Video Geo-Localization Employing Geo-Temporal Feature Learning and GPS Trajectory Smoothing

Krishna Regmi and Mubarak Shah
Center for Research in Computer Vision, University of Central Florida
krishna.regmi7@gmail.com, shah@crcv.ucf.edu

Abstract

In this paper, we address the problem of video geo-localization by proposing a Geo-Temporal Feature Learning (GTFL) Network to simultaneously learn the discriminative features for the query video frames and the gallery images for estimating the geo-spatial trajectory of a query video. Based on a transformer encoder architecture, our GTFL model encodes query and gallery data separately, via two dedicated branches. The proposed GPS Loss and Clip Triplet Loss exploit the geographical and temporal proximity between the frames and the clips to jointly learn the query and the gallery features. We also propose a deep learning approach to trajectory smoothing by predicting the outliers in the estimated GPS positions and learning the offsets to smooth the trajectory. We build a large dataset from four different regions of USA; New York, San Francisco, Berkeley and Bay Area using BDD driving videos as query, and by collecting corresponding Google StreetView (GSV) Images for gallery. Extensive evaluations of proposed method on this new dataset are provided . Code and dataset details is publicly available at https://github.com/kregmi/VTE.

1. Introduction

Image based geo-localization has attracted a lot of interest in computer vision community, where a query image is matched with geo-tagged reference images in the gallery, and the GPS location of the best matching reference image is assigned to the query image. Existing works in image geo-localization solve the same-view (ground to ground) [26, 35, 44], as well as the cross-view (ground to aerial) [2, 4, 12, 14, 17, 24, 25, 27, 32, 39] image matching problems by learning robust features for the query and the gallery set. With the increase in video data, there has never been more urgency of geo-localizing the *video clips*, where GPS trajectory corresponding to a query video is determined. In this work, we explore the task of video geo-localization for the same-view data, where both query videos and gallery images are from the ground views. More specifically, given a query video recorded from a moving camera, we want to determine the GPS coordinates of each frame in the video.

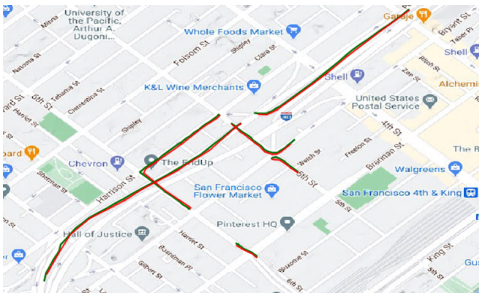


Figure 1: Sample geo-spatial trajectories for a subset of video clips from San Francisco Area of BDD dataset [42]. The ground truth trajectories are shown in green and their estimated geo-trajectories obtained by the proposed method are shown in red.

One possible way to solve this problem is to treat each frame in the video independently and apply frame-based matching methods to determine the GPS location of each frame. Earlier works in video geo-localization [9, 37] are based on classical computer vision methods, where first the SIFT descriptors [19] are computed for each frame in the clip as well as for the images in the gallery set (reference database). Then, for each frame in the query video the best matching reference image is computed and its corresponding GPS location is assigned to the query frame and the predicted GPS trajectory is obtained by connecting the frame GPS locations. Vaca-Castano *et al*. [37] use Bayesian Filtering to enforce temporal consistency on the estimated positions in order to obtain smooth trajectory. Recent works [11, 13, 45] use 2D CNN networks to obtain frame-level features instead of SIFT features for query frames and the gallery images, and follow the same approach for image matching. The features for each frame in the clip are expressed independent to each other, thus predicted GPS locations may not be smooth enough to represent a realistic trajectory of the moving camera since *no temporal closeness between the frames is exploited directly while learning*

the features for the clips.

In this paper, we propose to leverage the geo-temporal proximity between the video frames while learning their features, in order to enforce the predicted locations of the consecutive video frames to be close to each other. Motivated by the recent success of deep learning methods in video understanding and the effectiveness of transformer networks [38] to incorporate long range context dependencies between the inputs, we propose to use transformer based architecture to learn feature representations for the frames of the query videos. The network captures the coherent features for the video frames and hence provides smoother predicted trajectories. In addition to exploiting the temporal proximity between the frames within a clip, we propose to use a novel GPS loss to learn smoother features for clips that are geographically closer to each other. Typical imagery captures areas containing vegetation, landmarks and landscapes unique to those areas and can extend over a small geographical region. So, the clips over this geographical region should share similar feature representations. Thus, we propose to learn similar features for video clips corresponding to the same geographical locations by constraining the training of our proposed network by using GPS loss. Once the GPS locations for the query video are estimated, earlier works used b-spline [9] and minimum spanning tree based trajectory reconstruction algorithms [37] to smooth the initial estimates of these GPS positions. In this work, we propose a transformer encoder based trajectory smoothing network to determine the outliers in a set of estimated GPS locations for the query clip, and smooth the GPS values if they are determined to be noisy by the network.

There is no publicly available large-scale dataset for video geo-localization to evaluate the capability of the proposed framework. The dataset of Vaca-Castano *et al*. [37] consists of only 45 query videos making it impractical to train deep learning methods. Heng *et al*. [11] use dataset with image pairs that does not fit into our problem formulation. Therefore, in this work, we build a new video geo-localization benchmark dataset by utilizing Berkeley Driving Dataset (BDD) videos [42] and by collecting matching Google StreetView (GSV) images. The BDD videos are used as query videos and the GSV images form our gallery set. The dataset covers four different regions of the USA; San Francisco, Berkeley, Bay Area and New York. We provide evaluations on query videos from all four regions.

