Event Geo-Localization and Tracking From Crowd-Sourced Video Metadata

Amit More
Dept. of Electrical Engineering
Indian Institute of Technology Bombay
Mumbai, India
134070016@iitb.ac.in

Subhasis Chaudhuri
Dept. of Electrical Engineering
Indian Institute of Technology Bombay
Mumbai, India
sc@iitb.ac.in

ABSTRACT

We propose a novel technique for event geo-localization (*i.e.* 2-D location of the event on the surface of the earth) from the sensor metadata of crowd-sourced videos collected from smartphone devices. With the help of sensors available in the smartphone devices, such as digital compass and GPS receiver, we collect metadata information such as camera viewing direction and location along with the video. The event localization is then posed as a constrained optimization problem using available sensor metadata. Our results on the collected experimental data shows correct localization of events, which is particularly challenging for classical vision based methods because of the nature of the visual data. Since we only use sensor metadata in our approach, computational overhead is much less compared to what would be if video information is used. At the end, we illustrate the benefits of our work in analyzing the video data from multiple sources through geo-localization.

CCS Concepts

•Mathematics of computing → Convex optimization; •Human-centered computing → Social media; *Smartphones;* **•Computing methodologies → Interest point and salient region detections; Tracking;**

Keywords

Event localization, Smartphone, Digital Compass, Optimization, GPS

1. INTRODUCTION

The smartphone users generate a huge amount of data in the form of images and videos. This data is often stored on cloud-servers like Youtube, Vimeo, Flickr and Google-Panoramio for sharing and possible future usages. There are various methods available in the literature for processing such a video information, for example, video summarization, object detection and tracking, action recognition, event detection and structure recovery. These methods rely on visual information captured by cameras.

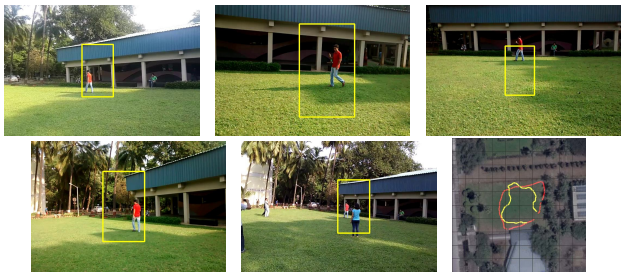

Figure 1: The estimated event track using metadata from the videos from five users is shown. Event location within the frames is highlighted using a window. Corresponding event track (yellow) and ground truth (red) are shown overlaid on Google Map.

However, all smartphones are equipped with a variety of sensors these days. Therefore, it is possible to capture various metadata information from sensors like GPS, digital compass, inertial measurement unit (IMU) and time stamps, along with videos. Use of this metadata information along with the camera often results in an improved performance for certain specific tasks, like videos and images can be grouped and retrieved using geo-tags for region based content retrieval systems [5, 2]. In [12, 15], collections of geo-tagged photos from Flickr and Panoramio are used to find popular tourist destinations. Metadata information such as GPS and time at which a photo was taken is used to build a tourist recommendation system. In [8], a query image from the user is compared with representative images found for a given region from geo-tagged images to produce tourist recommendations. Different modalities of geo-tagged media and their applications in event or landmark understanding, recognition, summarization, media organization and retrieval and data mining have been discussed in [16]. In [18], visual summaries of a given geographical region are produced using geo-tagged photos. In [10], popular regions and objects are detected and geo-located from crowd-sourced video metadata based on densities of camera viewing directions. In [4], sensor metadata from GPS, IMU, and compass is used along with visual data to improve user localization for an augmented-reality application. In [21], orientation information from compass is used to create rotation aware feature descriptor which are used for 3-D tracking of camera. Metadata from IMU has been used for improvement in SIFT feature matching in [13]. Different ways of fusing IMU data with the video for 3D tracking in an ex-

ICVGIP, December 18-22, 2016, Guwahati, India

DOI: http://dx.doi.org/10.1145/3009977.3009993

tended kalman filter framework is presented and an improvement in the accuracy of tracking have been demonstrated in [9]. Improved structure-from-motion (SFM) methods using metadata have been proposed in [11, 17]. GPS and IMU measurements are used as a prior to speed up structure recovery in [11]. A fast SFM algorithm is proposed in [17] using data from IMU. Key frame selection based on sensor metadata is proposed for structure recovery from videos in [19].

Thus one can be sure that the sensor metadata often captures information which can be used effectively to improve performance of the existing systems in many ways. With the increasing use of smartphones, we observe that many popular events, such as a concert, an exhibition or a parade, are often video recorded by many people at the same time. In general, one can define an event to be a moving or a stationary object which is recorded by many people at the same time. For example, Fig. 1 shows an illustration of an event recording scenario where five different users are video recording the event (a person in this case). Such collection of videos can be processed for event detection and tracking using classical algorithms. However, processing multiple views from different uncalibrated cameras is computation intensive and mathematically very difficult as camera poses, locations, color balances and mappings are all unknown. In case of Fig. 1, we have observed that there are hardly any feature matches among different views for detailed mathematical analysis. Since sensors can reveal some information about the camera such as location and orientation, this additional information can be leveraged.

