Joint Saliency Estimation and Matching using Image Regions for Geo-Localization of Online Video

Haoyue Shi
School of EECS, Peking University
No.5 Yiheyuan Rd.
Beijing, P. R. China 100871
hyshi@pku.edu.cn

Jia Chen
Language Technologies Institute,
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA, USA 15213
jiac@cs.cmu.edu

Alexander G. Hauptmann
Language Technologies Institute,
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA, USA 15213
alex@cs.cmu.edu

Abstract

In this paper, we study automatic geo-localization of online event videos, which is a key component behind various event analysis tasks such as cross-camera tracking and 3D reconstruction from co-located videos generated by multiple users. Different from general image localization task through matching, the appearance of an environment during significant events varies greatly from its daily appearance, since there are usually crowds, decorations, obstructions or even destruction when a major event happens. This introduces a major challenge: matching the event environment to the daily environment, e.g. as recorded by Google Street View. We observe that some regions in the image, as part of the environment, still preserve the daily appearance even though the whole image (environment) looks quite different. Based on this observation, we formulate the problem as joint saliency estimation and matching at the image region level, as opposed to the key point or whole-image level. As image-level labels of daily environment are easily generated with GPS information, we treat region based saliency estimation and matching as a weakly labeled learning problem over the training data. Our solution is to iteratively optimize saliency and the region-matching model. For saliency optimization, we derive a closed form solution, which has an intuitive explanation. For region matching model optimization, we use self-paced learning to learn from the pseudo labels generated by (sub-optimal) saliency values. In the test stage, we retrieve daily environment images for each frame in the video based on an image level similarity score calculated by region saliency and matching. We combine the GPS information of the retrieved and matched images to get the final geo-location for the video. We conduct extensive experiments on two challenging public datasets: Boston Marathon 2013 and Tokyo Time Machine. Experimental results show that our solution significantly improves over matching on whole images and the automatically learned saliency is a strong predictor of distinctive building areas.

Keywords

Video Geo-Localization, Region Saliency, Region Matching

ICMR '17, , June 6–9, 2017, Bucharest, Romania

ACM ISBN ISBN 978-1-4503-4701-3/17/06…$15.00.
https://doi.org/http://dx.doi.org/10.1145/XXXXXXX.XXXXXXX

ACM Reference format:
Haoyue Shi, Jia Chen, and Alexander G. Hauptmann. 2017. Joint Saliency Estimation and Matching using Image Regions for Geo-Localization of Online Video. In *Proceedings of ICMR '17, June 6–9, 2017, Bucharest, Romania,* , 9 pages.
https://doi.org/http://dx.doi.org/10.1145/XXXXXXX.XXXXXXX

1 Introduction

The popular smartphones with high quality cameras have enabled people to capture events all over the world by videos and share them via social media conveniently. When an event happens, different cameras may record it at different positions from different perspectives of view. For example, videos from surveillance cameras usually record the event from a fixed location from an aerial (usually 45-degree from top to down) view; videos from television reporters usually cover the event in a rather professional perspective (helicopter view or first person view) at major locations; the videos from passers-by usually capture event from a personal perspective at side locations that are often not covered by the news reporters. On the one hand, finding the location of the video content is the basic and essential element behind various event analysis tasks such as cross camera person tracking and scene reconstruction. On the other hand, unlike EXIF meta-data in images, most of the videos shared online do not come with GPS information together.

In this paper, we study the problem of automatic event video localization. The input of event video localization is the query, event video, and the database, environment images with GPS information, which could be street view images if the event video is taken on the ground or satellite images if the event video involves a 45-degree view. The task is to geo-localize the event video by matching its content to the database. To be specific, event video and environment images belong to two different domains and this is actually a cross-domain matching problem.

Abundant research works have been conducted in a related problem: general image localization through matching[1, 2, 7, 9] on public image datasets such as Paris dataset[19] and Oxford building[18]. Most of the query images in these datasets are photos of the landmarks in regular days. However, the appearance of the location could change a lot when an event happens. Here we give an example of location appearance change: Boston Marathon 2013 finish line explosion event. As shown in Figure 1, the appearance of the finish line (near the Boston Public Library) was quite different from the regular day appearance: additional audience stands were added to the sidewalk and blue end banner was raised. When an event happens, e.g. parade, protest or attack, the appearance of the district

is usually changed to some extent by either manual decoration or manual destruction. Thus, such dramatic appearance change is very common and unneglectable in event video localization task. To the best of our knowledge, the only public dataset that is built to take into account large appearance change is the Tokyo Time Machine (Tokyo TM) dataset[1]. However, it only contains daily street view images while the variety in event videos is much larger, including both 45 degree view and street view on event days.