In summary, we make the following contributions in this paper: (1) We propose a novel geo-temporal feature learning approach to learn coherent features for the query video frames for the problem of video geo-localization; (2) We propose a novel GPS loss to learn geographically smoother features; (3) We propose a novel trajectory smoothing network to refine the initial GPS predictions to obtain a smooth trajectory; and (4) We build a new video geo-localization dataset and provide extensive evaluations on query videos from four different regions of the USA.

2. Related Work

2.1. Image and Video Localization

Early works on image based geo-localization [26, 35, 44] employed the hand-crafted features for matching the query and the gallery images from the same (ground) view. Researchers followed up with cross-view image geolocalization [17, 32, 14, 27, 2] and matched the features between the aerial and ground images. The hand-crafted features for geolocalization include Bag of words [32], VLAD descriptors [14], and building facades [2].

Deep neural networks gained popularity in several research areas, including image matching. A slew of works [12, 40, 4, 39, 24, 18, 29, 30, 31] followed the deep-learning trends for cross-view image matching between ground and aerial images. Authors in [12] learn NetVLAD descriptors for the images. Recent works propose triplet loss [39], in-batch reweighting triplet loss [4] to learn discriminative features. Regmi and Shah [24] utilize synthesized images to obtain robust features for the query images, whereas Liu and Li [18] leverage the orientation cues, and Shi *et al*. [29] exploit the geometric and feature correspondences between the query and gallery images.

With success of above image-based localization methods, natural extension is to explore video localization. However, a limited amount of research on video geo-localization has been reported. Authors in [37, 9] estimate trajectories of streetview videos by comparing the SIFT [19] features of query and gallery frames. Recent work by Heng *et al*. [11] explores cross-view matching for autonomous vehicle navigation in street-view. Works by Hu and Hee Lee [13] predict the trajectory of a moving ground vehicle by matching the street-view panorama to aerial images, however assume that the initial pose (position and heading) of the moving vehicle is known. Earlier works by [3, 21] perform street view navigation of agents by streetview-to-streetview matching using reinforcement learning. The scope of these works is limited to learning image based features, and these approaches do not explore joint feature learning by exploiting the temporal proximity between the frames in the query video. In this paper, we propose transformer-encoder based deep neural network to learn temporally coherent features for the query frames and use them for the task of video geo-localization over large geographical area.

2.2. Trajectory Smoothing

Earlier work by Hakeem *et al*. [9] discarded the noisy outliers from the GPS predictions and used the remaining GPS points as control points for a b-spline to interpolate the remaining locations to smooth the trajectories. Chazal *et*

al. [5] propose data-driven trajectory smoothing framework by moving the noisy GPS points to the barycenter of their nearest neighbors in feature space. Authors in [37] smooth the noisy trajectories by using a Minimum Spanning Trees (MST) based trajectory reconstruction algorithm and eliminate trajectory loops or noisy estimations. Recent work by Hu *et al*. [13] utilize visual odometry readings of the vehicle in Particle Filter algorithm [33] to smooth the initial predictions of the vehicle location.

Different from the previous works, we propose a transformer-encoder based deep neural network to determine the offsets in the noisy predictions with confidence scores for the predictions, and add the offsets to the noisy predictions to smooth the trajectories.

2.3. Attention based Networks and Applications

The inception of Transformer network [38] based on attention mechanisms for long term sequence modelling to solve the language translation task has gained a lot of popularity and has been widely used in different applications. Fu *et al*. [8] propose dual attention network with position and channel attention modules for scene segmentation task. Different applications of attention network include text to speech synthesis [16], text summarization [28], object localization [7], audio-visual event localization [41], video action localization [6, 23]. Different from previous works, we extend the attention mechanism to the task of video geo-localization to learn temporally coherent features for the query video as well as to the task of trajectory smoothing.

2.4. Geolocalization Datasets

The existing geolocalization datasets can be broadly grouped into two categories: same-view and cross-view image datasets. Earlier works by [43, 10] build street-view image datasets to conduct the same-view image matching. Some popular datasets for image-based localization with ground and satellite pairs include CVUSA[46], CVACT[18], Vo and Hays [39], and UCF-OP [24]. Tian *et al*. [34] collect cityscale streetview and bird's eye view image pairs; whereas Zheng *et al*. [47] collect image from three platforms: synthetic drones, satellites and ground cameras.

The video dataset by Majdik *et al*. [20] was collected by flying a camera-mounted micro aerial vehicle (MAV) recording the scene from 10-20 meters above the ground and capturing the frontal view of the buildings. They build a reference set of images from Google Street-View data, however, their dataset is limited to a 2 km trajectory in downtown Zurich, Switzerland. Yu *et al*. [42] collect large scale driving dataset, BDD, covering four different regions of the USA; New York, San Francisco, Berkeley and Bay Area.

In this work, we utilize the driving videos from BDD dataset as query videos. We then build a reference set of images for corresponding BDD videos by collecting Google Street View using the GPS annotations of BDD videos. Thus, we broaden the applicability of BDD dataset to advance the research on video geo-localization.

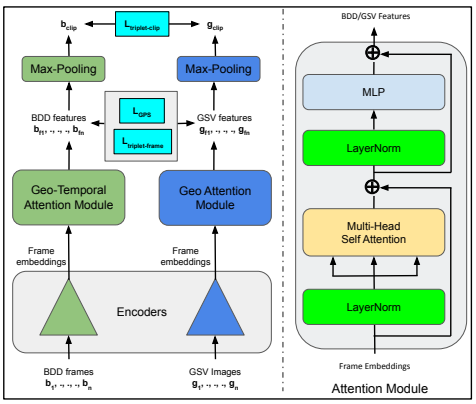


Figure 2: Geo-Temporal Feature Learning (GTFL) Network: Given a set of n frames for BDD (query) and corresponding n frames for GSV (gallery), frame embeddings are obtained using Encoders (VGG-16 network). Then, the Geo-Temporal Attention and Geo-Attention modules learn coherent feature representations for BDD and GSV frame embeddings respectively. The frame-based features are aggregated using Max-pooling operation to obtain the representative clip features, b_{clip} and g_{clip}. The Frame Triplet loss and GPS loss are applied on frame level features, and the Clip Triplet loss is applied on clip features. Detailed architecture for Attention module is shown on the right where BDD/GSV features are learnt for the frame embeddings.