The sensor metadata, however, is highly corrupted with the noise. In practical scenarios, users may not record the event properly due to human errors and distractions. Some users are not interested in a particular event and are recording something else. Under these conditions, utilizing metadata for event localization is a challenging task, hence, a proper mathematical formulation is necessary. We propose a mathematical formulation to geo-locate and track the event using sensor metadata associated with the crowd sourced videos. We rely on the location and orientation metadata obtained from the sensors available in common smartphone devices. Since we do not process any video data, the computational requirement is negligible.

Specific contributions of our work are as follows:

1. We pose event geo-localization as a convex optimization problem with the help of sensor metadata,
2. We provide a framework for handling sensor noise and detecting instantaneous camera locations and orientations,
3. We also provide a technique for smooth tracking of the event over a given period time,
4. Finally, we demonstrate the usefulness of the metadata in simplifying a typical computer vision problem.

2. PROBLEM FORMULATION

We use (C_{x_j}, C_{y_j}) to denote the measured location and Φ_j to denote the measured orientation of the j^{th} camera in a 2-D plane (*i.e.* camera pointing/viewing direction on the ground plane, we will use terms viewing direction and orientation interchangeably) obtained from sensor metadata. We

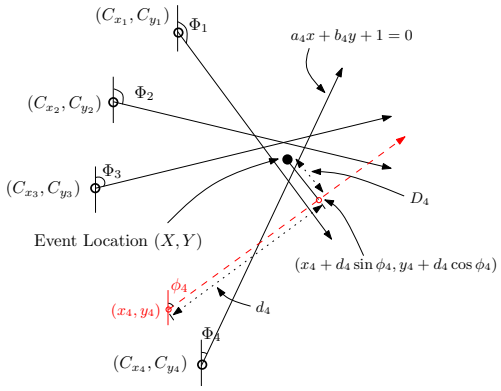

Figure 2: Cameras and their viewing directions are shown under the influence of noise. For the noise-free case and when the users are engrossed along the true viewing direction, all rays converge trivially at the event location (X, Y). Dashed red line shows viewing direction corresponding to the estimated parameters (x_j, y_j, ϕ_j) for one of the cameras (4^{th} camera in this case). The perpendicular distance D_j from viewing direction ϕ_j of the j^{th} camera to the event location (X, Y) is shown. d_j denotes the distance between a camera location and perpendicular from event location (X, Y) on viewing direction.

denote this metadata of j^{th} camera by a tuple $(C_{x_j}, C_{y_j}, \Phi_j)$. The corresponding estimates of the underlying true camera location and orientation are denoted by (x_j, y_j) and ϕ_j, respectively. We denote the event location on the ground plane as the latitude-longitude data by (X, Y). The corresponding time varying quantities are given by augmenting with the argument i (for *e.g.* $C_{x_j}(i)$,$C_{y_j}(i)$ and $\Phi_j(i)$). The event localization problem can then be stated as follows

Given a set of N independent videos with noisy sensor metadata collections $(C_{x_j}, C_{y_j}, \Phi_j)$ for $j = 1, ..., N$, find the event geo-location (X, Y), and geo-location and orientation (x_j, y_j, ϕ_j) for each camera *i.e.* for $j = 1, ..., N$.

Similarly, the event tracking problem involves estimating all above quantities, (X, Y) and $\{(x_j, y_j, \phi_j)\}|_{j=1}^{j=N}$ over a time period $i = 1$ to T when the corresponding temporal measurements $C_{x_j}(i)$,$C_{y_j}(i)$ and $\Phi_j(i)$ are available.

3. COLLECTING SENSOR METADATA

All modern smartphones are equipped with a GPS receiver and a digital compass. However, there are significant errors in the measurements from these sensors in smartphone devices. For example, accuracy of GPS is limited because of multi-path reception, atmospheric attenuation, satellite location errors and clock drift. Digital compass in the smartphones relies on accelerometer and magnetic sensor. Fast dynamic motion of the camera and the magnetic interference from the surrounding are the major causes of noise in the digital compass. When an event is video recorded by various users, the location and orientation metadata along with *timestamps* can also be recorded using GPS and comapss. This information, *i.e.* video along with the metadata (GPS location, compass orientation and *timestamps*) can be uploaded to the cloud server for event localization.

4. PROPOSED APPROACH

Under noise free conditions and when all the cameras are

pointing in at the event, the event can be localized as the intersection of the rays originating from camera locations, with viewing directions given by camera orientation. Under ideal conditions data from only two cameras is enough to geo-localize the event. However, with the noisy metadata, camera locations and orientations are not known with enough accuracy. A typical scenario is shown in the Fig. 2 where all cameras are crudely pointing in the direction of the event. Hence one have to pose the event localization as an optimization problem.

