visual tracker to track the target. We employ the visual tracker named "context tracker" [4] in our approach. More details of the context tracker can be found in [4]. Figure 13 shows some results of tracking within cameras.

4.2.2 Track reacquisition across views

If the tracker proposed in Section 4.2.1 reaches the last frame of the video segment or the confidence of the tracker has been low for 10 frames, e.g., the target becomes occluded or exits the view, GIFT is applied to select video segments which may cover this target. We track each person in the first couple of frames of video segments returned from the GIFT module in order to reacquire the target.

Without frame-by-frame spatial continuity as in single camera tracking, target re-identification across views is a difficult problem. To this end, we first learn target-specific discriminative appearance models to discriminate visually very similar targets. Positive and negative samples of the target are collected online from single camera tracking. For the target we aim to track, we learn features which are most discriminative in appearance matching using the collected training samples by Adaboost algorithm [1, 32]. The appearance affinity is then computed by the weighted feature similarity measurements [1].

The spatial affinity between the geo-locations of track T_i and track T_j is defined as follows:

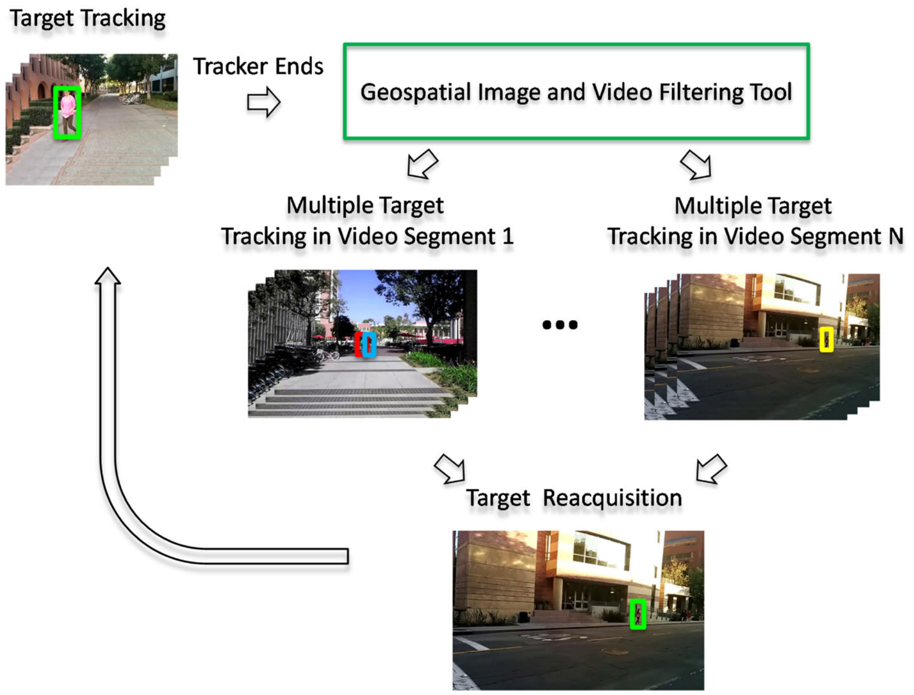

Target Tracking
Tracker Ends
Geospatial Image and Video Filtering Tool
Multiple Target Tracking in Video Segment 1
Multiple Target Tracking in Video Segment N
•••
Target Reacquisition

Fig. 11 An overview of the persistent tracking system

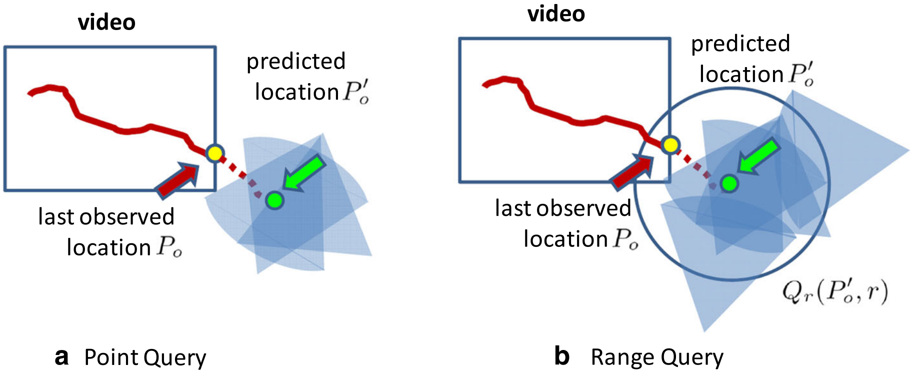

video
predicted location P'_o
last observed location P_o
a Point Query
video
predicted location P'_o
last observed location P_o
$Q_r(P'_o, r)$
b Range Query

Fig. 12 An illustration of point and range query in persistent tracking. Point query finds all FOVs which overlap with the predicted location of the target P'_o. Range query aims at finding all FOVs which overlap with a specified region $Q_r(P'_o, r)$. **a** Point query. **b** Range query

$$\text{Affinity}_{\text{pos}}(T_i, T_j) = \exp\left(\frac{-D(T_i^{'}, T_{j,\text{start}})}{\sigma}\right), \text{subject to}: \quad 0 \leq t_{j,\text{start}} - t_{i,\text{end}} < \tau \tag{3}$$

where $T_i^{'}$ is the predicted starting location of the track T_i in other video segments using constant velocity model. $D(\cdot)$ is the geographical distance. $t_{i,\text{end}}$ is the ending timestamp of track T_i, and $t_{j,\text{start}}$ is the starting timestamp of track T_j. τ is the same temporal window size as defined in Eq. 1. The target is reacquired by choosing the track with the largest affinity score of appearance and position.