3. Method

In this work, we leverage the temporal relationships between the frames in a query video to learn their features for the task of video geo-localization. We learn the representations of the query video frames and gallery images differently. The query video frames exhibit temporally smooth transition between consecutive frames, therefore the neighboring frames can be exploited to learn better features for the current frame. The gallery images, on the other hand, are collected from Google StreetView (GSV) and are more discrete and have non-uniform changes between the frames in a trajectory; thus only geographical relationships between the GSV frames is explored. We first obtain embeddings for query and gallery images using encoders, and subsequently improve the feature embeddings by employing transformer based attention networks. Finally, we smooth the estimated geo-locations of the query frames by utilizing a transformer based trajectory smoothing network.

3.1. Geo-Temporal Feature Learning Network

The proposed geo-temporal feature learning network is shown in Figure 2 and explained in detail next.

Encoders: Assume we are given a query BDD video clip of n frames, $B = [b_1, b_2, ..., b_n]$ and corresponding GSV images $G = [g_1, g_2, ..., g_n]$. The encoders, as shown in Fig-

ure 2, are used to obtain the embeddings for the input frames in our pipeline. The encoders are 2D CNN networks with VGG-16 architecture with NetVLAD [1] as final layer and shared weights[1]. We use the pre-trained weights from the network trained on Pittsburgh 250k dataset [36] and fine-tune them on our dataset. Thus obtained frame embeddings are utilized in the next stage of the pipeline.

Geo-Temporal Attention and Geo-Attention Modules: The attention modules, Geo-Temporal Attention module (upper branch) and Geo-Attention module (lower branch), have similar architectures, as shown in the right panel in Figure 2. The attention module consists of multi-head attention and feed-forward (MLP) layers similar to transformer encoder. We utilize 2 heads and 2 encoders in our attention modules.

The geo-temporal attention module exploits the temporal relationship between the frames in the video clip to learn good frame features. Each feature is learnt by attending to all the frames of the query video. On the other hand, the geo-attention module learns individual features by attending onto itself only. The modules are called 'Geo-' modules because the geo-locations of the frames are exploited in learning the features for the query video frames and the reference frames. This is done by using the GPS loss during the training, which is further explained next.

3.2. Loss Functions

We next explain the loss functions used to train the 2D CNN and the Attention blocks in our proposed architecture. We apply the triplet loss on the frame features as well as on the clip features and additional novel GPS loss to regularize the training.

Frame Triplet Loss: Consider the frame features $[b_{f1}, b_{f2}, ..., b_{fn}]$ for a query BDD video with frames $[b_1, b_2, ..., b_n]$, and the feature embeddings $[g_{f1}, g_{f2}, ..., g_{fn}]$ for corresponding matching GSV images $[g_1, g_2, ..., g_n]$. Also, consider the GSV images $[g'_1, g'_2, ..., g'_n]$ from a different location with feature representations as $[g'_{f1}, g'_{f2}, ..., g'_{fn}]$. For the BDD feature b_{fi}, the GSV feature g_{fi} is a positive feature and the GSV feature g'_{fi} is the negative feature. Now, if d_{pi} is the Euclidean distance between the positive feature pairs (b_{fi}, g_{fi}) and d_{ni} is the Euclidean distance between the negative feature pairs (b_{fi}, g'_{fi}), the objective of the frame triplet loss is to minimize d_{pi} as well as maximize d_{ni}. Thus, the frame triplet loss for query clip is computed as the sum of triplet losses for the individual frames of the clip, represented by the Equation 1.

$$L_{triplet-frame} = \sum_{i=1}^{n} max(0, m + d_{pi} - d_{ni}), \quad (1)$$

[1]Note that we performed experiments with better network like Resnet; since VGG-16 is pre-trained on Pittsburgh data set it performs better than Resnet pre-trained on ImageNet.

where, m is the margin and n is the length of the video clip.

Clip Triplet Loss: As shown in Figure 2, we apply max-pooling on the frame features of query video and the set of features for the GSV frames to obtain the representative features b_{clip} and g_{clip}. We observe that for a small window of 8 frames, the clip frames contain highly overlapping field of views and thus the clip features contain the representative features of the given location. Thus, the clip features for BDD and GSV can additionally be used into the training instead of just employing individual frame features. Therefore, along with the frame triplet loss, we propose to use clip triplet loss to optimize the training.

Assume b_{clip} and g_{clip} are BDD and GSV clip features respectively for a given geo-location, their features are considered to be positive feature pairs and their feature distance can be represented as d_{p-clip}. Similarly, if g'_{clip} is the GSV clip feature at a different geo-location, it is considered as a negative feature for b_{clip} and the feature distance between b_{clip} and g'_{clip} is represented as d_{n-clip}. The clip triplet loss is computed as shown in Equation 2.

$$L_{triplet-clip} = max(0, m + d_{p-clip} - d_{n-clip}), \quad (2)$$

where, m represents the margin.

GPS Loss: In addition to using frame triplet loss and clip triplet loss, we propose a new loss, GPS loss to further improve the training of our proposed GTFL network. The intuition behind the GPS loss is that the clips (or images) closer to each other in geographical distance are also similar in feature representations compared to the clips (or images) that are further apart in geographical distance. This is because each geographical location may have unique landmarks, landscapes and vegetation representing that region, which can spread over a small nearby area, however, this won't be valid in regions that are far away. GPS loss acts as an additional supervision to the training since most feature learning is done on image features using triplet losses.