Viewing direction for the j^{th} camera can be expressed using the equation of a line as $a_j x + b_j y + 1 = 0$ as shown in Fig. 2. We can find parameters a_j and b_j using GPS location (C_{x_j}, C_{y_j}) and orientation Φ_j as $a_j = \frac{-\cos\Phi_j}{C_{x_j}\cos\Phi_j - C_{y_j}\sin\Phi_j}$ and $b_j = \frac{\sin\Phi_j}{C_{x_j}\cos\Phi_j - C_{y_j}\sin\Phi_j}$. The perpendicular distance D_j from event location (X, Y) to the viewing direction of the j^{th} camera is given by

$$D_j = |(C_{x_j} - X)\cos\Phi_j + (Y - C_{y_j})\sin\Phi_j)| \ . \quad (1)$$

When viewing directions do not intersect at a single point, we define the event location to be a point that minimizes sum of square of perpendicular distances from all viewing directions. Thus event location can be defined as a minimizer of cost function

$$\arg\min_{X,Y} \sum_{j=1}^{N} ((C_{x_j} - X)\cos\Phi_j + (Y - C_{y_j})\sin\Phi_j)^2 \ . \quad (2)$$

However, such an estimate of the event is not reliable. Whenever there is a large error in the orientation Φ_j or in the GPS data (C_{x_j}, C_{y_j}), the corresponding estimate of event location is very poor. Hence we estimate camera parameters (location and orientation) and use these estimated parameters for event localization which will result in a more reliable and correct event location.

5. CONSTRAINED ROBUST ESTIMATOR

Since the camera metadata can be highly erroneous sometimes, it is required to generate an estimate of underlying true camera parameters (x_j, y_j, ϕ_j) for each camera which can be used for event localization. We estimate these parameters along with the event location using the same cost function in equation (2). We replace metadata tuple $(C_{x_j}, C_{y_j}, \Phi_j)$ from the cost function with the corresponding estimates (x_j, y_j, ϕ_j). We now minimize this cost functions with respect to (X, Y) and $(x_j, y_j, \phi_j)|_{j=1}^{N}$ instead of just (X, Y).

The cost function now gets modified to following

$$\arg\min_{\substack{X,Y,x_j,\\ y_j,\phi_j \forall j}} \sum_{j=1}^{N} ((x_j - X)\cos\phi_j + (Y - y_j)\sin\phi_j)^2 \ . \quad (3)$$

Since the terms in the above cost function involve only unknowns, it is an ill-posed problem and there are infinitely many possible solutions. To restrict the solution space and bring it close to the true solution, we can use metadata measurements. We use sensor metadata to develop constraints on (x_j, y_j, ϕ_j), thus restricting the range of solutions.

5.1 Constraining Camera Location

The GPS receivers available in smartphone devices also provide reliability of the current location as an accuracy parameter which can be treated as the variance of a Gaussian perturbation [1]. We denote the accuracy parameter of location for the j^{th} camera by σ_j. Therefore the probability of a true location (x_j, y_j) is given by a Gaussian Distribution with mean (C_{x_j}, C_{y_j}) and variance σ_j^2. We restrict the range of (x_j, y_j) within a circle of radius σ_j around (C_{x_j}, C_{y_j}). The true camera location lies in this region with 46% probability. This constraint can be expressed as

$$(x_j - C_{x_j})^2 + (y_j - C_{y_j})^2 \leq \sigma_j^2 \ . \quad (4)$$

5.2 Constraining Camera Viewing Directions

Unlike GPS, there is no available measure of accuracy for the orientation obtained from the digital compass. To the best of our knowledge there are no methods available to recover correct orientation from a single stand-alone uncalibrated compass. Based on our experience with the digital compass in the smartphones, we propose an empirical formula to estimate the amount of error in the orientation based on magnetic interference. Strength of the earth's magnetic field remains quite stable over a given geographical region. Any kind of interference will affect the strength of the field measured by the magnetometer. One can take deviation of the measured field from its standard value as a measure of error. We choose the inverse bell shaped curve to measure the error as

$$\Delta\Phi = min\{\pi, \phi_0 + \lambda e^{(\frac{h_e - h_m}{\sigma_h})^2}\} \ , \quad (5)$$

where h_e is the typical earths magnetic field strength, h_m is the strength of the field measured by magnetometer. Parameters ϕ_0, λ and σ_h are chosen appropriately. It may be noted that ideally $\Delta\Phi$ should be within a few degrees. Unfortunately, most commercially available sensors in smartphones are very prone to large errors [20]. We use $\Delta\Phi_j$ to define constraints on orientation estimation ϕ_j for the j^{th} camera as shown below

$$\Phi_j - \Delta\Phi_j \leq \phi_j \leq \Phi_j + \Delta\Phi_j \ . \quad (6)$$

The camera parameter estimates (x_j, y_j) and ϕ_j must satisfy the constraints from equation (4) and (6).

6. CONVEX OPTIMIZATION

Minimization of equation (3) is a non-convex problem because of involvement of transcendental terms. In this section, we introduce a slack variable and formulate a convex cost function. As shown in Fig. 2, let d_j denote the distance between camera location (x_j, y_j) and the projection of the event location (X, Y) on the viewing direction ϕ_j. This projection has a coordinate $(d_j \sin\phi_j, d_j \cos\phi_j)$ with respect to the camera location. The perpendicular distance from the event location to the viewing direction D_j is given as

$$D_j = \sqrt{(X - x_j - d_j\sin\phi_j)^2 + (Y - y_j - d_j\cos\phi_j)^2} \ . \quad (7)$$

Representing the co-ordinates $(d_j \sin\phi_j, d_j \cos\phi_j)$ by (x_{d_j}, y_{d_j}), the above equation can be written as

$$D_j = \sqrt{(X - x_j - x_{d_j})^2 + (Y - y_j - y_{d_j})^2} \ . \quad (8)$$

This representation does not involve trigonometric terms and hence is convex in terms of variables. The final convex cost function is

$$\arg\min_{\substack{X,Y,x_j,y_j,\\ x_{d_j},y_{d_j}\forall j}} \sum_{j=1}^{N} (X - x_j - x_{d_j})^2 + (Y - y_j - y_{d_j})^2 \ . \quad (9)$$