GIFT largely reduces many unnecessary target matching by filtering out geographically far apart targets. Furthermore, the latitude and longitude of the tracks can be inferred from camera calibration (Section 4.2.3). It is possible to compare the spatial affinities of the tracks based on the geo-coordinates. Therefore, the accuracy and efficiency of track acquisition are improved with GIFT.

Once the target is reacquired in one video segment, the single camera tracker is then switched to this video segment where tracking is continued. If the target is not reacquired, e.g, no observations from the query results can be matched with this target, the persistent tracking task is finished.

4.2.3 Acquisition of the target's geo-coordinates

So far, we have introduced our tracking methods to persistently track the target both within and across video segments. In this section, we introduce how to get the

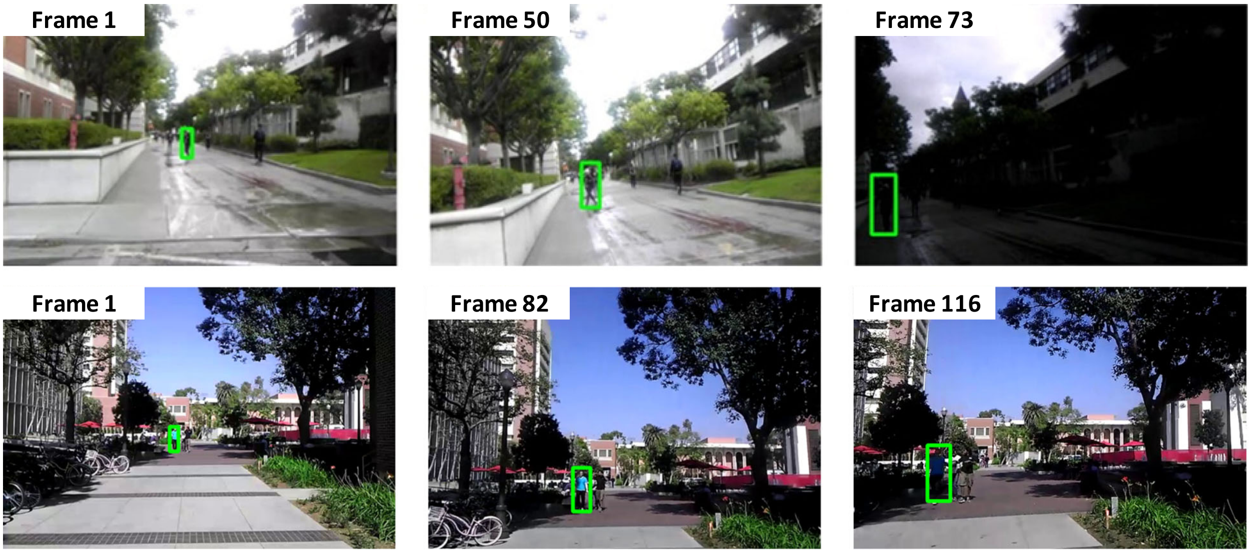


Fig. 13 Results of target tracking within cameras

geo-coordinates of the target based on the pixel coordinates on the image.

From the metadata of each video frame, camera tilt θ_x can be obtained. Camera optical center v_c is then computed as follows [8]:

$$v_c = 2f \cdot \tan(\theta_x/2) + v_0 \tag{4}$$

where f is the focal length of the camera and v_0 is the vertical position of the horizon line. We assume that the mobile videos are captured by people standing on the ground. Therefore, the focal length f is typically 1.4 times the image height [8]. The vanishing points are estimated at each video frame to get v_0 [26]. The ground distance between the mobile camera to the target is then given by

$$z = \frac{f y_c}{f \sin\theta_x - (v_c - v_b)\cos\theta_x} \tag{5}$$

where y_c is the altitude of the camera. v_b is the foot location of the target in image coordinate. We assume that the altitude of the camera equals human's average height which is 1.7 m.

Based on the distance/angle from the mobile camera to the target, the latitude and longitude of the target can be obtained. Finally, the trajectories of the target are smoothed by Kalman filtering (KF).

5 Experimental results and discussions

Using the persistent tracking system introduced in Section 4, we carry out experiments on synthetic datasets and real-world datasets to evaluate the effectiveness and the efficiency of the proposed GIFT. Here, we aim to answer the following questions:

- How effective is GIFT at filtering out unrelated video segments? (performance)
- How complete and accurate are the video segment results returned by GIFT that may cover the target? (accuracy)

To answer the first question, we use the communication load as the metric to evaluate the performance of GIFT. To define the completeness and accuracy of the search results returned by GIFT, we use precision and recall, which are two widely used metrics in image retrieval.