The GPS loss is formulated as follows. Given GPS info for each frame, we compute the geodesic distance between the frames using the Algorithm for Geodesics [15]. We also compute their feature distances using the learnt feature representations. Let b_{f1} and b_{f2} refer to feature representations for two frames, and let (lat_1, lon_1) and (lat_2, lon_2) be their GPS locations respectively. The geographical distance between two GPS points, d_{gps} is computed using the algorithm in [15]. Similarly, their feature distance d_{feat} is obtained as shown in Equation 3.

$$d_{feat} = || \, b_{f1} - b_{f2} \, ||_2^2, \quad (3)$$

We then hypothesize that the normalized feature distance between the frames should be proportional to their normalized geographical distance as shown in Equation 4.

$$d_{feat} \propto d_{gps}, \quad (4)$$

To verify this, we compute the feature distances d_{feats} for the images and the physical distances d_{gps} for their GPS positions. We visualize these distances in a scatter-plot and fit a line through the points and establish a linear relationship between d_{feats} and d_{gps} with slope 1.077 and intercept of -0.2313; as reported in the Supplementary material.

We then minimize the L_1 distance between the normalized feature distance and the normalized gps distance as shown by Equation 5.

$$L_{GPS} = || \ d_{feat} - d_{gps} \ ||_1, \tag{5}$$

Any deviation in difference between the feature distance and GPS distance is penalized while training the network.

Total Loss: The overall expression for the total loss function is the sum of Equations 1, 2 and 5, as shown in Equation 6.

$$L_{total} = L_{triplet-frame} + \lambda_1 * L_{triplet-clip} + \lambda_2 * L_{GPS}, \tag{6}$$

where, λ_1 and λ_2 are the hyperparameters for the loss terms.

3.3. Trajectory Smoothing Network

The proposed GTFL network shown in Figure 2 is used to obtain the feature representations for the BDD query video frames and the GSV reference frames. The geo-location of each query frame is estimated by matching individual features to the frame features of the images in gallery set. The sequence of estimated geo-locations for the query frames represents the trajectory of the moving camera that captured the query video. The predicted trajectory may not be smooth because even though the query features are learnt jointly, the GPS positions are estimated independently by matching query frame features with the reference image features. Due to some incorrect matches or some outliers, the resultant GPS trajectory may lack temporal smoothness. We, thus, propose a trajectory smoothing method to refine the noisy GPS locations. Our approach for temporal smoothing is to determine the noisy GPS values in a set of predicted GPS locations for a query clip. A confidence score along with an offset value for each estimated GPS location is determined such that the addition of the offset to the noisy trajectory will result in a smooth trajectory.

The trajectory smoothing network consists of architecture as shown in Figure 3. It consists of linear projection (fc) layer, Transformer encoder layer followed by two parallel heads: a regression head (fc-layer) and a prediction head (fc-layer). The linear projection layer is a fully connected layer that maps a GPS location (2D) to a higher dimensional embedding; a 512 dimensional feature vector. The transformer encoder works on higher dimensional representations for the geo-locations and learns to correct GPS values in the input trajectory. The architecture of the transformer encoder layer is similar to the attention module shown in

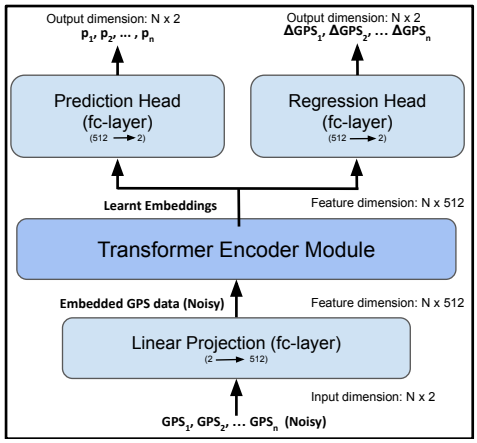


Figure 3: Proposed Trajectory Smoothing Network: The noisy GPS sequence $[GPS_1, GPS_2, ..., GPS_n]$ is input to the network to compute the error offsets $[\Delta GPS_1, \Delta GPS_2, ..., \Delta GPS_n]$ and the confidence scores $[p_1, p_2, ..., p_n]$ for each input value. The offsets are added to only those GPS values in the input sequence if they are deemed to be noisy by their confidence scores.

Figure 2, right panel. The learnt embeddings from the transformer encoder are projected back to 2-D GPS space using the regression head. The regressed values represent the normalized values of the offset in GPS error. Also, the learnt embeddings from the transformer are input to the prediction head (fc-layer) that predicts the confidence score whether a GPS location in the input sequence is noisy.

Let $[GPS_1, GPS_2, ..., GPS_n]$ be the predicted geo-locations for the query frames $[b_1, b_2, ..., b_n]$. The estimated geo-locations for majority of the query frames are close to each other, with some possible outliers that account for large errors in localization of the clip. The trajectory smoothing network determines the offset $[\Delta GPS_1, \Delta GPS_2, ..., \Delta GPS_n]$ for each input GPS values, as well as the confidence scores $[p_1, p_2, ..., p_n]$. Depending on the confidence score, the ΔGPS is added to the noisy inputs to obtain the smoothed version of the GPS values; using the Equation 7. The confidence score threshold is kept at 0.5.

$$GPS'_i = GPS_i + p_i.\Delta GPS_i. \tag{7}$$

4. Experimental Setup

This section provides details about the datasets used and the experimental setups followed in our work.