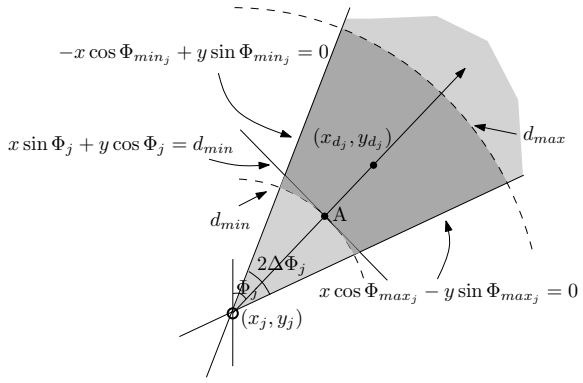

Figure 3: The gray region shows constraints on the viewing directions based on the orientation error $\Delta\Phi_j$. The arcs of the circle corresponding to maximum and minimum visible distances are also shown. The intersection of area between arc of the circles with gray region is shown as dark-gray region which corresponds to the constraints on (x_{d_j}, y_{d_j})

However, the variables x_{d_j} and y_{d_j} are not independent and one requires different constraints to solve the problem.

6.1 Redefining the Constraints

We have removed trigonometric terms from the cost function involving ϕ_j by introducing additional variables x_{d_j} and y_{d_j}. By construction, the point (x_{d_j}, y_{d_j}) lies on the camera viewing direction given by ϕ_j, which in turn is constrained by equation (6). Figure 3 shows constraints on ϕ_j as a cone with the vertex at (x_j, y_j), which can be seen as an intersection of two half-spaces shown as the gray region in the figure. Camera viewing direction must lie in this region. Since point (x_{d_j}, y_{d_j}) must also lie in the same region, we have following two constraints on x_{d_j} and y_{d_j}

$$-x_{d_j}\cos\Phi_{min_j} + y_{d_j}\sin\Phi_{min_j} \leq 0 \ , \tag{10}$$

$$x_{d_j}\cos\Phi_{max_j} - y_{d_j}\sin\Phi_{max_j} \leq 0 \ , \tag{11}$$

where $\Phi_{min_j} = \Phi_j - \Delta\Phi_j$ and $\Phi_{max_j} = \Phi_j + \Delta\Phi_j$.

We also put additional constraints on these variables knowing that the maximum visible distance (d_{max}) of a camera is limited to few tens of meters. Further, since the event cannot be closer to the camera than a certain distance, we consider a minimum visible distance (d_{min}) as well. Thus, following are two more constraints on these variables

$$d_{min}^2 \leq x_{d_j}^2 + y_{d_j}^2 \leq d_{max}^2 \ . \tag{12}$$

The first inequality in the equation (12) gives a non-convex constraints which corresponds to the exterior of a circle at (x_j, y_j) with a radius of d_{min} as shown in Fig. 3. This constraint is generally difficult to solve. Hence we approximate the arc of the circle by a straight line tangent at point A, as shown in Fig. 3. The corresponding constraint is

$$-x_{d_j}\sin\Phi_j - y_{d_j}\cos\Phi_j \leq -d_{min} \ . \tag{13}$$

The estimate must lie within the overall intersection area shaded dark gray in Fig. 3. We now summarize our final cost function and corresponding constraints. We minimize the cost function given in equation (9) with respect to the event location (X, Y) and parameters $(x_j, y_j, x_{d_j}, y_{d_j})|_{j=1}^{N}$ which are all unknowns.

$$\arg \min_{\substack{X,Y,\\ x_j,y_j,\\ x_{d_j},y_{d_j}\forall j}} \sum_{j=1}^{N} (X - x_j - x_{d_j})^2 + (Y - y_j - y_{d_j})^2 \ , \tag{14}$$

The constraints on the camera parameters are given by

$$(x_j - C_{x_j})^2 + (y_j - C_{y_j})^2 \leq \sigma_j^2, \ \forall j \ , \tag{15}$$

$$-x_{d_j}\cos\Phi_{min_j} + y_{d_j}\sin\Phi_{min_j} \leq 0, \ \forall j \ , \tag{16}$$

$$x_{d_j}\cos\Phi_{max_j} - y_{d_j}\sin\Phi_{max_j} \leq 0, \ \forall j \ , \tag{17}$$

$$x_{d_j}^2 + y_{d_j}^2 \leq d_{max}^2, \ \forall j \ , \tag{18}$$

$$-x_{d_j}\sin\Phi_j - y_{d_j}\cos\Phi_j \leq -d_{min}, \ \forall j \ . \tag{19}$$

So we have formulated convex cost free from transcendental terms by introducing a slack variable. The constraints were modified accordingly and approximations were introduced to have convexity in the constraints. Minimizing equation (14) is a standard constrained optimization problem. There are many algorithms available to solve this class of problems and interior-point methods are very popular approach [7]. We use python implementation of this algorithm *(cvxopt)* [3] for solving this problem.