5.1 Experimental methodology

We assume in our persistent tracking experiments that targets are moving freely over an area. Videos are captured as people move in a casual manner (e.g., mobile videos from smartphones) and include various shooting directions. GIFT is applied to select video segments to reacquire the target when the tracking loses the target or the target exits from the current video.

We introduce two baseline methods for comparison. For both baseline methods, the camera viewable scene is described as a circular region centered around the camera location. In this case, the camera view direction is not known. In the remainder of this paper, we call this circular viewable region *CircleScene*. More specifically:

- Baseline 1: Each video clip is represented by a *single* camera location with a *CircleScene*. The location of the first video frame defines the location for the entire video.
- Baseline 2: Each video clip is represented by a *sequence* of camera locations with associated *CircleScenes*.

Figure 14 schematically depicts these two baseline methods showing the associated *CircleScene* camera representations (marked as P_1, P_2, P_3) and the predicted locations of targets (marked as $P'_{o,1}, P'_{o,2}, P'_{o,3}$). The rationale for using a single *CircleScene* representation for the entire video in baseline 1 is motivated by the fact that many online media management systems, e.g., YouTube, use a single camera location to represent an entire video. On the contrary, baseline 2 provides a fine-grained spatial resolution by associating *CircleScene* to each frame. Geospatial search results for both baseline methods rely on a *range query* that returns video segments for videos frames which *CircleScene* intersects the predicted location of the target.

The predicted target location $P'_{o,i}$ is estimated using a linear model from the last observed target location $P_{o,i}$ (recorded when the tracker loses the target) by

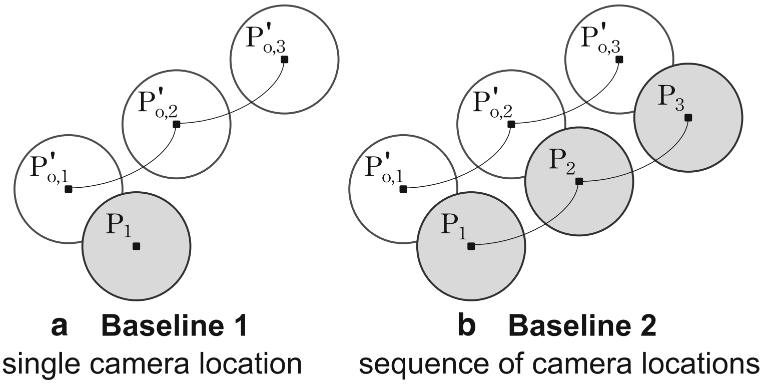
$$P'_{o,i} = P_{o,i} + v_o * (t'_o - t_o) \;\; 0 \le t'_o - t_o < \tau \tag{6}$$

where v_o is the target velocity estimated over the last 10 frames. We return video frames within a temporal range of $[\,t_o, t_o + \tau]$ which may cover the target with τ = 100 frames in all experiments. For all experiments, the radius *CircleScene* model and the radius of the range query (denoted by r in Eq. 1) in GIFT and the baseline methods are set to 0.1 km. The influences of different query radiuses are discussed. We use the same parameter settings in the baseline methods as GIFT for a fair comparison.

If the target exits the view of one video, GIFT and baseline methods return FOVs which may observe this target. Consecutive FOVs are grouped into video segments for target reacquisition and tracking. We eliminate the video segments which are shorter than 5 s. In the synthetic dataset, one random video segment is chosen from the returned video segments to resume target tracking. In the real-world dataset, target reacquisition is performed in all the returned video segments to reacquire the target.

5.2 Evaluation metrics

Communication load, precision, and recall are used to evaluate the proposed GIFT. Without loss of generality, the communication load is defined as the number of FOVs



a Baseline 1
single camera location

b Baseline 2
sequence of camera locations

Fig. 14 Baseline methods with camera *CircleScenes* shown in *gray*. In baseline methods, range queries are performed. **a** *Baseline 1* single camera location. **b** *Baseline 2* sequence of camera locations

transferred over the network, which also represents the size of input data for vision applications. Precision and recall are used to evaluate the accuracy and the completeness of the proposed GIFT which are defined as follows

$$\text{Precision} = \frac{|\text{FOVR} \cap \text{GT}|}{|\text{FOVR}|}$$

$$\text{Recall} = \frac{|\text{FOVR} \cap \text{GT}|}{|\text{GT}|}$$

where |FOVR| is the number of FOVs returned from GIFT or the baseline methods. |GT| is the number of FOVs from the ground truth which covers a target. It should be noted that, in our experiments, all baseline methods and GIFT return FOVs based on the predicted location of the target since the target is not always visible to the cameras. On the other hand, |GT| is the number of frames where the FOVs truly cover the ground truth location of the target. To compute |GT|, we replace the predicted location of the target with the ground truth location.

