City Scale Geo-Spatial Trajectory Estimation of a Moving Camera

Gonzalo Vaca-Castano, Amir Roshan Zamir and Mubarak Shah
Computer Vision Lab, University of Central Florida
gonzalo@knights.ucf.edu,aroshan@cs.ucf.edu, shah@eecs.ucf.edu

Abstract

This paper presents a novel method for estimating the geospatial trajectory of a moving camera with unknown intrinsic parameters, in a city-scale urban environment. The proposed method is based on a three step process that includes: 1) finding the best visual matches of individual images to a dataset of geo-referenced street view images, 2) Bayesian tracking to estimate the frame localization and its temporal evolution, and 3) a trajectory reconstruction algorithm to eliminate inconsistent estimations. As a result of matching features in query image with the features in the reference geo-taged images, in the first step, we obtain a distribution of geolocated votes of matching features which is interpreted as the likelihood of the location (latitude and longitude) given the current observation. In the second step, Bayesian tracking framework is used to estimate the temporal evolution of frame geolocalization based on the previous state probabilities and current likelihood. Finally, once a trajectory is estimated, we perform a Minimum Spanning Trees (MST) based trajectory reconstruction algorithm to eliminate trajectory loops or noisy estimations. The proposed method was tested on sixty minutes of video, which included footage downloaded from YouTube and footage captured by random users in Orlando and Pittsburgh.

1. Introduction

The organization of visual data from the web based on location information has attracted a lot of interest in the past few years. Geospatial categorization of images have been available for years on websites such as Panoramio [19] and Flickr-map [11]. However, a similar classification for videos has not been developed yet. Such a classification would be of particular interest due to its potential effect on the user experience of browsing online video repositories. A main obstacle in establishing geospatial classification is the lack of detailed geolocation information associated with most of the videos currently available on the web. Additionally, while an image may have a single-spot geotag, a video contains more detailed information about the locations of different frames of videos in terms of geospatial *trajectory*, which represents the camera motion over the course of the video. In this paper, we address the problem of extracting the geospatial trajectory of the camera from unconstrained videos recorded in a city (see figure 1). The proposed method is intended for user-uploaded videos available on the web, which typically include undesired recording defects, such as blurred or uninformative frames, abrupt changes in camera motion, zooming effect, dynamical environment of the scene such as vehicles and pedestrians occluding distinct features, and lack of information of the initial position and pose (e.g. meta data) where the video was recorded.

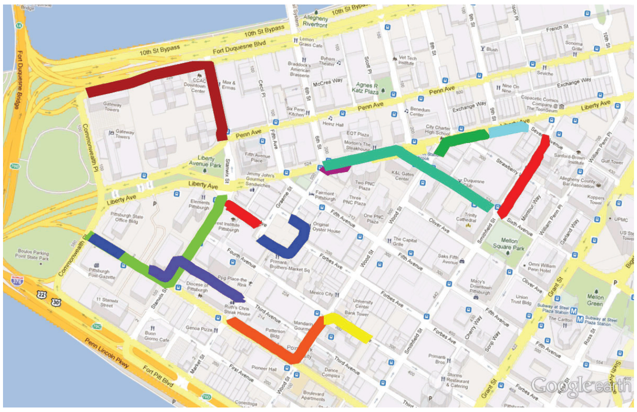

Figure 1. Geospatial trajectories of a subset of videos in our dataset in downtown Pittsburgh.

Visual Odometry (VO) and Visual SLAM (V-SLAM) are the two main research topics that focus on trajectory estimation from videos. Visual odometry (VO) is the process of estimating the egomotion of an agent using the single or multiple cameras connected to it. The term was originally coined by Nister in [18] as an ode to the wheel odometer on vehicles. Visual odometry is concerned only with local consistency (typically, over the last n poses) of the trajectory. Most methods assume some simplifying constraints such as having the camera attached to a vehicle, the availability of

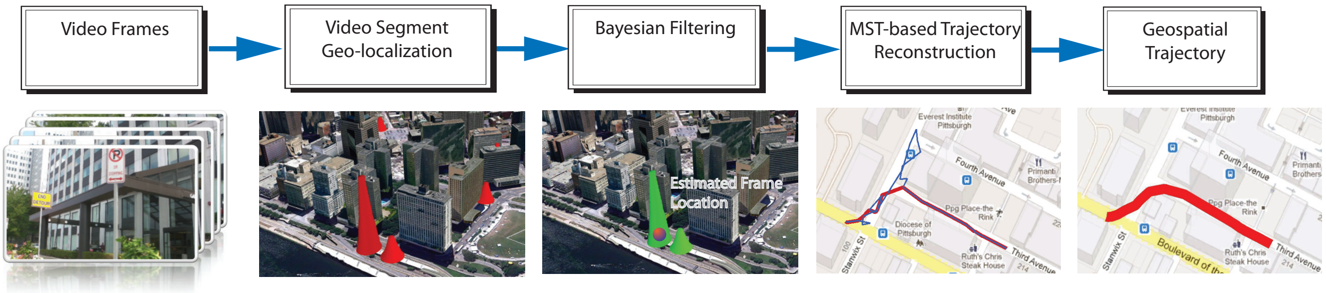

Figure 2. Schematic of our method for estimating the geospatial trajectory of a camera in a city.