6.2 Event Tracking

For the purpose of event tracking, one may solve the cost function in equation (14) at each time instant. Unfortunately, sudden large errors commonly associated with the metadata means weaker constraints on the camera parameters. As a result, the resultant event-track appears very discontinuous and noisy. This problem can be overcome by jointly minimizing the cost functions for each time instant along with smoothness terms so that resultant solution give a smooth event track.

We introduce an additional suffix i for all variables to denote time instant. We denote by T the total number of metadata samples. Sensor data sampled at the i^{th} time instant from the j^{th} camera is represented by a tuple $(C_{x_{ji}}, C_{y_{ji}}, \Phi_{ji})$ and corresponding unknowns are $(X_i, Y_i, x_{ji}, y_{ji}, x_{d_{ji}}, y_{d_{ji}})$.

The cost function from equation (14) now changes as

$$\begin{aligned}
&\arg \min_{\substack{X_i,Y_i,\\ x_{ji},y_{ji},\\ x_{d_{ji}},y_{d_{ji}}\\ \forall j,\forall i}} \sum_{\forall j,\forall i} (X_i - x_{ji} - x_{d_{ji}})^2 + (Y_i - y_{ji} - y_{d_{ji}})^2 + \\
&\lambda_1 \sum_{i=2}^{T} \{(X_i - X_{i-1})^2 + (Y_i - Y_{i-1})^2\} + \\
&\lambda_2 \sum_{\forall j} \sum_{i=2}^{T} \{(x_{ji} - x_{ji-1})^2 + (y_{ji} - y_{ji-1})^2\} + \\
&\lambda_3 \sum_{\forall j} \sum_{i=2}^{T} \{(x_{d_{ji}} - x_{d_{ji-1}})^2 + (y_{d_{ji}} - y_{d_{ji-1}})^2\}
\end{aligned} \tag{20}$$

The constraint equations (15-19) remain the same. The last three terms in the cost function impose smoothness on the event locations (X_i, Y_i) and camera parameters (x_{ji}, y_{ji}) and $(x_{d_{ji}}, y_{d_{ji}})$, respectively. When there are N cameras recording the event and T number of metadata samples, we have to solve for $(2T + 4NT)$ variables along with $5NT$ constraints which is a difficult task. To get around this problem we solve for the event locations successively for each time instant.

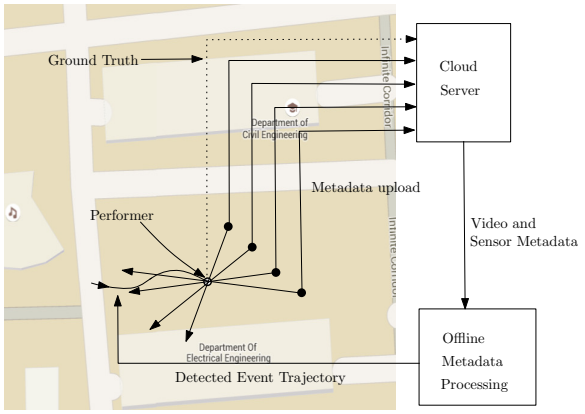

Figure 4: Illustration of generating metadata for a typical event recording scenario. The event location, event trajectory and camera locations (black dots) and orientations (black arrows) are overlaid on Google Map view.

7. SOURCING CROWD DATA

In order to validate the proposed method, one requires to identify a cloud where event video recordings by general crowd is available. Zimmermann *et al.* [2] have created and maintained one such data base. The videos and corresponding metadata are freely accessible through web portal and API's with the help of geo-location and time queries. To collect data, Zimmermann *et al.* have developed a smartphone application *GeoVid Recorder* for android and iOS platforms. The application records videos at 15-30 fps and collects sensor metadata at $5Hz$ frequency, which is then uploaded to the *GeoVid* server. There are more than 2500 videos uploaded to this server, with most of the videos being captured at different location and different time.

We create an event recording scenario at our university campus to collect experimental data along with the ground truth. In our experiments, we take an event to be a person. In a real scenario, an event could be a concert, a protest march or a sports-person on the field. We envisage these scenarios for the purpose of collecting ground truthed experimental data as shown in Fig. 4. A volunteer performs an event while carrying a smartphone and recording the metadata using the *GeoVid Recorder* application while other volunteers record the actual event (a volunteer) using the same application. Finally the data is uploaded to the server. The location ground truth is now made available from the GPS data of the performer.

8. EXPERIMENTAL ANALYSIS

8.1 Baseline Comparison

We use the work presented in [10] for a baseline comparison. In this work, all the viewing directions from the videos captured over large duration of time are considered, all together, for estimating the event location. The intersections between all pairs of viewing directions are computed and a clustering method is used over these points to localize the event and generate overall shape of the event. This approach is particularly suitable for estimating static events such as famous tourist locations and landmarks and needs a larger dataset. We modified this method to consider only those viewing directions which are generated at the same time. We compute intersection points for these viewing directions from all the cameras. In the original method, a clustering algorithm is used to get overall region of the event. However, since we are interested in the event location, we compute mean of the intersection points as event location estimate.