4.1. Datasets

Since there is no existing large dataset to work on video geo-localization problem, we utilize the video clips of BDD dataset [42] provided by Yu *et al*. as query clips. The BDD dataset is a large-scale driving dataset collected over four different regions of the USA, New York (NY), Berkeley, San Francisco (SF) and Bay Area. The videos are around 40 seconds in length. The dataset provides geo-location

Table 1: The GPS window of each region under consideration and the area of each region in square kilometers and the number of clips considered from each of the regions from BDD dataset. We employ video sequences only from San Francisco area for training and video sequences from all four areas for testing as shown in the last column of this table.

Regions	Latitude Range	Longitude Range	Area (square kms)	# dataset pairs train	test
San Francisco	[37.65 , 37.81]	[-122.5, -122.38]	188.06	750	95
Bay Area	[37.419279, 37.507089]	[-122.258048, -122.1054]	131.69	0	81
Berkeley	[37.72409913, 37.897474]	[-122.312608, -122.100853]	359.24	0	51
New York	[40.7073, 40.7381]	[-74.01486, -74.0072]	18.26	0	106

Figure 4: The frames from a BDD sequences at times T = 3, 6, 9, 12, 15 and 41 (Left Panel, Top); and GSV images corresponding to the same GPS locations (Left Panel, Bottom). The plot of the trajectory (green curve) along with the frame locations (red and blue dots) for different times on the aerial map (Right Panel). Numbers 3, 6, 9, 12, 15 and 41 marked with blue dots on aerial map illustrate the position of moving camera at respective times.

(GPS) annotations for the driving trajectories annotated at 1frame/second. The dataset consists of diverse scene types such as city streets, residential areas and highways. In this work, we consider the BDD video clip as a query, and estimate its corresponding GPS trajectory.

To solve trajectory estimation problem, a reference database of gallery images with known GPS is needed. The feature representations of the query is matched with the gallery image features, and the location of the gallery feature with the highest similarity to the query is selected as the estimated location of the query. Since BDD dataset doesn't provide the gallery set, we use the GPS annotations of BDD videos to download corresponding Google StreetView (GSV) images at those locations and build a gallery set. For each query location, we download four GSV images, with camera headings of 0, 90, 180 and 270 degrees. We then manually annotate the dataset to select the image that has the highest overlap with the BDD query frames.

We select different GPS windows for constructing the dataset as shown in Table 1. We collect a total of about 750 BDD-GSV pairs (query videos and gallery image) for training and 333 pairs for testing that spreads over 697.25 km^2 area. Data from San Francisco area is used for both training and testing, whereas the other three regions, Bay Area, Berkeley and New York are used only for testing.

A sample sequence of BDD and GSV frames from the dataset is shown in Figure 4. The upper row in left panel shows BDD frames at time instances T = 3, 6, 9, 12, 15 and 41 and their corresponding GSV frames are shown in the lower row. We can observe high similarity and overlap in fields of views between the BDD and GSV frames; justifying that we are successful in constructing a meaningful dataset employing BDD video frames and GSV images. The right panel shows the camera location at different time instances with (red and blue) dots on an aerial map; and the green curve connecting them demonstrates the path that the camera takes.

4.2. Implementation Details

In this section, we present the implementation details of our transformer based architecture for geo-temporal feature learning and trajectory smoothing networks. We use PyTorch [22] for the implementations.

Geo-Temporal Feature Learning Network: The GTFL network consists of encoders followed by attention modules as shown in Figure 2. The encoders are VGG-16 networks, with NetVLAD layer as the final layer, and shared weights for both branches. We use the pretrained weights of the network trained on Pittsburgh 250k dataset to initialize the parameters of the network and fine-tune them on our dataset. The output of frame-encoder network is a 32,768-dimensional feature representation for each input frame. Geo-temporal attention and geo-attention modules consist of encoder modules of Transformer network. We use 2 attention heads and 2 encoder layers for both modules. The weights are randomly initialized.

We use triplet losses on the frame features, and the clip features as well as the proposed GPS loss on the frame features. λ_1 and λ_2 are the balancing factors between the losses and their values are set at 10 and 10 respectively. The query and gallery features are represented by 512-dim. vectors.

Trajectory Smoothing Network: The trajectory smoothing network consists of a fully-connected layer that maps 2-dimensional GPS values to 512 dimensional representations, followed by the encoder module of transformer. The transformer encoder consists of four attention heads and two encoder layers. The output of the transformer encoder is passed through two parallel heads, fully connected layers that map the 512-dimensional representations to 2-dimensional values of offset regression and confidence score prediction. During the training, we employ data augmentation by feeding noisy GPS values and artificially perturbed GPS values as input to the network. We observe that by artificially perturbing some ground truth GPS values and using them as input provides the network with strong guidance that not all GPS location are noisy, and only some need modifications, whereas the rest should be kept unchanged.

5. Results

We present extensive evaluation of our proposed method demonstrating the effectiveness of collectively learning the clip features to estimate a smoother trajectories for the query videos.

5.1. Evaluation Metric

We provide evaluation in terms of localization error as well as recall accuracy. For localization error, we first compute the distance in meters between the estimated GPS positions of the query video frames and their ground truth GPS locations. Average of the error distances for the frames provides the localization error for the query clip.

Recall accuracy is reported in terms of recall at top-K and recall at distance threshold. For recall accuracy at top-K, a matching is successful if the correct match is within a set of K closest images, in Euclidean distance of their features. For recall accuracy at distance threshold, a query is correctly localized if its distance in meters to its ground truth position is within the threshold distance.

5.2. Quantitative Evaluation

We present the quantitative evaluation of the baselines and our proposed method in terms of localization error and recall accuracy.

Localization Error: We compare our proposed approach with the baseline 2D CNN and 3D CNN architectures as well as with an IBL method by Zemene *et al*. [45]. The baseline networks are explained next.