Precision is the ratio of retrieved relevant FOVs to all retrieved FOVs. A lower value of precision implies the search result set contains a large number of invisible regions in tracking. Precision evaluates the accuracy of the query results. Recall is the ratio of retrieved relevant FOVs to all relevant FOVs in the dataset. A lower recall value means more regions that should be returned as visible are ignored. Recall evaluates the completeness of the query results.

Table 1 Summary of the synthetic dataset

Total # of FOVs	0.48M
Total # of videos	500
FOV # per second	1
Average length of the video (hours)	0.28
Average camera moving speed (km/h)	5
Average camera rotation speed (degree/s)	11
Viewable distance *R* (meters)	100
Viewable angle α (degrees)	60

5.3 Results with synthetic dataset

We generate synthetic mobile video metadata using the generation algorithm method in [30] to evaluate the proposed GIFT. The synthetic video metadata generated are of realistic geographical properties (e.g., series of FOVs of mobile videos) which is summarized in Table 1. The number of FOVs generated per second is 1. The FOVs are generated uniformly distributed around an area of 1 km by 1 km initially. We assume that the mobile videos are taken by 500 freely moving people walking randomly in the area at an average speed of 5.13 km/h [30]. Therefore, there are altogether 500 camera trajectories. Figure 15 shows a visualization of the synthetic camera trajectories and FOVs from mobile videos. Different camera trajectories are represented by different colors. For the purpose of visualization, only part of the trajectories and FOVs are shown in Fig. 15.

We also simulate target's trajectories using the generator used in [30] to generate twenty targets in the scene. As the targets move around in the scene, GIFT queries are issued to select FOVs which may cover these targets to achieve persistent tracking. Figure 16 shows a visualization of the returned FOVs from GIFT which may cover one target as the target moves around in the scene. The trajectory of the target is plotted with red in Fig. 16. FOVs returned by GIFT at different query time are plotted with different colors. For the visualization purpose, part of the target's trajectory and FOVs returned from GIFT are shown.

First, we perform tracking for a single target for the entire testing duration of 0.28 h (around 17 min). During this tracking, 17 queries are issued. The query radius is set to 0.1 km. Figure 17 shows the results of the accumulated

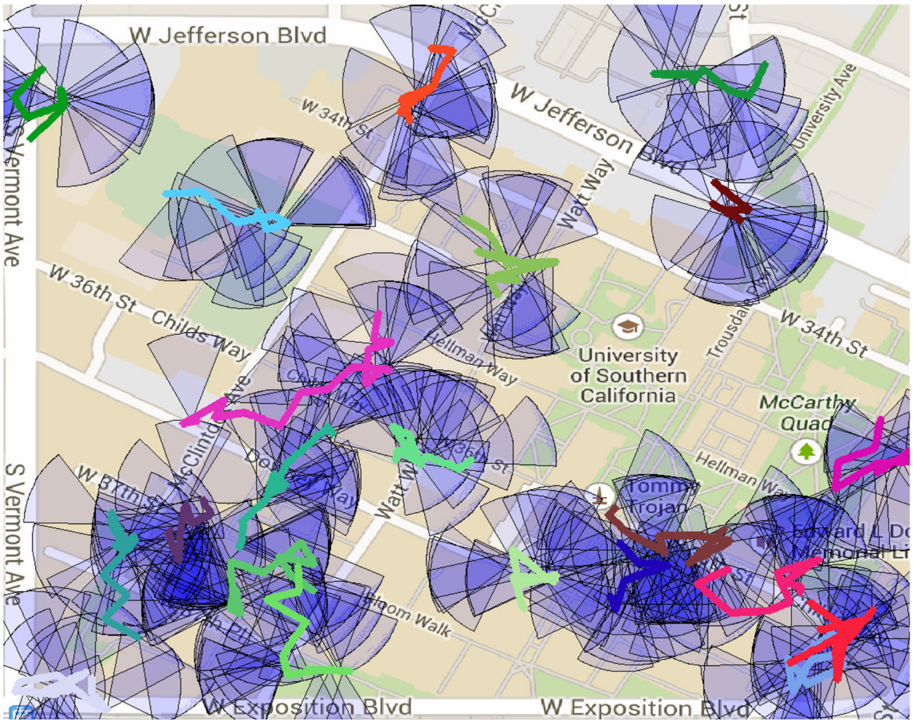


Fig. 15 Visualization of the synthetic camera trajectories and FOVs. For the visualization purpose, only part of the trajectories and FOVs are shown