additional sensors (e.g. IMU), or the use of omnidirectional cameras. For instance, Scaramuzza [22] used the Non-Holonomic constraint to reduce the number of correspondences in the Structure from Motion (SFM) problem. Tardif et al. [24] presented an approach for VO on a car using an omnidirectional camera which decoupled the rotation and translation estimation. Howard [10] proposed a method for simultaneous visual odometry and localization of the camera which is based on estimating the relative motion from successive stereo image pairs. Less constrained approaches such as Mouragnon and Lhuillier et al. [16], assume the use of a calibrated camera, which is not available in most of the naturally recorded videos. In Visual SLAM (V-SLAM) the objective is to incrementally build a consistent map of the environment while simultaneously determining its location on the map [6]. Two main categories of V-SLAM methods include: 1) those that use filtering (like EKF) to fuse the information from all the images with a probability distribution [1, 4], and 2) keyframe methods that retain the optimization of batch techniques, like global bundle adjustment to selected keyframes [12]. These methods also depend on calibrated cameras and are highly sensitive to outliers, such as those caused by vehicles or pedestrians, that effect the consistency of the map. Even assuming ideal conditions, such as calibrated cameras, robust point correspondences between frames, static scenery, and low accumulative error, the scale ambiguity still needs to be addressed and a global position in the map has to be determined. These approaches require finding a geometric relationship between either different frames of the query video or the query frames and some reference data [9, 10, 16]. Establishing this geometric frame-to-frame or frame-to-reference data relationship may be feasible for controlled environments, however such methods achieve limited success when applied to typical user-uploaded videos where difficulties such as frequent abrupt changes in camera motion, existence of uninformative or blurred frames, lack of meta data, and large area of localization is taken into account. A relatively different approach to image localization is developed in [25], which is primarily focused on indoor localization based on contextual features and providing priors for object recognition. Due to the high rate of failure of such methods on user-uploaded videos, we propose a different approach which does not require the traditional establishment of such geometric relationship explicitly. Our method leverages the temporal consistency of a video to extract its geospatial trajectory, instead of employing the more complex epipolar geometry based methods. Our method share some similarity with FAB-MAP [2, 3] in the sense that appearance-based place recognition is performed using a Bayesian framework. The main differences between our method and FB-MAP are: 1) FAB-MAP uses bag of visual words model to find similarity between images, while we use the localization method of [27]; 2) We propose a curve reconstruction algorithm to handle noisy estimations and unnecessary loops. Most of the videos being uploaded to online video repositories do not have a fine location tag, i.e. the availability of prior location information finer than city scale cannot be assumed. However, the capability of fine geolocalization in an area as large as a city is very useful. Several large scale reference data sets have been used for similar purposes to date [27, 23, 21, 26]. [21] utilizes a large reference data set of user-uploaded images. Such data sets are appropriate for locations which are heavily traveled by people. On the other hand, using Street View imagery [27, 23] which provides a uniform coverage of the area, has recently become popular. We used a data set of 10,000 GPS-tagged Street View images collected from downtown Pittsburgh, PA, and 6,620 images from downtown Orlando, FL, as the reference data [7, 27] in this work. In the subsequent sections of this paper, we will present a solution to the problem of estimating a city-scale trajectory of a moving camera with unknown intrinsic parameters based on the combination of image geolocalization, data association and tracking. We will also present an algorithm to remove noisy estimations of the predicted trajectory using Minimum Spanning Tree. We have tested the proposed algorithm on a data set of 45 videos, with the durations ranging from 60 to 120 seconds and total number of 106,200 frames. The query videos are from downtown Pittsburgh, PA and downtown Orlando, FL; they were downloaded from YouTube [8] or recorded by different users using a consumer grade video cameras while walking or driving in the city without prior knowledge about the usage of the videos.