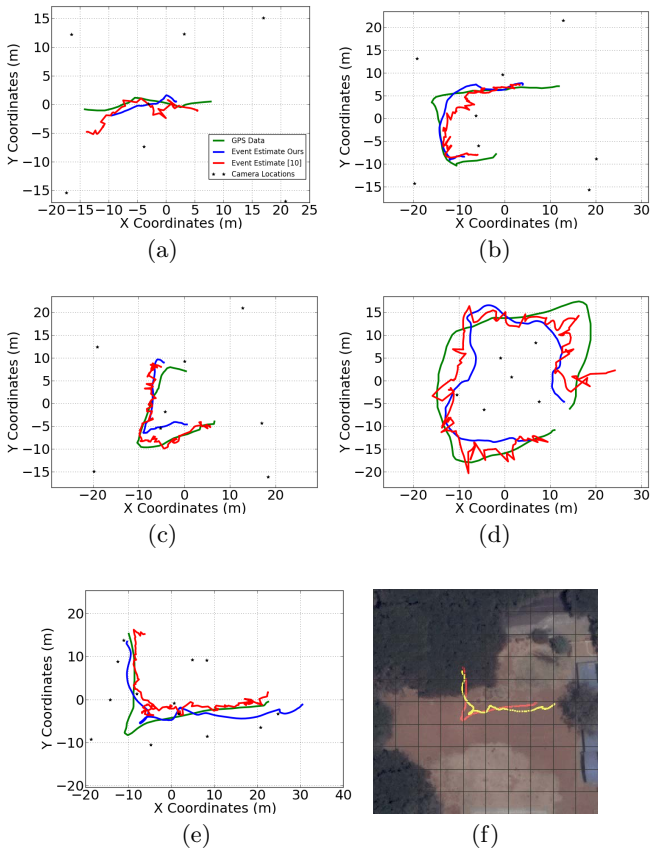

Figure 5: Estimated event trajectories using proposed method and [10] along with GPS data for event location are shown in (a)-(e). Average camera locations during the experiments are shown using black stars. Event trajectory is shown overlaid on the Google Map region of size $100m$x$100m$ in (f) for qualitative evaluation, the GPS data is shown in *red* and our estimation is shown in *yellow*.

8.2 Comparative analysis along with available GPS data for event location

We substantiate the performance of the proposed algorithm on various experimental data. We show the comparative analysis for the experiments where GPS track of the event was available to serve as a possible ground truth, details of which are shown in Table 1. Second and third rows of the table show the number of cameras recording the event and the common duration of the recording, respectively.

Exp. No.	1	2	3	4	5
No. of Cameras	6	7	7	5	12
Duration (s)	21	46	42	83	37

Table 1: Details of the experiments with the availability of the GPS track of the event is shown.

The estimated event tracks along with the available GPS track for the five experiments are shown in Fig. 5. The GPS trajectory is shown as green curve in the figure, estimated event tracks using proposed method are shown in blue and the same using [10] are shown in red. One may note that event tracks generated using [10] are very discontinuous where as proposed method naturally generated smooth event track. Furthermore, proposed approach also estimates camera parameters. The figures clearly show that our method is successfully able to generate the smooth event trajectory and is superior than [10].

In the experiment 4, all the cameras recording the event were also moving along with the event. For experiments 1-3, one camera had very noisy compass data which resulted in completely false camera orientation metadata, however, the correct orientation is recovered using proposed method. The experiment 5 had large number of outliers present which corresponds to human errors and distractions etc. Roughly 8 cameras in this experiment were pointed elsewhere (not *looking* at the event) for the duration ranging from 2 sec. to 13 sec. at different time instants. In Fig. 5e we see that correct event track is recovered, and it is clear that proposed method is quite robust to the outliers. The Fig. 5f show the estimated event track and GPS ground truth for experiment 5 overlaid on the Google Map for illustration, which is used for qualitative assertion of the estimated event track.

8.3 Analysis in absence of ground truth

Fig. 6 shows video frames from some of the cameras for the 4 experiments for which no GPS data for the event location was available. Each row in the figure corresponds to the individual experiment. First four images show views from some of the cameras for respective experiments and the last image show the estimated event track. The event is captured by 7, 10, 5 and 5 cameras and the duration of the experiments are 93, 63, 15 and 82 seconds, respectively. For experiments 3 and 4, the event is more or less stationary which is correctly estimated as can be seen in the last column of 3rd and 4th row. For both these experiments, there are 1 and 2 cameras, respectively, with very noisy compass metadata and a single camera outlier. For experiment 4, an outlier camera is constantly panning from left-to-right and from right-to-left many times for around 50 seconds during the experiment. One can notice that even in the presence of both outlier and noisy metadata, the event is successfully localized.

One can verify the correctness of event location with the help of available videos by computing instantaneous event position in the video frame. In order to locate the event within the video frame, one needs to know the position of the event in the camera field of view on the 2-D plane. We model the camera field of view using a pie-slice shape [6] with 50° as the camera field of view. Once an event is geo-localized, its position in the field of view is used to find the corresponding position of the event in the video frame using smaller window as shown in Fig. 6. The size of the window is chosen appropriately for these experiments and is varied as per the estimated distance between the camera and the event. We use these windows in the video frames to qualitatively verify that the event is correctly captured within these windows as seen in Fig. 6.

8.4 Validating Camera Parameter

(a) (b)

Figure 7: Estimated event tracks and the camera parameters for experiment 1 from Fig. 6 are shown in the figure. Camera locations are shown as green dots and viewing directions are shown as green arrows. The left figure shows event location estimates when camera parameters are fixed to the values of sensor metadata. In the right figure, both camera parameters and event location are estimated. Event track is shown in yellow.