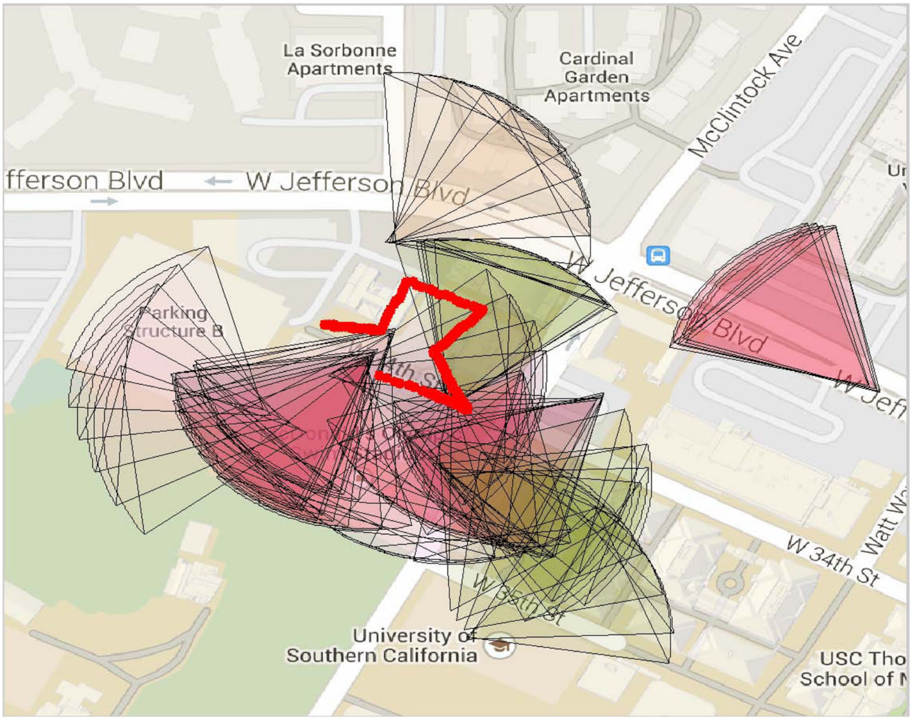


Fig. 16 Visualization of the FOVs which may cover the target as the target moves around in the scene

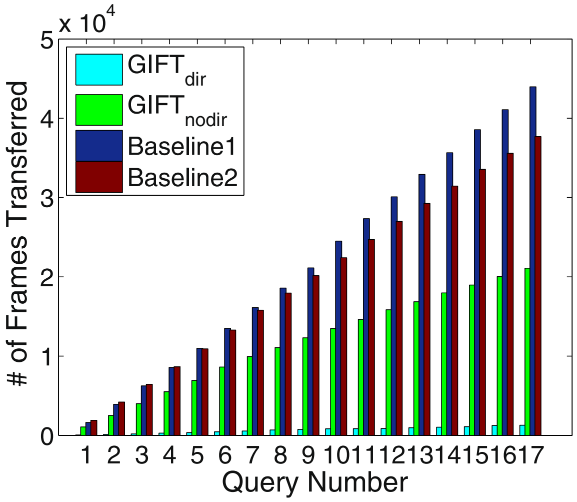


Fig. 17 Accumulated communication loads of GIFT and baseline methods from tracking one target

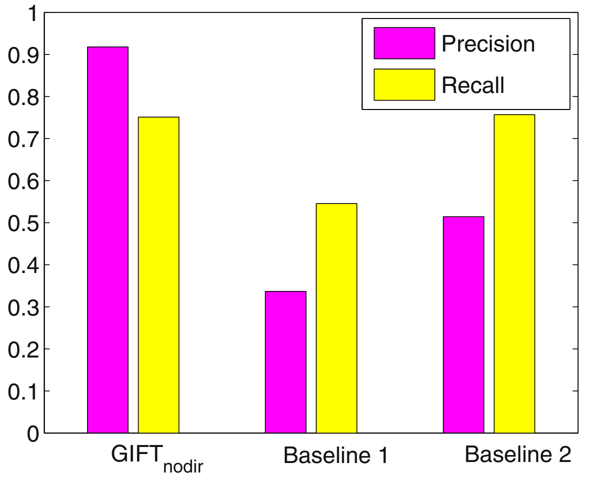


Fig. 19 Average precision and recall after tracking 20 targets

communication loads (e.g., the number of returned FOVs as result). The x-axis denotes the query number while the y-axis indicates the accumulated number of frames transferred over the network after xth query. In this experiment, we differentiate GIFT_{dir} with $\text{GIFT}_{\text{nodir}}$ depending on if direction is used in the experiments (Eq. 2). The direction margin δ in Eq. 2 is set to 15°. It is observed in Fig. 17 that by applying the direction query in Eq. 2, it is possible to get a consistent frontal view of the target while further reducing the number of frames transferred over the network. GIFT significantly reduces the communication loads over the network comparing to baseline methods, which improves the overall efficiency of the persistent tracking system.