2. Proposed Method

Figure 2 shows the steps of our method for estimating the trajectory of a moving camera in a city. Initially, frames are sampled periodically from the video. Each frame of a video is geolocalized according to the procedure described in [27]. The output of the individual frame geolocalization algorithm is a probability map with votes over the most probable locations, as described below. In [27], the highest peak in the probability map of votes is selected as the GPS location of the query image. Frame by frame estimation using this technique fails because video sequences typically contain many frames that are not assigned to the correct geolocation. Instead of using the individual estimation of the geolocation, utilizing the aforementioned procedure, we interpret the probability map votes as the 2D likelihood (with random variables of latitude and longitude) given the current frame observation. Thereby, multiple feasible hypotheses are considered for the current frame location, as opposed to a single specific frame position, which was mentioned previously. With multiple possible locations for each frame, the problem can be understood as a measurement association and single target tracking problem. Therefore, the next step in our method is a Bayesian tracking filter. The Bayesian tracking algorithm enforces the temporal consistency. In an analogy to tracking formulation, we set up a "range gate" where only votes inside the gate region are considered, while detections (votes) outside of the gate are ignored. Data association raises additional difficulties in this problem. Firstly, the geolocalization of individual frames based on visual features is often not accurate. As a consequence, the probability map tends to be very noisy. In fact, it is very common to find probability maps where the highest vote location does not correspond to the real location of the camera in the evaluated frame. Secondly, the size of the gate must be a large region in terms of the local position. The vote maps are associated with GPS tags that are sampled discretely. Then, the selected gate must be large enough to cover several locations of these geo-referenced tags, which can encompass hundreds of meters. Due to the aforementioned difficulties, the standard data association techniques cannot easily be adapted to obtain precise trajectories. For example, standard nearest neighbor filter will fail because the data is too sparse, which will produce noisy measurements. Moreover, splitting the track into multiple hypotheses every time more than one vote in the validation region is detected becomes impractical due to the large number of false alarms. Therefore, the trajectory estimation output from the Bayesian formulation is still noisy, particularly when the images are taken close to street corners where building façades look similar from both sides of intersecting streets, and where distant buildings get into the camera field of view, causing a false estimation that produces inaccuracies in the trajectory. Therefore, the final step of our approach is a trajectory reconstruction method which will eliminate loops and noisy estimations of the trajectory using our Minimum Spanning Tree (MST) based trajectory reconstruction algorithm. Each step of our method is explained in detail below.

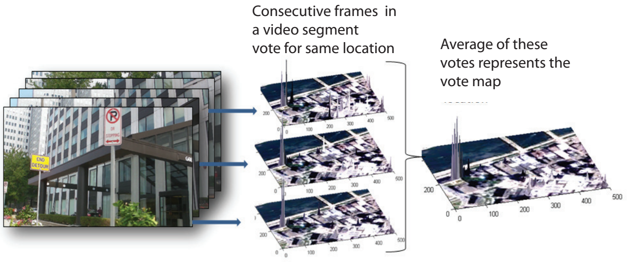

Figure 3. The GPS placemarks in the reference dataset are located approximately every 12 meters. The vote distribution of frames in a video segment during a period of time where the displacement was shorter than 12 meters are averaged since they are essentially voting for the same placemark.

2.1. Geolocalization of a Video Segment

We have chosen to use the method described in [27] as a baseline method for single image geolocalization since it produces a vote distribution instead of a single geolocation. In this method, interest points of the query image and reference images are described using SIFT descriptors [14]. For every SIFT descriptor of the query image, a set of nearest-neighbors is extracted from the reference database using a tree search [17]. Each of the these nearest-neighbors votes for their corresponding geographical position in the database, creating a map of votes for the city. Then, some of the votes are discarded according to the proposed criteria relating the proximity of the query descriptor to the matched descriptor and the geographical proximity of the set of nearest neighbors. The GPS location of the image corresponds to the highest peak found in the obtained map. Also, a *Confidence of Localization (CoL)* parameter, which can be used as a measure of the reliability of the estimation, is derived from the Kurtosis of the map. The GPS placemark locations in the reference dataset are spaced approximately every 12 meters. All the sampled frames from the query video corresponding to the time period where the camera moves 12 meters around a placemark, should vote for the same location in the reference dataset. This can be interpreted as a quantization process since we are constraining a continuous set of values (global position) to some discrete set of values (GPS placemarks). Processing individual frames will produce a quantization error in the frame position estimation. Therefore, it is more helpful to gather sets of consecutive frames and treat them as a video segment. Hence, a map of votes corresponding to a video segment is achieved by averaging the vote maps of each one of the frames that belong to the segment. Geolocalization of a video segment has also two positive side effects. The first is the enforcement of the most common vote locations in the segment of frames, which typically correspond to correct geolocations. The other positive side effect is the attenuation of votes at locations which fewer frames vote for, that typically corresponds to false alarms. Indirectly, geolocalization of video segments facilitates the data association.

2.2. A Bayesian Formulation

A Bayesian formulation is plausible, if the vote distribution of the video segment is interpreted as the likelihood of the location (latitude and longitude) given the current observation (see figure 4(a)). A video is constrained in the spatial and temporal domain because consecutive frames correspond to close spatial locations. Consequently, Bayesian tracking is used to estimate the frame localization and its temporal evolution. The objective is to estimate the state x (latitude and longitude) at any sampling time t.