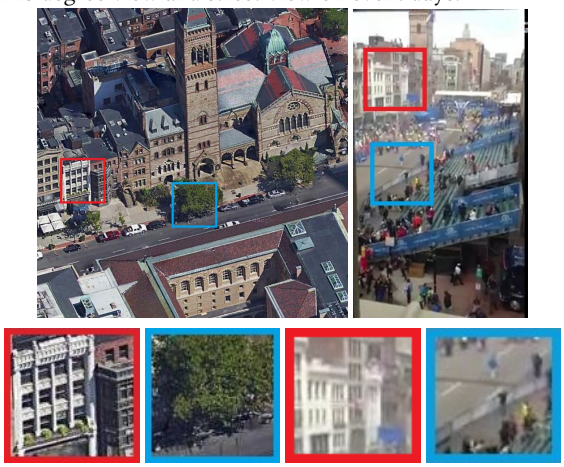

Figure 1: Comparison between the satellite image (regular appearance) and the video appearance of the place near Boston Public Library (the finish line of Boston Marathon 2013). The regions in red frames are salient, while those in blue frames are non-salient.

The major challenge of event video localization lies in cross-domain large appearance change. We observe that despite the large appearance change of the whole environment during the event, certain parts of the environment preserves their daily environment appearance. That is, though the whole appearance of each frame is quite different from daily environment image, some regions in a frame are similar to the regions in the daily environment images. Based on this key observation, we formulate the task as a joint saliency estimation and matching problem, which simultaneously outputs the matching score and the matched region as supporting evidence. Such problem formulation benefits us in three aspects.

First, by explicitly introducing saliency estimation in the model, we are able to distinguish between meaningful matching and meaningless matching. For example, the matching between building areas across two images is meaningful since building areas are usually discernible for localization. The matching between road and tree areas across two images is meaningless since such matching could occur everywhere and is not helpful to localization. Saliency estimation helps to distinguish between these two types of matching and its result will be used to down-weight the meaningless matching and emphasize on meaningful matching in our joint model.

Second, joint modeling preserves the interdependency between saliency estimation and matching. Region saliency is actually domain dependent and depends to matching. For example, if a video frame region matches many irrelevant environment images district A , it is not reliable to localize the video in district A based on this frame region. That is, this video frame region is not salient in district A. Now consider the same video frame region but the environment images of a different district B. The same video frame region may only match a few relevant environment images in district B, which means that it is reliable to localize the video based on this frame region in district B. That is, the same video frame region turns to be salient in district B. In our solution, we capture such interdependency between saliency estimation and matching.

Third, addition output of matched region makes the system output more explainable. For example, when the system fails, the "matched" region will be shown to users that there is not enough discernible environment clues in the video content so that it fails reasonably. Given such supporting evidence, the users will be more likely to trust the system as they learn to know the capability limit of the system in their usage.

Our solution is composed of three components:
Automatic weak label generation: we first generate the labels, which indicate whether two images are matched, by their GPS coordinate. If their distance in the GPS coordinate system is less than a threshold, we label them as matched, otherwise unmatched.
Saliency estimation with region matching fixed: we derive a closed form solution for saliency value, which has an intuitive explanation.
Self-paced learning for region matching model: we leverage self-paced learning to learn region matching model from pseudo labels generated by saliency estimation.

The contributions of this paper are threefold.
1. We formulate the event video localization task as a region based joint saliency estimation and matching problem.
2. We derive an iterative solution to the non-convex optimization problem, which is composed of two steps: closed form solution for saliency estimation and self-paced learning for region matching model.
3. Our solution output is not only accurate but also interpretable by human beings.

The rest of the paper is organized as follows. Section 2 introduces related work. Section 3 formalizes the problem for the video event localization task. Section 4 gives the solution to the problem. Section 5 presents the experiment results. Section 6 draws some conclusions and the work in the future.

2 Related Work

As we have mentioned in Section 1, early work by Hays and Efros [9] revealed the feasibility of the image localization task. Zamir and Roshan [23] used SIFT descriptors for image localization by voting, and constructed a dataset based on Google Street-View to test the efficiency of the algorithm. Lin et al. [15] proposed the first ground-to-overhead geo-localization method, of which the key idea is to learn the relationship between the ground-level images and their over-head appearances. After that, Lin et al. [16] then published an approach by convolutional neural network(CNN) to learn deep representations for ground-to-overhead geo-localization. Vo and Hays [22] then explored several deep CNN architectures for the cross-domain matching and improved the accuracy of image geo-localization. What's more, Bansal et al. [3] proposed a method to capture the structure of self-similarity of patterns on facades for image geo-localization.