The 2D CNN baseline consists of VGG-16 architecture with NetVLAD [1] as the final layer and uses the pretrained weights, same as the Encoders explained in section 3.1. We conduct the evaluation employing the raw features obtained using the pretrained weights and report the results in first row of Table 2. We next finetune the 2D CNN baseline on our dataset. The features for each frame in the query video clip are learnt independently and their GPS locations are predicted using the query features. The predicted locations are smoothed using the proposed smoothing network. The results are presented in the second row of Table 2. Next, we use a recent image based localization (IBL) method by Zemene *et al*. [45] to conduct evaluation on our dataset. We also perform trajectory smoothing by Interpolation as a baseline. First, a GPS is determined an outlier if its distance to all other GPS positions in the trajectory is larger than a threshold. Then, a new GPS value is assigned to it by interpolating the GPS values of the previous and next frame. The results are presented in the third row of Table 2.

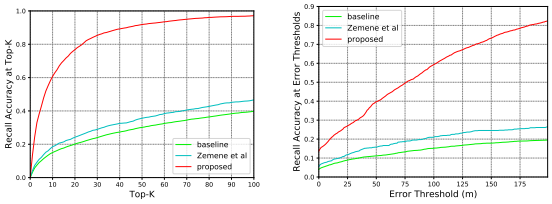
Table 2: Comparison of proposed approach with baseline methods in terms of localization error (meters). 2D CNN : Evaluation using raw features from pretrained VGG network. 2D CNN^f : VGG network fine-tuned on our dataset. Smoothing* : Smoothing by Interpolation.

Methods	SF	Bay Area	Berkeley	NY
2D CNN	2516	4686	7020	1818
2D CNN + Smoothing	2290	3999.17	5425	1292
2D CNN^f	2091.66	4509.15	6687.61	1332.08
2D CNN^f + Smoothing	1710.54	4112.15	4565.46	1222.88
[45]	1742.39	4257.08	5031.35	1164.83
[45] + Smoothing*	1327.77	3253.06	3755.98	935.78
3D CNN	4247.09	6183.71	6677.93	1572
3D CNN + Smoothing	3848.83	5201.65	6503.96	1399.17
Proposed	300.47	524.28	424.79	493.43
Proposed + Smoothing	**128.94**	**206.32**	**161.51**	**285.41**

We also conduct the baseline experiment using 3D CNN architecture to learn the feature embeddings for the query frames. Here, the ResNet R3D-18 is used with the modification that the temporal dimension is preserved; meaning the output for N input frames in a clip will have N features. But these features are learnt by considering the neighboring frames as well, since the kernel size of 3 for temporal dimension is considered. The results are presented in the fourth row of Table 2. Since the network is trained from scratch, it performs worse compared to 2D CNN baseline with fine-tuning.

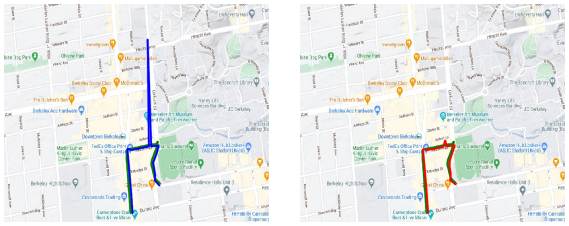
Finally, we present the results for the proposed method in the fifth row of Table 2. For our proposed method where the network is able to consider all the frames in the input to generate their individual features, the results are significantly better than the baseline networks. Also, smoothing of the trajectory helps to reduce the localization error further.

Recall Accuracy: We next report the comparison of our proposed method with the baseline method (2D-CNN) and SOTA-IBL method (Zemene *et al*. [45]) in terms of recall accuracies. We present the top-K recall accuracy for K = 1 to 100 in Figure 5a. Here, we visualize the recall accuracy plot for all four regions. We observe that our proposed method performs significantly better than the base-



(a) (b)

Figure 5: Comparison of proposed approach with baseline 2D CNN^f and Zemene *et al*. [45] in terms of recall accuracies.



(a) (b)

Figure 6: Trajectory smoothing example for a query clip from Berkeley region. (a) shows the ground truth (green curves) and the predicted trajectories before smoothing (blue curves). (b) show the ground truth (green curves) and the predicted smooth trajectories (red curves).

line methods. We also report recall accuracy with respect to distance threshold in Figure 5b. We observe that the proposed method is better than the baselines for all values of distance thresholds. The recall accuracy plots illustrate the superiority of our proposed approach to video geo-location compared to the frame based baseline networks.

5.3. Qualitative Evaluation

Figure 1 shows the geo-spatial trajectories predicted by our proposed method on subset of videos from San Francisco Area and their comparison with the ground truth trajectories. The green curves represent the ground truth trajectories for the camera while the red curves are the trajectories predicted by our proposed method. These qualitative results demonstrate the capability of the proposed method in large-scale video localization.

We also present a trajectory smoothing example in Figure 6. Figure 6a presents a noisy trajectory (blue curve) obtained by using the proposed GTFL network. The trajectory smoothing network refines the noisy trajectory resulting in the smooth trajectory as shown in Figure 6b. We observe significant impact of trajectory smoothing network in obtaining smoother predicted trajectory. Additional qualitative results are provided in the Supplementary material.

5.4. Ablation Study

We conduct ablation studies to understand the impact of different loss functions used in our experiments. We also conduct ablations on the parameters of trajectory smoothing network to determine the best hyperparameters. We report ablations on feature dimensions of GTFL network and the contribution of NetVLAD layer in Supplementary material.

Table 3: Ablation study of GPS errors in meters with respect to different losses during the training. TL$_f$: Frame Triplet loss; TL$_c$: Clip Triplet loss; GL : GPS Loss.