Let x_t represent the state (latitude and longitude) at the time t, z_t represent the observation at the time t, Z_t represent the history of the observations $z_1, z_2, \dots, z_t$. We are interested in obtaining the distribution $p(x_t|Z_t)$, which describes the probability of the state x given the previous history of observations. It is evident that the distribution $p(x_t|Z_t)$ can be rearranged as $p(x_t|Z_t) = p(x_t|z_t Z_{t-1})$. Using the Bayes rule, we have:

$$p(x_t|Z_t) = \frac{p(z_t|x_t Z_{t-1})p(x_t|Z_{t-1})}{p(z_t|Z_{t-1})}. \tag{1}$$

The term in the denominator is not related to the variable x, it is simply a normalization constant that does not effect the probability distribution. Then, the denominator is replaced by a constant c to obtain:

$$p(x_t|Z_t) = \frac{p(z_t|x_t Z_{t-1})p(x_t|Z_{t-1})}{c}. \tag{2}$$

However, the observation process at the frame t is not related to the observation process at the previous time. Therefore, we can rewrite the previous equation as:

$$p(x_t|Z_t) = \frac{p(z_t|x_t)p(x_t|Z_{t-1})}{c}, \tag{3}$$

where $p(z_t|x_t)$ is the observation model or likelihood, and $p(x_t|Z_{t-1})$ is the predictive model of the process. The dynamical model is assumed to be a Markov model, which implies that the current state depends only on the previous state. This is an appropriate assumption when the previously estimated frame localization is correct. The marginalization of the probability distribution representing the predictive model becomes:

$$p(x_t|Z_{t-1}) = \int_{x_{t-1}} p(x_t|x_{t-1})p(x_{t-1}|Z_{t-1}). \tag{4}$$

Figure 4 illustrates this process. The term $p(x_t|x_{t-1})$ is the probability of the future state given the current state, which can be derived from the motion model of the camera (constant velocity model); the term $p(x_{t-1}|Z_{t-1})$ is the former probability of the state given the observation, which is available from the previous state estimation. The above equation can be substituted in equation 3 to obtain the probability of the state given the current observation. The constant of normalization c can be calculated as:

$$c = \int p(z_t|x_t)p(x_t|Z_{t-1}). \tag{5}$$

The normalization constant is also a measure of how often the number of votes estimated in the current state is close to the probability predicted from the previous state (gate). In other words, a value c close to zero could indicate false localization. Therefore, the value of the variable c is used in our algorithm to discard some of the untrustworthy observations. The state estimation in cases where c is close to zero are discarded, and a new probability function is built using the earlier state estimation as the most probable state. In the case that the value of c is close to zero in several consecutives estimations, a redetection process is performed, the same way the initial geolocation is computed, as described below. In the case where the values of c are not close to zero, the estimation of the state would be given by the expectation:

$$E[x_t|Z_t] = \int x_t p(x_t|Z_t)dx. \tag{6}$$

Figure 4(c) shows the result of the new state distribution after taking the product of the frame segment observation (figure 4(a)) and the state prediction (figure 4(b)) according to equation 3. The state localization for a frame segment at the time t is computed using equation 6, and is marked by the red marker in the figure 4(c).

Discrete version using a constant velocity model. Our experiments show that the constant velocity motion model performed slightly better than the constant acceleration and random walk/Brownian motion models in our method. A discrete version of the formulation can be implemented by defining the city map as a dense grid. The vote distribution which symbolizes the likelihood of the current state is represented as an array, as is the state given the observations and state prediction. The probability of the future

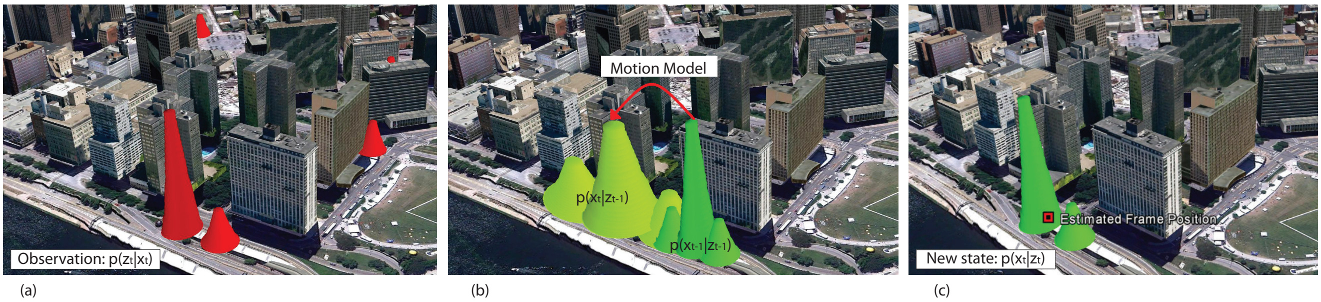

Figure 4. Bayesian estimation process. a) The observation is the vote distribution from a video segment. b) The prediction of the state(latitude,longitude) based on the previous state. c) The new state probability function computed using the state prediction and observation.