Leung et al. [14] designed a monocular vision based particle filter localization system for urban settings that uses aerial reference map.

However, because of the great similarity between the task of image localization and video localization, researchers often treat them as the same task.

Visual place recognition is a similar task of image geo-localization. Arandjelovic et al. [1] developed NetVLAD, a CNN architecture of which the main component is the VLAD ("Vectors of Locally Aggregated Descriptors") layer and got the state-of-the-art performance on two challenging place recognition datasets.

While attention-based models are widely used in recognition tasks, many recent works show that attention-based model can improve the performance of machine learning models [17, 24]. The key idea of attention-based model is to add attention information to build a representation. This idea is intuitive. For example, when looking at an image, we human beings often recognize the objects on it and then receive the information from the background, rather than receiving information simultaneously from the foreground objects and background scenes. In our approach, we compute the attention to each region by its saliency.

In machine learning, a sequence of gradually added training samples [4] is called a curriculum. Inspired by the cognitive process of humans and animals, a straightforward way to generate such a sequence is to add samples based on their "easiness" to learn. However, such "easiness" is based on specific problem and hard to generalize. In order to solve this problem, Self-Paced Learning (SPL) was introduced by Kumar et al. [13], which embeds curriculum designing into model learning. In SPL, the curriculum is gradually generated by the model itself based on what it has learned, and it's also a general implementation for curriculum learning. Following that, several works have improved self-paced learning [11, 12, 21]. In this paper, we propose self-paced learning method to automatically estimate the saliency score for each region of a frame.

3 Problem Formulation

The event video localization task is to retrieve environment images from database given the query event video. The database images are automatically downloaded from online map service given a circle or rectangle shaped district where the event happened. If the video is taken on the ground, the database images will be Google street view images; if the video is taken from a 45 degree up to down perspective of view, the database images will be Google satellite images. The details of database image generation will be elaborated in section 4.

A query video is composed of frames $\{q\}$. As we process one frame per time in matching the database, we abbreviate one frame from the query video as query for the ease of representation. The environment image from the database is denoted as d and the database is denoted as $D = \{d\}$. We divide images to different regions on multi-scales and each region is assigned a saliency score s to indicate how import the region is in matching. We use subscript to represent region index. That is, we have saliency score $s(q_i)$ for region i in query q and $s(d_j)$ for region j in the database image d. We define the non-negative matching score between two regions as $m(q_i, d_j; \mathbf{\Theta})$, where $\mathbf{\Theta}$ is the parameters to be learn. The similarity of two regions is proportional to their matching score. For the ease of notation, we will use $m(q_i, d_j; \mathbf{\Theta})$ and $m(q_i, d_j)$ interchangeably in the following.

Now we have defined regions in the image, saliency score on regions and matching score between regions. Putting all these together, we define the matching score between query and database image as weighted sum of region matching scores:

$$M(q, d) = \sum_{i,j} w_{ij} m(q_i, d_j) \tag{1}$$

$$w_{ij} = s(q_i) + s(d_j) \tag{2}$$

where the weight w_{ij} is defined as the sum of saliency score of both regions. On the one hand, if both regions are salient and their matching score is high, such region pair contribute much to the final matching score $M(q, d)$. On the other hand, even if the matching score between two regions is high, but if both regions are not salient (e.g. road or tree), such region pair doesn't contribute much to the final matching score $M(q, d)$.

In the training stage, we automatically generate the labels of image pairs from database D by leveraging the GPS information assigned to each environment image. The label $y_{(q,d)}$ of image q and d is 1 if two images are matched and 0 vice versa. The training data is denoted as:

$$\begin{aligned} D_{train} = \{(q, d, y_{(q,d)})\} \\ q \in D \\ d \in D \end{aligned} \tag{3}$$