In order to substantiate the validity of the estimated camera parameters, we compare estimated event track using proposed method with the track estimated without including camera parameters as part of the estimation process. The parameters are instead fixed to the values obtained from sensor metadata.

The resultant event track along with camera parameters (fixed to sensor metadata) for experiment 1 from Fig. 6 are shown in the Fig. 7(a). In Fig. 7(b), we show estimated event track using proposed method. The volunteer carrying out this event followed a path around the edges of the lawn which is recovered correctly as seen in Fig. 7(b). We also show the camera parameters at a particular time instant near the end of the experiment for both the cases. In the left figure, the camera parameters clearly appear to be little random because of noisy metadata, where as in the other figure, all the viewing directions correctly point at the event which is verified by inspecting the corresponding video frames.

8.5 Metadata Driven Image Analysis

In this section we demonstrate how the result of event tracking and geo-localization can help expedite the analysis of crowd sourced video data. As motivated earlier one may like to do a video analysis of the event (say, a street performer). To recognize the same performer in different views in absence of any other information, one can use image co-segmentation to extract the common object of interest in multiple views. Image co-segmentation is defined as a problem of segmenting an object from multiple images with similar features. We use the state-of-the art image co-segmentation method [14] on the video data and the results are shown in Fig. 8a. Here one captures predominantly the background, but fails to identify the performer (*i.e.* event). In Fig. 8c we show the results obtained using the window estimates shown in Fig. 8b with the help of event geo-localization (as illustrated in Fig. 1 and 6). One can notice that the background is drastically reduced compared to the earlier case and the event segmented successfully, which can then be used for further semantic analysis.

8.6 Computational Requirements

We calculate the computational requirements in terms of an average time taken to solve the cost function from equation (20) under respective constraints. The equation is

Figure 6: Experimental results when GPS data of the event location is not available are shown along the rows. Fig. (a)-(d) in a single row show some of the video frames for individual experiments. Event position in the frame is highlighted by a window using an approach described in section 8.3. In last column estimated event track along with camera parameters are shown.

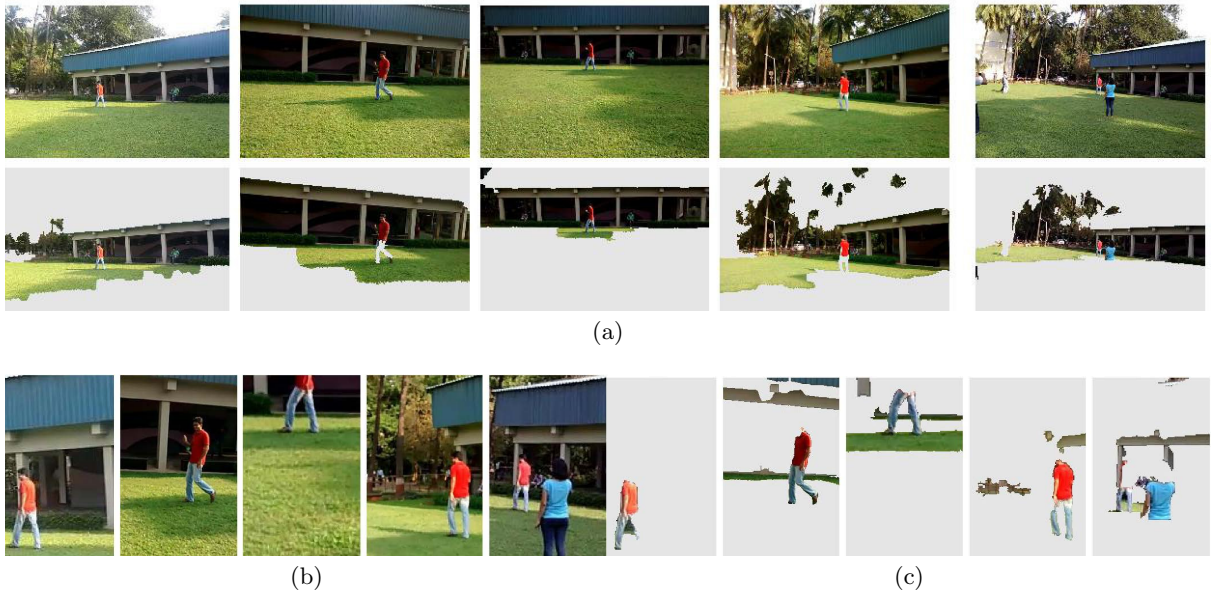

Figure 8: (a): First row shows the set of video frames from one of our experiments used for visual event identification. Second row shows co-segmentation results on these frames using [14]. (b): Cropped video frames from part (a) are shown. Frames are cropped based on the estimated window within the video frame using event localization. (c): Corresponding co-segmentation results for (b) are shown. One can notice a drastic reduction in the background.

solved for each time instant for the experiment and average time required to localize the event is computed. We have used intel core-i7 machine running at $3.4GHz$ with $32GB$ memory for our experimentation. The average time required

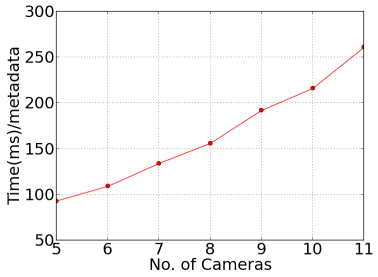

Figure 9: Computational time required for the event localization is plotted as a function of no. of available cameras.

is plotted in Fig. 9 as a function of the number of cameras. The plot show computational requirement grows linearly as a function of number of cameras. One can see from the figure that overall complexity is much less (100-300 msec) which suggests possible real-time application. Since the size of the metadata is very low, users can upload it to the cloud server in a real-time without much difficulties which can then be processed for event localization and a feedback to the users can be provided in a real-time such as how efficiently a user might be capturing the event.