Losses	SF	Bay Area	Berkeley	NY
TL$_f$	704.16	1172.99	1253.6	978.51
TL$_f$ + TL$_c$	524.81	692.34	819.54	836.13
TL$_f$ + TL$_c$ + GL	**300.47**	**524.28**	**424.79**	**493.43**

Table 4: Ablation on the number of heads in self-attention module and the number of transformer encoder layers.

Parameters	SF	Bay Area	Berkeley	NY
Heads = 2, Layers = 1	158.07	233.86	189.54	307.91
Heads = 2, Layers = 2	144.21	211.59	168.61	292.24
Heads = 4, Layers = 2	**128.94**	**206.32**	**161.51**	**285.41**
Heads = 4, Layers = 4	129.46	227.78	183.97	312.57

Ablation on Losses: For this ablation, we conduct the experiments with different combinations of loss functions for our proposed method. The results after the application of the trajectory smoothing on the predicted GPS are shown. The results are presented in Table 3. The numbers suggest that utilizing the clip triplet loss helps in obtaining better GPS estimation compared to with only frame triplet loss; and the use of GPS loss further helps the network to learn discriminative features for the clips and the localization error decreases further.

Ablation on parameters for Trajectory Smoothing Network: For this ablation, we conduct experiments for smoothing the predicted GPS trajectory by varying the number of layers of transformer encoder network and varying the number of heads in the self-attention layer. The result is shown in Table 4. We observe that the best results are obtained for heads = 4 and layers = 2.

6. Conclusion

In this paper, we have presented a novel application of transformer based networks for long-term feature learning between the frames of a query video clip for the task of video geo-localization as well as for geo-trajectory smoothing. We formulated novel GPS loss and validated its contribution in learning better features for the query and gallery frames. We built a new benchmark dataset for video geo-localization and report significant improvement of proposed method over frame based feature learning approach where the temporal relations between the frames are not captured and over 3D-CNN baseline where only short term temporal information is incorporated.

Acknowledgements: The authors would like to thank Amir Roshan Zamir and Gonzalo Vaca-Castano for their help with their dataset; and Yonatan Tariku Tesfaye and Eyasu Zemene for their help with their code.

References

[1] Relja Arandjelovic, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. Netvlad: Cnn architecture for weakly supervised place recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5297–5307, 2016. 4, 7

[2] Mayank Bansal, Kostas Daniilidis, and Harpreet Sawhney. Ultra-wide baseline facade matching for geo-localization. In Andrea Fusiello, Vittorio Murino, and Rita Cucchiara, editors, *Computer Vision – ECCV 2012. Workshops and Demonstrations*, pages 175–186, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg. 1, 2

[3] Samarth Brahmbhatt and James Hays. Deepnav: Learning to navigate large cities. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5193–5202, 2017. 2

[4] Sudong Cai, Yulan Guo, Salman Khan, Jiwei Hu, and Gongjian Wen. Ground-to-aerial image geo-localization with a hard exemplar reweighting triplet loss. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019. 1, 2

[5] Frédéric Chazal, Daniel Chen, Leonidas J. Guibas, Xiaoye Jiang, and Christian Sommer. Data-driven trajectory smoothing. In *19th ACM SIGSPATIAL International Symposium on Advances in Geographic Information Systems (GIS)*, pages 251–260, 2011. 3

[6] Peihao Chen, Chuang Gan, Guangyao Shen, Wenbing Huang, Runhao Zeng, and Mingkui Tan. Relation attention for temporal action localization. *IEEE Transactions on Multimedia*, 22(10):2723–2733, 2019. 3

[7] Junsuk Choe and Hyunjung Shim. Attention-based dropout layer for weakly supervised object localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2219–2228, 2019. 3

[8] Jun Fu, Jing Liu, Haijie Tian, Yong Li, Yongjun Bao, Zhiwei Fang, and Hanqing Lu. Dual attention network for scene segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 3

[9] Asaad Hakeem, Roberto Vezzani, Mubarak Shah, and Rita Cucchiara. Estimating geospatial trajectory of a moving camera. In *18th International Conference on Pattern Recognition (ICPR'06)*, volume 2, pages 82–87. IEEE, 2006. 1, 2

[10] James Hays and Alexei A. Efros. im2gps: estimating geographic information from a single image. In *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2008. 3

[11] Lionel Heng, Benjamin Choi, Zhaopeng Cui, Marcel Geppert, Sixing Hu, Benson Kuan, Peidong Liu, Rang Nguyen, Ye Chuan Yeo, Andreas Geiger, et al. Project autovision: Localization and 3d scene perception for an autonomous vehicle with a multi-camera system. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 4695–4702. IEEE, 2019. 1, 2

[12] Sixing Hu, Mengdan Feng, Rang M. H. Nguyen, and Gim Hee Lee. Cvm-net: Cross-view matching network for image-based ground-to-aerial geo-localization. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 1, 2

[13] Sixing Hu and Gim Hee Lee. Image-based geo-localization using satellite imagery. *International Journal of Computer Vision*, pages 1–15, 2019. 1, 2, 3

[14] Hervé Jégou, Matthijs Douze, Cordelia Schmid, and Patrick Pérez. Aggregating local descriptors into a compact image representation. In *CVPR 2010-23rd IEEE Conference on Computer Vision & Pattern Recognition*, pages 3304–3311. IEEE Computer Society, 2010. 1, 2

[15] Charles FF Karney. Algorithms for geodesics. *Journal of Geodesy*, 87(1):43–55, 2013. 4

[16] Naihan Li, Shujie Liu, Yanqing Liu, Sheng Zhao, and Ming Liu. Neural speech synthesis with transformer network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6706–6713, 2019. 3