The details of automatically generating D_{train} from D are given in section 4. Note that the labels are given on image pairs rather than region pairs. As the matching model $m(q_i, d_j; \mathbf{\Theta})$ is defined on region pair, the labels are actually weak labels. We use contrastive noise loss function as $L(q, d, y)$:

$$\begin{aligned} L(q, d, y) &= y\, L_P(q, d, y) + (1 - y)\, L_N(q, d, y) \\ L_P(q, d, y) &= - \sum_{y_{(q,d)}=1} M(q, d) \\ L_N(q, d, y) &= - \sum_{y_{(q,d)}=0} \max(0, \Delta - M(q, d)) \end{aligned} \tag{4}$$

where Δ is a hyper-parameter to keep unmatched pairs having small matching scores, L_P is the loss function for matched pairs, L_N is the loss function for unmatched pairs. We use eq (1) to expand image matching $M(q, d)$ term in the loss function:

$$\begin{aligned} L(q, d, y) &= - y \sum_{y_{(q,d)}=1} M(q, d) - (1 - y) \sum_{y_{(q,d)}=0} \max(0, \Delta - M(q, d)) \\ &= - y \sum_{i,j} \sum_{y_{(q,d)}=1} w_{ij}\, m(q_i, d_j) \\ &\quad - (1 - y) \sum_{i,j} \sum_{y_{(q,d)}=0} w_{ij}\, \max(0, \Delta^{'} - m(q_i, d_j)) \\ &= y \underbrace{\sum_{i,j} w_{ij}\, L_P(q_i, d_j)}_{L_P(q,d)} + (1 - y) \underbrace{\sum_{i,j} w_{ij}\, L_N(q_i, d_j)}_{L_N(q,d)} \end{aligned} \tag{5}$$

In the last step substitution, we denote

$$L_P(q_i, d_j, y) = -\sum_{i,j} \sum_{y_{(q,d)}=1} m(q_i, d_j)$$
$$L_N(q_i, d_j, y) = -\sum_{i,j} \sum_{y_{(q,d)}=0} \max(0, \Delta^{'} - m(q_i, d_j)) \quad (6)$$

The last step in eq (5) shows that the two components in image level loss, $L_P(q, d, y)$ and $L_N(q, d, y)$, are weighted sum of the two components in region level loss, $L_P(q_i, d_j, y)$ and $L_N(q_i, d_j, y)$, respectively. Furthermore, if we consider the region level label to be the same as its corresponding image level label, i.e. $y_{(q_i,d_j)} = y_{(q,d)}$, we could see that region level loss $L_P(q_i, d_j, y)$, $L_N(q_i, d_j, y)$ in eq (6) has exactly the same form of image level loss $L_P(q, d, y)$, $L_N(q, d, y)$ in eq (4). However, recall that the major challenge in event video localization, large appearance change, leads to the fact that $y_{(q,d)} = 1$ doesn't imply that $y_{(q_i,d_j)} = 1, \forall q_i \in q, \forall d_j \in d$. In our problem formulation, we solve this issue by multiplying a region pair dependent weight w_{ij} to each region level loss $L_P(q_i, d_j)$. Note that w_{ij} is defined by saliency $s(q_i)$ and $s(d_j)$. That is, we could suppress the influence of "wrong" region level label to the final loss function through saliency.

We recap all the above deduction into one main problem.

main problem:

$$\begin{aligned} min_{\mathbf{S},\boldsymbol{\Theta}} L(q, d, y; \mathbf{S}, \boldsymbol{\Theta}) = & -y \sum_{i,j} \sum_{y_{(q,d)}=1} w_{ij}\, m(q_i, d_j; \boldsymbol{\Theta}) \\ & - (1-y) \sum_{i,j} \sum_{y_{(q,d)}=0} w_{ij}\, \max(\Delta^{'} - m(q_i, d_j; \boldsymbol{\Theta})) \\ w_{ij} = & \; s(q_i) + s(d_j) \\ \mathbf{S} = & \; \{s(q_i)\} \cup \{s(d_j)\} \\ s.t. ||s(q)||_2 = & \; 1 \quad \forall q \\ ||s(d)||_2 = & \; 1 \quad \forall d \\ s(q_i), s(d_j) \in & \; [0, 1] \\ y \in & \; \{+1, 0\} \end{aligned}$$

The notations introduced in this section are summarized in table 1.

q_i	region i in the query image q
d_j	region j in the database image d
$s(q_i), s(d_j)$	the saliency of the region q_i, d_j
$m(q_i, d_j)$	non-negative matching score between region q_i and d_j
Q	all video frames $\{q\}$
D	all database images $\{d\}$

Table 1: Notations introduced in this section.