state given the current state $p(x_t|x_{t-1})$ is expected to be a shifted version (according to the constant velocity model) of the previous state distribution with some randomness added. A mathematical expression that fairly characterizes the state prediction given the precedent observations (4) is

$$p(x_t|Z_{t-1}) \approx U(x_t - (x_{t-1} + v_t)) * p(x_{t-1}|Z_{t-1}), \quad (7)$$

where U represents a uniform distribution centered around the origin, v_t is the velocity (shift) at time t, and the symbol $*$ represents a 2D convolution operator.
Finally, the state estimation for the latitude and longitude is obtained by using the discrete version of the expectation equation:

$$E[x_t|Z_t] = \int x_t p(x_t|Z_t) dx = \sum_i x_t^i p^i(x_t|Z_t). \quad (8)$$

Estimation of the Initial Geolocations

In order to obtain the initial geolocalization, we consider a group of periodically sampled frames around the first frame of the video. For each one of the sampled frames, its frame geolocalization is estimated as the highest peak of the vote distribution of the frame. It is highly probable that some of these frames are not correctly geolocalized; therefore, we have used two different pruning steps to remove them. The first step is to reduce the number of false geolocalizations using the information provided by the *Confidence of Localization (CoL)*, which was used in [27]. The estimated frame geolocalization is discarded by thresholding the *CoL* value. The second step is to discard the geographically isolated frame localizations by counting the number of frame geolocations within a prudent radius r of the frame being tested. The frame is discarded if the number of surrounding neighbors is less than the threshold. After applying these two pruning steps, the remaining frames are averaged to obtain an estimation of the initial geolocalization.

2.3. *Minimum Spanning Tree*-based Trajectory Reconstruction

Ideally, employing a sophisticated motion model which is capable of handling abrupt changes in the direction, zooming, tilting, lack of meta data, noisy frame-by-frame localizations, etc. in our Bayesian framework would yield a smooth and appropriate trajectory for a video. However, such motion model which is capable of addressing all aforementioned complications is not developed to date. Typically, any on-line (causal) approach to enforcing temporal consistency which exploits a motion model poses some *inertia* in motion estimation, due to the presumptions the motion mode is based on. Additionally, all the large scale image localization methods [27, 23, 21] which provide the input to the Bayesian Filter, are expected to geolocate a frame with an error of a few tens of meters. Although this error value is acceptable for a city scale localization algorithm, it can cause inconsistency in the trajectory that the video segments form, even after applying the Bayesian filter. For instance, the inertia of motion model along with an error value of a few tens of meters in the video segment locations can cause the trajectory to go straight at an intersection for at least a few video segments while the camera has actually made a turn. An example is depicted by the magenta contour in the figure 5(a). Video segment locations which slightly deviate from but are still close to the main stream of the trajectory result in another case of inconsistencies caused by the slight inaccuracy of individual frame localization method which Bayesian filter cannot effectively handle (depicted by black contour in figure 5(a)). These cases, along with other types of complications (e.g. inaccurate yet repeated frame locations, which are due to zooming and focusing on a nearby buildings) cause the extracted geospatial trajectory to possess special characteristics which can not be handled effectively by basic trajectory reconstruction or smoothing methods like moving average (MA). Therefore, we propose a trajectory reconstruction method based on Minimum Spanning Trees which can effectively handle

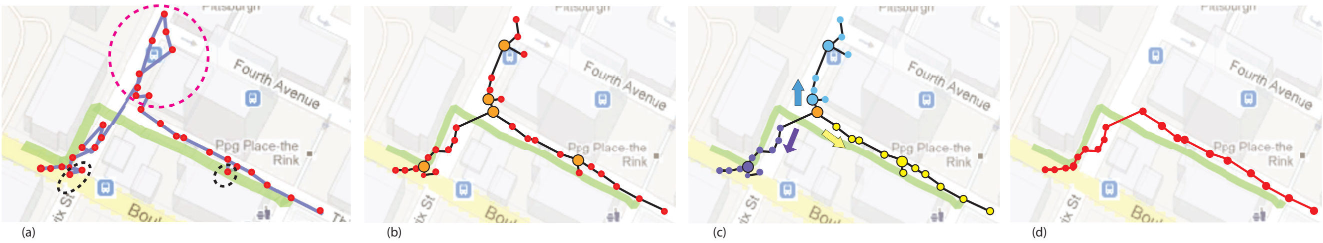

Figure 5. Illustration of the different steps of MST based trajectory reconstruction. The green trajectory represents the ground truth. a) Output of the Bayesian filter. b) Minimum Spanning Tree. The nodes with a degree higher than two are shown in orange. c) The branches of a particular node with a degree higher than two (shown in orange) are marked with arrows. Yellow and purple branches are retained and the blue one is removed as it has less weight. d) The final reconstructed trajectory.

these complications.