9. CONCLUSIONS

We have shown that with the help of sensor metadata, the event localization can be posed as an optimization problem with a convex cost function and a set of convex constraints and can be solved with a reasonably good accuracy. The computational requirements are much less which is suitable for necessary real time applications. For the challenging visual data, such as videos recorded from smartphones, proposed approach is particularly useful where classical vision based approaches might fail or requires large amount of computations. In such cases, the proposed work can result in various hybrid algorithms where the performance of traditional algorithms can be improved in terms of accuracy and complexity by event localization using metadata.

The constraints used on the camera parameters are hard and might result in reduced accuracy of the event location due to sensor noise and bias. Such a limitation can be overcome by using a probabilistic framework. Our future work will focus on developing such a framework and extending it for augmenting with the visual analysis. There could be multiple events present at the same time within a small geo-neighborhood, such as in amusement parks. In such a case, since cameras are viewing different events, current approach may not be able to locate any of the events correctly.

10. REFERENCES

[1] http://developer.android.com.

[2] http://www.geovid.org.

[3] D. Andersen, J. Dahl, and L. Vandenberghe. Cvxopt: Python software for convex optimization, 2013.

[4] C. Arth, A. Mulloni, and D. Schmalstieg. Exploiting sensors on mobile phones to improve wide-area localization. In *IEEE Proc. ICPR*, pages 2152–2156, 2012.

[5] S. Ay, L. Zhang, S. Kim, M. He, and R. Zimmermann. GRVS: A georeferenced video search engine. In *ACM Proc. Multimedia*, pages 977–978, 2009.

[6] S. Ay, R. Zimmermann, and S. Kim. Viewable scene modeling for geospatial video search. In *ACM Proc. Multimedia*, pages 309–318, 2008.

[7] S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[8] L. Cao, J. Luo, A. Gallagher, X. Jin, J. Han, and T. Huang. A worldwide tourism recommendation system based on geo-tagged web photos. In *IEEE Proc. ICASSP*, pages 2274–2277, 2010.

[9] A. Erdem and A. Ercan. Fusing inertial sensor data in an extended kalman filter for 3d camera tracking. *IEEE Trans. Image Process.*, 24(2):538–548, 2015.

[10] J. Hao, G. Wang, B. Seo, and R. Zimmermann. Point of interest detection and visual distance estimation for sensor-rich video. *IEEE Trans. Multimedia*, 16(7):1929–1941, 2014.

[11] A. Irschara, C. Hoppe, H. Bischof, and S. Kluckner. Efficient structure from motion with weak position and orientation priors. In *IEEE CVPRW*, pages 21–28, 2011.

[12] K. Jiang, H. Yin, P. Wang, and N. Yu. Learning from contextual information of geo-tagged web photos to rank personalized tourism attractions. *Neurocomputing*, 119:17–25, 2013.

[13] D. Kurz and S. B. Himane. Inertial sensor aligned visual feature descriptors. In *IEEE Proc. CVPR*, pages 161–166, 2011.

[14] C. Lee, W.-D. Jang, J.-Y. Sim, and C.-S. Kim. Multiple random walkers and their application to image cosegmentation. In *IEEE Porc. CVPR*, pages 3837–3845, 2015.

[15] I. Lee, G. Cai, and K. Lee. Exploration of geo-tagged photos through data mining approaches. *Expert Systems with Applications*, 41(2):397–405, 2014.

[16] J. Luo, D. Joshi, J. Yu, and A. Gallagher. Geotagging in multimedia and computer vision-a survey. *Multimedia Tools and Applications*, 51(1):187–211, 2011.

[17] M. Ramachandran, A. Veeraraghavan, and R. Chellappa. A fast bilinear structure from motion algorithm using a video sequence and inertial sensors. *IEEE Trans. PAMI*, 33(1):186–193, 2011.

[18] S. Rudinac, A. Hanjalic, and M. Larson. Generating visual summaries of geographic areas using community-contributed images. *IEEE Trans. Multimedia*, 15(4):921–932, 2013.

[19] G. Wang, Y. Lu, L. Zhang, A. Alfarrarjeh, R. Zimmermann, S. Kim, and C. Shahabi. Active key frame selection for 3d model reconstruction from crowdsourced geo-tagged videos. In *IEEE Proc. ICME*, pages 1–6, 2014.

[20] G. Wang, Y. Yin, B. Seo, R. Zimmermann, and Z. Shen. Orientation data correction with georeferenced mobile videos. In *ACM Proc. SIGSPATIAL*, pages 400–403, 2013.

[21] L. Yu, S. Ong, and A. Nee. A tracking solution for mobile augmented reality based on sensor-aided marker-less tracking and panoramic mapping. *Multimedia Tools and Applications*, pages 1–22, 2015.