4 Solution

4.1 Training

The main problem is non-convex and very difficult to optimize. One typical solution is to use cyclic coordinate method (CCM)[8]. To be specific, we iteratively optimize the objective function by fixing $\mathbf{S}$ with respect to $\boldsymbol{\Theta}$ and vice versa. The major problem in such approach is that the quality of the solution depends highly on the initialization as it converges to a local minimum. To be specific, if we get a good estimation of $\mathbf{S}$, it is relatively easy to get a good solution for $\boldsymbol{\Theta}$ as the final loss function puts more emphasis on the correct region pair matching and vice versa.

Figure 2: An example for collecting training data with Google Map.

Thus, we propose a region based joint model on saliency estimation and region matching for training. Our pipeline is composed of five main steps:

1. *Image level (weak) label generation.* In this step, we generate the weak labels based on GPS information automatically. Google Map provides satellite images, as well as street view images, together with their GPS information to users, so that we can download them conveniently as database. Figure 2 (color masked) shows the region that our database for Boston dataset (introduced in Section 5) covers. The colored masks are drawn by Google Map automatically, indicating the buildings. After the generation of our database, we sample images in the range that the database covers randomly. Google Map provides four different perspectives of view from each point. If we just move the sampling point a little, the image with the same perspective would not change a lot. Thus, we can get two different images containing almost the same scene, and that is the rationality of our method to generate weak labels automatically. More specifically, we use Euclidean distance to judge whether two sampling points are near to each other. We set the matching label of two images q, d using

$$y_{(q,d)} = \begin{cases} 1 & dist(q, d) \le \theta \\ 0 & dist(q, d) > \theta \end{cases} \quad (7)$$

where q is the sampled image, d is the image from our database, $dist(q, d)$ is the Euclidean distance between the locations of q and d in the real world, θ is a hyper-parameter that denotes the distance threshold of matching. In addition, by zooming, rotating and shifting, we can augment our database to let it contain images with larger diversity. Thus, we have completed the generation of training data D_{train}. For database generation, we set $\theta = 25m$.

2. *Initialization of matching model.* In this step, we choose a good initialization of matching model. To be specific, we train a matching model $m(q_i, d_j; \boldsymbol{\Theta})$ directly using the image pairs. The structure of the network should be the same as that in the following procedure. That is, we start from the degenerate case where the entire image is treated as one region.

In our training procedure, we applied Siamese network for initializing the matching model. The structure of the Siamese network we applied is shown in Figure 3. The CNN is used to extract features from images. In this step, the region pairs are exactly image pairs

themselves. We will introduce more details about the structure of the network in *Step 4*.

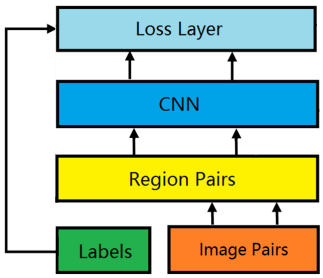

Figure 3: The structure of Siamese network.

3. *Saliency Estimation.* Apply the matching model $m(q_i, d_j; \mathbf{\Theta})$ on regions and optimize the main problem with respect to saliency $\mathbf{S}$. This problem can be decomposed to $|Q| + |D|$ separate linear objective quadratic constraint convex optimization problems:

$$\max_{\mathbf{s}(q)} \sum_{q_i \in q} (m_{i.}^{+} - m_{i.}^{-}) s(q_i)$$

where

$$\begin{aligned} m_{i.}^{+} &= \sum_{\{d_j | y(q_i,d_j)=1\}} m(q_i, d_j) \\ m_{i.}^{-} &= \sum_{\{d_j | y(q_i,d_j)=0\}} \min(0, m(q_i, d_j) - \Delta^{'}) \\ s.t. &||\mathbf{s}(q)||_2 = 1 \\ & s(q_i) \in [0, 1] \end{aligned} \tag{8}$$

We can add the other $|D|$ formulas with respect to d analogously. This problem has closed form solution:

$$\begin{aligned} \mathbf{m}_q &= [..., m_{i.}^{+} - m_{i.}^{-}, ...] \geq \mathbf{0} \\ \hat{\mathbf{s}}(q) &= \frac{\mathbf{m}_q}{||\mathbf{m_q}||_2} \end{aligned} \tag{9}$$

Here, $m_{i.}^{+}, m_{i.}^{-}$ correspond to the matching status of q_i to regions in positive labels and those in negative labels. Note that $m(q_i, d_j)$ is always non-negative due to our constraint in the section of problem formulation, $m_{i.}^{+}$ is always non-negative and $m_{i.}^{-}$ is always non-positive. Hence, we have the solution vector $\mathbf{m}_q = [..., m_{i.}^{+} - m_{i.}^{-}, ...] \geq \mathbf{0}$. Analogously, we can compute the solution for $\hat{\mathbf{s}}(d)$.