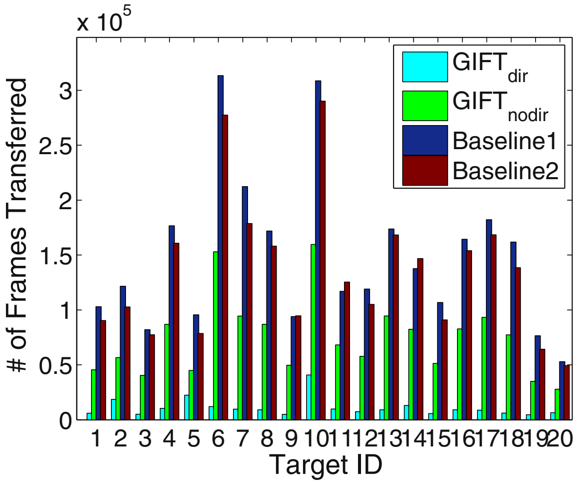


Fig. 18 Comparison of communication loads from tracking 20 targets

Next, we track 20 different targets and measure the communication loads for each target tracking case. Figure 18 shows that there are significant differences in communication load per target depending on the trajectory of each target and the amount of overlapping videos along a specific target trajectory. However, GIFT perform far better than baseline methods in all cases. On average, the number of transferred FOVs per a target tracking for 0.28 h is around 10.9K, 74.4K, 148K, and 136K for GIFT_{dir}, $\text{GIFT}_{\text{nodir}}$, baseline 1, and baseline 2, respectively. GIFT_{dir} demonstrate the best performance in all experiments; however, note that it is not always possible to apply GIFT_{dir} for all analysis because the frontal view of target may not be always available in videos. In our experiments with synthetic data, we assume that the frontal views are available. Nonetheless, GIFT_{dir} is promising when the right dataset is available.

Figure 19 shows the average precision and recall of the $\text{GIFT}_{\text{nodir}}$ and the baseline methods after tracking 20 targets. It is observed that the precision of GIFT is greatly higher compared with baseline methods and GIFT effectively filter out unrelated FOVs while returning a reasonably complete set of the FOVs which may cover the target. Recall greatly depends on how well the algorithm predicts the location of a target. We use a simple constant velocity method in the simulated dataset. Any advanced

Table 2 Impact of query radiuses in $\text{GIFT}_{\text{nodir}}$

Query radius	Communication loads	Precision	Recall
0.05 km	0.62M FOVs	0.82	0.68
0.1 km	1.48M FOVs	0.92	0.75
0.15 km	2.68M FOVs	0.93	0.77

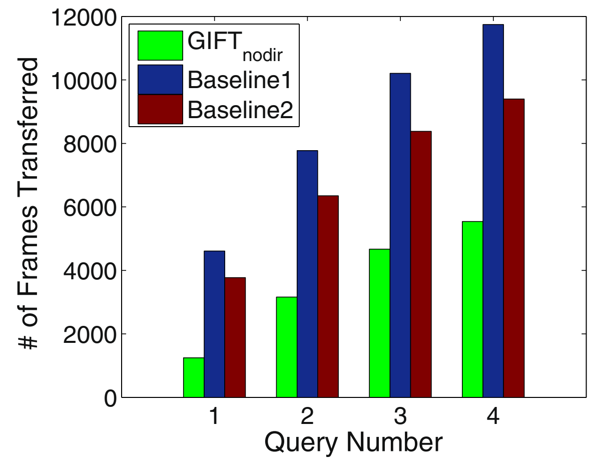


Fig. 20 A comparison of accumulated communication loads using real-world dataset

prediction can enhance recall, which is beyond the scope of this paper.

We also evaluate the impact of query radius on the performance of $\text{GIFT}_{\text{nodir}}$. Table 2 summarizes the performance of different query radiuses when we apply GIFT for persistent tracking. The average communication loads, precision, and recall using 20 targets are shown here. We observe that with the increase of the query radius, more FOVs are returned, thereby increasing communication loads. The tradeoff depends on the accuracy of the predicted location. If the predicted location of the target is far from the ground truth location, e.g., the target suddenly changes the moving direction, querying a larger radius will still return FOVs which may cover this target while querying with a smaller radius will fail.

5.4 Results with real-world dataset

In the experiments with real-world dataset, we use MediaQ [14] to collect and store mobile videos with geospatial metadata. Real videos are taken using smartphones by five people with one target in the view. We perform a persistent tracking for a single target during 5 min. Similar to our synthetic data case, the result video segments of GIFT and baseline methods are transferred from MediaQ system to the persistent tracking system over the network. Consecutive video frames which are longer than 5 s are grouped into video segments for target reacquisition.

We first tag a target (a person) at the first frame of one video and persistently track the target for a certain space and time. If the tracker end, e.g., the tracker reach the last frame of the video or the tracker lose the target, GIFT is applied to select the next video segments which may cover the target. We track each person in the first $min(100, len(Vi))$ frames of each video segment returned from the GIFT module to reacquire the target.

Figure 20 shows the results of the accumulated communication loads using the real-world scenario. Note that in this experiments, we use 30 FOVs per second for an accurate analysis of images for tracking. We observe that GIFT effectively reduce the number of frames transferred over the network which in turn reduce the number of unnecessary target matching in the target reacquisition module. By reducing the number of unnecessary target matching, the accuracy of the tracking and reacquisition is improved. Our persistent tracking system successfully track the target using GIFT as demonstrated in Fig. 21. Figure 21 shows the snapshots of our persistent tracking system while the target moves around in the scene. The main focus of this paper is GIFT and its applications to computer vision applications. Therefore, a thorough description and evaluation of the used tracking algorithm is beyond the scope of this paper.