Minimum Spanning Trees(MST) have been used extensively in a variety of fields ranging from network design [20] to medical image analysis [15]. Ma. et al. [15] use MST in robust image registration. Perlman [20] utilizes MST for the efficient design of computer networks. MST has been used in curve formation as well. I. Lee [13] proposes a curve reconstruction method based on moving least square improved by MST. Figueiredo and Gomes [5] use MST to reconstruct differentiable arcs from dense samples. The reason behind the varied uses of MST is its characteristic ability to find a minimal way of linking some entities. In our case, these entities are the video segment locations acquired from the Bayesian filter. The proposed geospatial trajectory reconstruction method using Minimum Spanning Trees is described in Algorithm 1:

Algorithm 1 MST-Based Trajectory Reconstruction

```
1: Find the Minimum Spanning Tree of $G = (\mathbf{N}, \mathbf{E}, \mathbf{W})$
for $i$ where (degree of node $i$) $> 2$ do
    2: Set Root to node $i$.
    3: Set Weight of each branch connected to the Root to
    the number of nodes on it.
    4: Retain the two branches with higher weights and
    remove others.
end for
5: return Minimum Spanning Tree with retained nodes.
```

In Algorithm 1, the nodes, edges and cost of edges of the graph G are represented by $\mathbf{N}$, $\mathbf{E}$ and $\mathbf{W}$, respectively. Each video segment is represented by one node in $\mathbf{N}$. Each node has the feature vector (x_i, t_i), where x_i is the corresponding video segment's geolocation and t_i is its respective time obtained from Bayesian filter. E includes the edges between all possible pairs of nodes. The cost of each edge is defined as the Euclidean distance between the feature vectors of the nodes that edge connects.

The process of MST-based trajectory reconstruction is illustrated in figure 5. First, the output locations of the video segments and their respective time (fig. 5 (a)) are acquired from the Bayesian Filter and the graph G is formed. Then, the Minimum Spanning Tree of G is found (figure 5 (b)). The degree of a node in a MST is defined as the number of edges connected to it. The next step is to identify the nodes with a degree higher than two (orange nodes in figure 5 (b)). For such nodes, we define the weight of each connected branch as the number of nodes connected through that branch to the root. This is illustrated in figure 5 (c) for one of the nodes with a degree higher than two. Then, the nodes on the two branches with the highest weights are retained and the rest are removed. When a node with a degree higher than two is observed, it means there is a node which is likely off the mainstream of the trajectory and consequently an additional branch has appeared. Such a node is either geospatially or sequentially inconsistent with the rest of the path. The process of assigning a weight to each branch is intended to identify the branch(es) which contains an outlier and consequently should be removed. The branch which has fewer nodes that are connected to the root is less likely to be on the mainstream, since fewer video segment locations are consistent with its location. Therefore, we retain the two branches with highest weights, which ensures the connectivity of the trajectory, and remove the rest. The final trajectory is shown in figure 5 (d). Note that the features used to determine the MST, include both time and geolocation information. Therefore, if the camera revisits a previously visited location, the nodes corresponding to the first and second visit will not be mistakenly linked in the MST as their time features are very different even though their geospatial locations are close. An alternative algorithm to the one in line 3 of Algorithm 1 performs *breadth first search* with the root set to x_i and retains the nodes of the two *deepest* branches rather than those with the highest weights. However this method would be computationally more expensive than the original algorithm in line 3, yet it performs better if the branches, including the correct ones, are highly contaminated with outlier nodes.

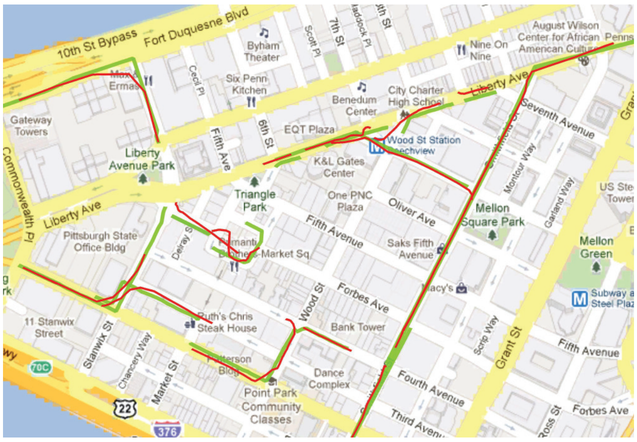

Figure 6. A subset of trajectories obtained from videos in downtown Pittsburgh. The green trajectories correspond to the ground truth, while the red ones correspond to our Bayesian framework + MST trajectory reconstruction.

3. Experimental Results

Figure 6 shows the trajectories obtained from videos recorded in downtown Pittsburgh using our proposed Bayesian filtering and MST based trajectory reconstruction. In the figure, the green lines represent the ground truth trajectories of the camera, while the red ones are the trajectories produced by our algorithm. These qualitative results corroborate that our algorithm is successful in obtaining the accurate trajectory of a camera in an area as large as a city (Note that the area covered by the dataset is larger than the frame shown in figure 6). Figure 7 shows examples of two trajectories obtained using Bayesian filtering, and their outputs after performing MST-based curve reconstruction.