[17] Min Lin, Qiang Chen, and Shuicheng Yan. Network in network. *arXiv preprint arXiv:1312.4400*, 2013. 1, 2

[18] Liu Liu and Hongdong Li. Lending orientation to neural networks for cross-view geo-localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5624–5633, 2019. 2, 3

[19] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004. 1, 2

[20] András L Majdik, Yves Albers-Schoenberg, and Davide Scaramuzza. Mav urban localization from google street view data. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3979–3986. IEEE, 2013. 3

[21] Piotr Mirowski, Matt Grimes, Mateusz Malinowski, Karl Moritz Hermann, Keith Anderson, Denis Teplyashin, Karen Simonyan, Andrew Zisserman, Raia Hadsell, et al. Learning to navigate in cities without a map. In *Advances in Neural Information Processing Systems*, pages 2419–2430, 2018. 2

[22] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. 6

[23] Rizard Renanda Adhi Pramono, Yie-Tarng Chen, and Wen-Hsien Fang. Hierarchical self-attention network for action localization in videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019. 3

[24] Krishna Regmi and Mubarak Shah. Bridging the domain gap for ground-to-aerial image matching. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019. 1, 2, 3

[25] Royston Rodrigues and Masahiro Tani. Are these from the same place? seeing the unseen in cross-view image geo-localization. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 3753–3761, January 2021. 1

[26] Torsten Sattler, Michal Havlena, Konrad Schindler, and Marc Pollefeys. Large-scale location recognition and the geometric burstiness problem. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 1, 2

[27] Qi Shan, Changchang Wu, Brian Curless, Yasutaka Furukawa, Carlos Hernandez, and Steven M Seitz. Accurate geo-registration by ground-to-aerial image matching. In *2014 2nd International Conference on 3D Vision*, volume 1, pages 525–532. IEEE, 2014. 1, 2

[28] Tian Shi, Yaser Keneshloo, Naren Ramakrishnan, and Chandan K Reddy. Neural abstractive text summarization with sequence-to-sequence models. *ACM Transactions on Data Science*, 2(1):1–37, 2021. 3

[29] Yujiao Shi, Liu Liu, Xin Yu, and Hongdong Li. Spatial-aware feature aggregation for image based cross-view geo-localization. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 10090–10100. Curran Associates, Inc., 2019. 2

[30] Yujiao Shi, Xin Yu, Dylan Campbell, and Hongdong Li. Where am i looking at? joint location and orientation estimation by cross-view matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4064–4072, 2020. 2

[31] Yujiao Shi, Xin Yu, Liu Liu, Tong Zhang, and Hongdong Li. Optimal feature transport for cross-view image geo-localization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 11990–11997, 2020. 2

[32] Josef Sivic and Andrew Zisserman. Video google: Efficient visual search of videos. In *Toward category-level object recognition*, pages 127–144. Springer, 2006. 1, 2

[33] Sebastian Thrun. Particle filters in robotics. In *Proceedings of the Eighteenth Conference on Uncertainty in Artificial Intelligence*, UAI'02, page 511–518, San Francisco, CA, USA, 2002. Morgan Kaufmann Publishers Inc. 3

[34] Yicong Tian, Chen Chen, and Mubarak Shah. Cross-view image matching for geo-localization in urban environments. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3608–3616, 2017. 3

[35] Akihiko Torii, Josef Sivic, and Tomas Pajdla. Visual localization by linear combination of image descriptors. In *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, pages 102–109. IEEE, 2011. 1, 2

[36] A. Torii, J. Sivic, T. Pajdla, and M. Okutomi. Visual place recognition with repetitive structures. In *CVPR*, 2013. 4

[37] Gonzalo Vaca-Castano, Amir Roshan Zamir, and Mubarak Shah. City scale geo-spatial trajectory estimation of a moving camera. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1186–1193. IEEE, 2012. 1, 2, 3

[38] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *arXiv preprint arXiv:1706.03762*, 2017. 2, 3

[39] Nam Vo, Nathan Jacobs, and James Hays. Revisiting im2gps in the deep learning era. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017. 1, 2, 3

[40] Scott Workman, Richard Souvenir, and Nathan Jacobs. Wide-area image geolocalization with aerial reference imagery. In *IEEE International Conference on Computer Vision (ICCV)*, 2015. 2

[41] Yu Wu, Linchao Zhu, Yan Yan, and Yi Yang. Dual attention matching for audio-visual event localization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6292–6300, 2019. 3

[42] Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. Bdd100k: A diverse driving dataset for heterogeneous multitask learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2636–2645, 2020. 1, 2, 3, 5

[43] Amir Roshan Zamir and Mubarak Shah. Accurate image localization based on google maps street view. In *European Conference on Computer Vision*, pages 255–268. Springer, 2010. 3

[44] Amir Roshan Zamir and Mubarak Shah. Image geo-localization based on multiple nearest neighbor feature matching using generalized graphs. *IEEE Trans. Pattern Anal. Mach. Intell.*, 36(8):1546–1558, 2014. 1, 2

[45] Eyasu Zemene, Yonatan Tariku Tesfaye, Haroon Idrees, Andrea Prati, Marcello Pelillo, and Mubarak Shah. Large-scale image geo-localization using dominant sets. *IEEE transactions on pattern analysis and machine intelligence*, 41(1):148–161, 2019. 1, 7, 8

[46] Menghua Zhai, Zachary Bessinger, Scott Workman, and Nathan Jacobs. Predicting ground-level scene layout from aerial imagery. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 3

[47] Zhedong Zheng, Yunchao Wei, and Yi Yang. University-1652: A multi-view multi-source benchmark for drone-based geo-localization. In *Proceedings of the 28th ACM international conference on Multimedia*, pages 1395–1403, 2020. 3