6 Conclusions

In this paper, we presented a novel geospatial image and video filtering tool (GIFT) to select the most relevant set of input images and videos for computer vision applications. For the evaluation of GIFT, persistent tracking application was introduced and evaluated using both synthetic and real dataset. We observed a significantly

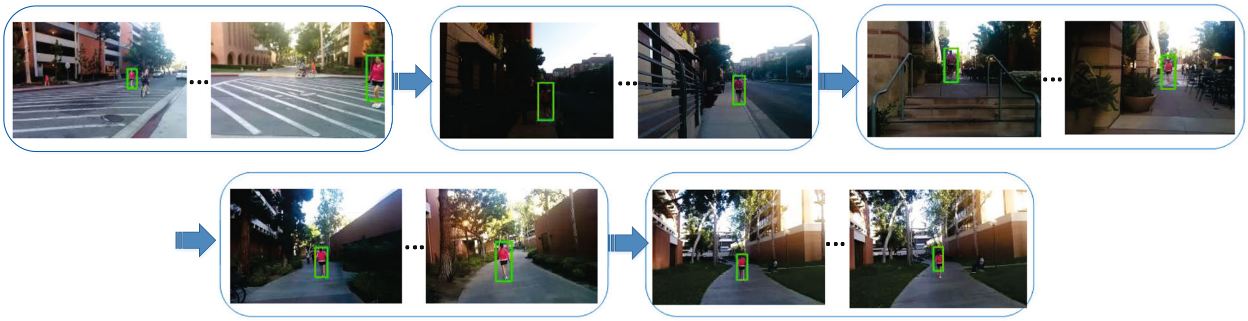


Fig. 21 Snapshots of the persistent tracking system

improved performance of computer vision application in terms of lower communication load, an improved efficiency, and accuracy to handle a large amount of video data. Our results showed that carefully selected input video frames using spatio-temporal property could expedite the total processing time significantly without impacting the accuracy of the result.

GIFT can be extended to serve as a general purpose tool for efficient video management which can be applied to any kind of media type with geospatial properties. Consequently, GIFT can be tied and used with many computer vision and multimedia applications in the presence of geo-tagged images and videos.

Funding

This research has been funded in part by NSF grant IIS-1320149, CNS-1461963 the USC Integrated Media Systems Center (IMSC), National Natural Science Foundation of China #61503381, and unrestricted cash gifts from Google and Northrop Grumman. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of any of the sponsors such as the National Science Foundation.

Authors' contributions

YHC and YL drafted the main part of the manuscript. YHC carried out the experiments. CS, SHK, and LN conceived of the study, participated in the design of the experiments, and helped modify the manuscript. All authors read and approved the final manuscript.

Competing interests

The authors declare that they have no competing interests.

Author details

[1]Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China. [2]Integrated Media Systems Center, University of Southern California, Los Angeles, CA, 90089, USA.

Received: 30 August 2016 Accepted: 29 January 2017
Published online: 23 February 2017