Implementation details.
Each one of the videos is sampled every ten frames to produce a frame rate of approximately 3 frames per second (fps). Ten of these sampled frames are used to form a video segment, since the displacement of an object in a city is typically less than 12 meters in 3 seconds. In the reference dataset, Scale Invariant Feature Transform (SIFT) points are computed for each one of the Google street view images. The SIFT descriptors and their corresponding GPS tags are indexed in a tree using FLANN [17]. A map of votes for each query frame is calculated by computing SIFT descriptors in the query image, obtaining a list of nearest neighbors to the indexed features for each interest, and using the voting scheme previously described. The initial geolocalization estimation proposed in section 2.2 is employed to initialize the algorithm. The value of the *CoL* threshold is set to 40% and the radius r is set to 40 meters. The uniform function described in equation 7 was set to cover an approximate radius of 70 meters.

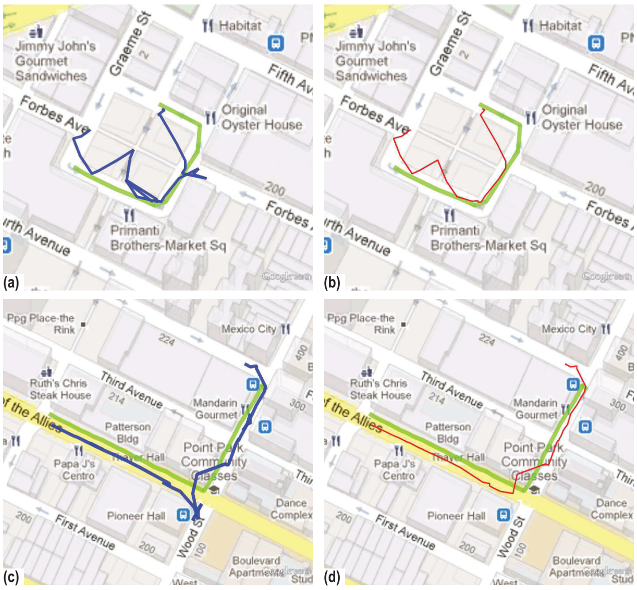

Figure 7. Two MST based trajectory reconstruction examples. The figures a) and c) correspond to the Bayesian filtering of the two examples. Figures b) and d) are the trajectories obtained after applying MST based trajectory reconstruction to the trajectories in a) and c).

Quantitative results.
The proposed method was evaluated using a test data set of forty five videos downloaded from YouTube or recorded by random users in Pittsburgh or Orlando. In order to compare our algorithm to FAB-MAP[2, 3], we used the bag of visual words model with the Chow-Liu tree to perform individual frame localization, utilizing the Google street view geo-tagged images as the history of observations. Then, we computed the likelihood of each of the video frames being in any of the possible geo-tagged locations. The average frame-by-frame error of the first step of FAB-MAP algorithm was 441.01 meters. The high error value in the first step prevents the algorithm from forming an appropriate trajectory in the later steps. The large error value is primarily due to the differences in our problem and the one FAB-MAP addresses, which is detecting if a robot is revisiting a previously visited location. The history of frames showing previously visited locations is assumed to be recorded using the same robot, which significantly simplifies the frame localization step compared to our problem which requires matching wild video frames to reference street view images. Note that the mean individual frame localization error of our method is 268.6 meters. Table 1 shows the results of the experiments for a set of 15 randomly selected videos of the test dataset. The error metric is defined as the mean distance (error) between the estimated frame geolocalizations and the closest ground truth frame. The results in the

	Randomly Selected Videos from Pittsburgh and Orlando															
	Avg Error.	Seq. 1	Seq. 2	Seq. 3	Seq. 4	Seq. 5	Seq. 6	Seq. 7	Seq. 8	Seq. 9	Seq. 10	Seq. 11	Seq. 12	Seq. 13	Seq. 14	Seq. 15
Frame by frame	268.6	332	207	198	102.3	161.9	197	143.2	196.1	151.9	276.3	235.0	249.0	261.4	102.1	326
Bayesian filtering	10.57	**2.10**	**2.42**	5.60	7.21	12.27	**1.06**	18.01	7.18	1.13	11.03	13.15	16.85	6.70	13.19	6.48
Bayesian + M.A.	10.17	3.20	3.07	6.03	7.13	11.80	1.08	17.31	6.86	**1.00**	11.17	12.96	**15.78**	6.67	**12.76**	6.49
Bayesian + MST	**9.94**	**2.10**	**2.42**	**5.26**	**5.15**	**11.68**	**1.06**	**10.84**	**5.93**	1.13	**10.80**	**12.26**	16.15	**3.07**	13.19	**3.79**

Table 1. Comparison of the mean error in meters for a subset of 15 videos from our test set.

table 1 are listed in meters. The first row of the table contains the results obtained using individual frame by frame geolocalization. Mean errors of individual frame by frame geolocalization of these videos range from 66 to 535 meters. These values demonstrate the low performance of frame by frame geolocalization in determining a trajectory. In contrast, the mean errors of the proposed Bayesian filter has an average mean error value of 10.57 meters. The mean errors in most of the videos are lower than 20 meters. The subsequent rows in the table compare the trajectories obtained after applying moving average (MA) smoother to the output of the Bayesian filter versus the trajectories obtained using the proposed MST based trajectory reconstruction applied to the output of the Bayesian filter. The best performances are indicated by bold characters. As can be seen, most of them correspond to the MST-based trajectory reconstruction method.

4. Conclusions

We introduced the problem of estimating a geolocalized trajectory of a camera from videos in the ”wild”. We proposed a solution to the problem based on individual geolocalization of frames, Bayesian filtering, and a MST-based curve reconstruction algorithm that produces a trajectory estimation with low average error.

References

[1] J. Civera, O. Grasa, A. Davison, and J. Montiel. 1-point ransac for ekf filtering: Application to real-time structure from motion and visual odometry. *Journal of Field Robotics*, 27:609–631, 2010. 2

[2] M. Cummins and P. Newman. Fab-map: Probabilistic localization and mapping in the space of appearance. *International Journal of Robotics Research*, June 2008. 2, 7

[3] M. Cummins and P. Newman. Appearance-only slam at large scale with fab-map 2.0. *International Journal of Robotics Research*, November 2010. 2, 7

[4] A. Davison. Real-time simultaneous localisation and mapping with a single camera. In *ICCV*, 2003. 2

[5] L. H. de Figueiredo and J. de Miranda Gomes. Computational morphology of curves. *The Visual Computer*, 11(2):105–112, 1994. 6

[6] H. Durrant-Whyte and T. Bailey. Simultaneous localization and mapping (slam): Part i the essential algorithms. *Robotics and Automation Magazine*, 13(2):99–110, 2006. 2

[7] Google. http://maps.google.com. 2

[8] Google. http://www.youtube.com. 3

[9] A. Hakeem, R. Vezzani, M. Shah, and R. Cucchiara. Estimating geospatial trajectory of a moving camera. In *ICPR*, 2006. 2

[10] A. Howard. Real-time stereo visual odometry for autonomous ground vehicles. In *IROS*, 2008. 2

[11] http://www.flickr.com. Flickr. 1

[12] G. Klein and D. Murray. Parallel tracking and mapping for smaller workspaces. In *International Symposium on Mixed and Augmented Reality*, 2007. 2

[13] I.-K. Lee. Curve reconstruction from unorganized points. *Computer Aided Geometric Design*, 17:161–177, 2000. 6

[14] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, (2), 2004. 3

[15] B. Ma, A. Hero, J. Gorman, and O. Michel. Image registration with minimum spanning tree algorithm. In *International Conference on Image Processing*, 2000. 6

[16] E. Mouragnon, M. Lhuillier, M. Dhome, F. Dekeyser, and P. Sayd. Real time localization and 3d reconstruction. In *Computer Vision and Pattern Recognition (CVPR)*, 2006. 2

[17] M. Muja and D. Lowe. Fast approximate nearest neighbors with automatic algorithm configuration. In *VISAPP'09*. 3, 7

[18] D. Nister, O. Naroditsky, and J. Bergen. Visual odometry. In *CVPR*, 2004. 1

[19] Panoramio. http://www.panoramio. 1

[20] R. Perlman. An algorithm for distributed computation of a spanningtree in an extended lan. In *SIGCOMM*, 1985. 6

[21] T. Sattler, B. Leibe, and L. Kobbelt. Fast image-based localization using direct 2d-to-3d matching. In *ICCV'11*. 2, 5

[22] D. Scaramuzza. 1-point-ransac structure from motion for vehicle-mounted cameras by exploiting non-holonomic constraints. *IJCV*, 95(1), 2010. 2

[23] G. Schindler, M. Brown, and R. Szeliski. City-scale location recognition. In *CVPR*, 2007. 2, 5

[24] J. Tardif, Y. Pavlidis, and K. Daniilidis. Monocular visual odometry in urban environments using an omnidirectional camera. In *International Conference on Intelligent Robots and Systems*, 2008. 2

[25] A. Torralba, K. P. Murphy, W. T. Freeman, and M. A. Rubin. Context-based vision system for place and object recognition. In *ICCV*, 2003. 2

[26] A. R. Zamir, A. Darino, and M. Shah. Street view challenge: Identification of commercial entities in street view imagery. In *ICMLA*, 2011. 2

[27] A. R. Zamir and M. Shah. Accurate image localization based on google maps street view. In *European Conference on Computer Vision (ECCV)*, 2010. 2, 3, 5