References

1. Y Cai, G Medioni, in *Proceedings of IEEE Winter Conference on Applications of Computer Vision*. Exploring context information for inter-camera multiple target tracking (IEEE Computer Society, Washington, 2014)
2. D Crandall, L Backstrom, D Huttenlocher, J Kleinberg, in *Proceedings of International Conference on World-Wide Web*. Mapping the world's photos (IW3C2, Switzerland, 2009)
3. H Detmold, A van den Hengel, A Dick, A Cichowski, R Hill, E Kocadag, K Falkner, DS Munro, in *Proceedings of International Conference on Distributed Smart Cameras*. Topology estimation for thousand-camera surveillance networks (IEEE Computer Society, Washington, 2007)
4. T Dinh, N Vo, G Medioni, in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. Context tracker: exploring supporters and distracters in unconstrained environments (IEEE Computer Society, Washington, 2011)
5. A Gilbert, R Bowden, in *Proceedings of European Conference on Computer Vision*. Tracking objects across cameras by incrementally learning inter-camera colour calibration and patterns of activity (Springer, Berlin, Heidelberg, 2006)
6. F Gilboa-Solomon, G Ashour, O Azulai, in *Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, SIGSPATIAL'13*. Efficient storage and retrieval of geo-referenced video from moving sensors (ACM, New York, 2013), pp. 404–407
7. A Guttman, in *Proceedings of ACM SIGMOD*. R-trees: a dynamic index structure for spatial searching (ACM, New York, 1984), pp. 47–57
8. D Hoiem, AA Efros, M Hebert, Putting objects in perspective. Int. J. Comput. Vis. **80**(1), 3–15 (2008)
9. C Huang, R Nevatia, in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. High performance object detection by collaborative learning of joint ranking of granules features (IEEE Computer Society, Washington, 2010)
10. O Javed, Z Rasheed, K Shafique, M Shah, in *Proceedings of International Conference on Computer Vision*. Tracking across multiple cameras with disjoint views (IEEE Computer Society, Washington, 2003)
11. O Javed, K Shafique, M Shah, in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. Appearance modeling for tracking in multiple non-overlapping cameras (IEEE Computer Society, Washington, 2005)
12. K Jungling, M Arens, in *Proceedings of IEEE Conference on Advanced Video and Signal-based Surveillance*. View-invariant person re-identification with an implicit shape model (IEEE Computer Society, Washington, 2011)
13. SH Kim, A Sakire, B Yu, R Zimmermann, in *Proceedings of ACM International Conference on Multimedia System*. Vector model in support of versatile georeferenced video search (ACM, New York, 2010), pp. 235–246
14. SH Kim, Y Lu, G Constantinou, C Shahabi, G Wang, R Zimmermann, in *Proceedings of ACM International Conference on Multimedia System*. MediaQ: mobile media management framework (ACM, New York, 2014)
15. SH Kim, Y Lu, J Shi, A Alfarrarjeh, C Shahabi, G Wang, R Zimmermann, in *Proceedings of International Symposium on Web and Wireless Geographical Information Systems*. Key frame selection algorithms for automatic generation of panoramic images from crowdsourced geo-tagged videos (ACM, New York, 2014)
16. Y Kim, J Kim, H Yu, Geotree: Using spatial information for georeferenced video search. Knowl.-Based Syst, 1–12 (2014)
17. D Lee, J Oh, W-K Loh, H Yu, GeoVideoIndex: indexing for georeferenced videos. Inf Sci. **374**, 210–223 (2016)
18. Y Lu, C Shahabi, SH Kim, in *Proceedings of ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, SIGSPATIAL'14*. An efficient index structure for large-scale geo-tagged video databases (ACM, New York, 2014)
19. Y Lu, C Shahabi, SH Kim, Efficient indexing and retrieval of large-scale geo-tagged video databases. GeoInformatica. **20**(4), 829–857 (2016)
20. X Lu, C Wang, J Yang, Y Pang, L Zhang, in *Proceedings of ACM Conference on Multimedia*. Photo2trip: Generating travel routes from geo-tagged photos for trip planning (ACM, New York, 2010)
21. H Ma, A Sakire, R Zimmermann, SH Kim, Large-scale geo-tagged video indexing and queries. GeoInformatica. **18**(4), 671–697 (2014)
22. D Makris, T Ellis, J Black, in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. Bridging the gaps between cameras (IEEE Computer Society, Washington, 2004), pp. 205–210
23. G Medioni, Y Cai, in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition Workshops*. Persistent people tracking and face capture over a wide area (IEEE Computer Society, Washington, 2014)
24. P Mildner, F Claus, S Kopf, W Effelsberg, in *Proceedings of the 5th Workshop on Mobile Video*. Navigating videos by location (ACM, New York, 2013), pp. 43–48
25. T Navarrete, J Blat, in *Proceedings of International Conference on geographic information science*. Videogis: segmenting and indexing video based on geographic information (Springer, Berlin, Heidelberg, 2002), pp. 1–9
26. M Nieto, L Salgado, in *Proceedings of SPIE*. Real-time robust estimation of vanishing points through nonlinear optimization, vol. 7724 (SPIE, Washington, 2010)
27. M Park, J Luo, RT Collins, Y Liu, Estimating the camera direction of a geotagged image using reference images. Pattern Recognit. **47**(9), 2880–2893 (2014)
28. E Reinhard, M Ashikhmin, B Gooch, P Shirley, Color transfer between images. IEEE Comput. Graphics Appl. **21**(5), 34–41 (2001)
29. A Sakire, R Zimmermann, SH Kim, in *Proceedings of ACM Conference on Multimedia*. Viewable Scene Modeling for Geospatial Video search (ACM, New York, 2008)
30. A Sakire, SH Kim, R Zimmermann, in *Proceedings of ACM SIGSPATIAL GIS*. Generating synthetic meta-data for georeferenced video management (ACM, New York, 2010)
31. A Sakire, R Zimmermann, SH Kim, in *Proceedings of ACM International Conference on Multimedia System*. Relevance ranking in georeferenced video search vol. 16 (ACM, New York, 2010)
32. RE Schapire, Y Singer, Improved boosting algorithms using confidence-rated predictions. Mach. Learn. **37**(3), 297–336 (1999)

33. K Tieu, G Dalley, WEL Grimson, in *Proceedings of International Conference on Computer Vision*. Inference of non-overlapping camera network topology by measuring statistical dependence (IEEE Computer Society, Washington, 2005), pp. 1842–1849
34. K Toyama, R Logan, A Roseway, in *Proceedings of ACM Conference on Multimedia*. Geographic location tags on digital images (ACM, New York, 2003), pp. 156–166
35. TH Tsai, WC Jhou, WH Cheng, MC Hu, IC Shen, T Lim, KL Hua, A Ghoneim, MA Hossain, SC Hidayati, Photo sundial: estimating the time of capture in consumer photos. Neurocomputing. **177**, 529–542 (2016)
36. K Whang, R Krishnamurthy, in *Proceedings of International Conference on Database Systems for Advanced Applications*. The multilevel grid file—a dynamic hierarchical multidimensional file structure (Springer, Berlin, Heidelberg, 1991), pp. 449–459