The solution saliency $\hat{\mathbf{S}} = \{\hat{\mathbf{s}}(q)\} \cup \{\hat{\mathbf{s}}(d)\}$ can be used to generate pseudo labels on region level. The label is called "pseudo" because we fix the parameter $\mathbf{\Theta}$ from an imperfect matching model $m(q_i, d_j; \mathbf{\Theta})$ to get the solution.

With the hypothesis that different parts of a building would not look very different, if an image pair (q, d) is labelled as matched, then the pair of salient regions (q_i, d_j) with high $s(q_i)$ and $s(d_j)$ should be matched with high level confidence. Oppositely, if an image pair (q, d) is labelled as unmatched, the pair of salient regions (q_i, d_j) with high $s(q_i)$ and $s(d_j)$ should be unmatched with high level confidence, too. Thus, we use Eq (2) (i.e. the sum of $s(q_i)$ and $s(d_j)$) to measure the confidence of the pseudo label $y_{(q,d)}$ for region pair (q_i, d_j).

4. *Region based matching optimization.* After estimating the saliency score of each region, we only use most confident pseudo region level labels to train the region matching model $m(q_i, d_j; \mathbf{\Theta})$. To refine the region level matching model, we minimize Eq (6) as the loss function with the Siamese network structure, which has been shown in Figure 3. Since the training samples are fed to the network one by one (or batch by batch), the loss function for a region pair (q_i, d_j) with label y should be rewritten as

$$L(q_i, d_j, y) = y \cdot m(q_i, d_j) + (1 - y) \cdot \max(0, \Delta^{'} - m(q_i, d_j)) \tag{10}$$

where $\Delta^{'}$ is the hyper-parameter to keep unmatched pairs having a small matching score.

5. We repeat step 3 and step 4 and gradually lower the confidence level for the pseudo label as the model becomes more and more accurate and we could rely on more pseudo labels generated. The confidence lowering procedure is controlled by the self-paced function[12].

In self-paced curriculum learning, a the self-paced function $f(\mathbf{v}; \lambda)$ determines a learning scheme, where $\mathbf{v} \in [0, 1]^n$ denotes a vector of weight variable for each training sample and λ controls the learning pace. In a self-paced function, $||\mathbf{v}||_1$ increases with respect to λ. For a fixed function $f(\mathbf{v}; \lambda)$, the sample selecting weight $\mathbf{v}$ can be computed by

$$\mathbf{v} = \underset{\mathbf{v} \in [0,1]^n}{\operatorname{argmin}} \sum v_i l_i + f(\mathbf{v}; \lambda) \tag{11}$$

where l_i is the loss function for the i^{th} sample.

To be more specific, we select training samples $\{(q_i, d_j, y)\}$by

$$\mathbf{v} = \underset{\mathbf{v} \in [0,1]^n}{\operatorname{argmin}} \sum v_{ij} \frac{1}{s(q_i) + s(d_j)} + f(\mathbf{v}; \lambda) \tag{12}$$

where n denotes the total number of our candidate samples (q_i, d_j, y). To make the loss function l_i satisfy the requirements of self-paced learning scheme [12], we set $l_i = \frac{1}{s(q_i)+s(d_j)}$, which is the multiplicative inverse of Eq (2).

The self-paced function we apply is

$$f(\mathbf{v}; \lambda) = \lambda - ||\mathbf{v}||_1 = -\lambda \sum_{i=1}^{n} v_i \tag{13}$$

representing a linear scheme.

Thus, we reformulate *main problem* into a self-paced version.

self-paced version of main problem:

$$\begin{aligned} min_{\mathbf{S},\mathbf{\Theta}} L(q, d, y; \mathbf{S}, \mathbf{\Theta}) = & -y \sum_{i,j} \sum_{y_{(q,d)}=1} w_{ij}\, m(q_i, d_j; \mathbf{\Theta}) \\ & - (1 - y) \sum_{i,j} \sum_{y_{(q,d)}=0} w_{ij} \max(\Delta^{'} - m(q_i, d_j; \mathbf{\Theta})) \\ & + f(w_{ij}; \lambda) \\ w_{ij} &= s(q_i) + s(d_j) \\ \mathbf{S} &= \{s(q_i)\} \cup \{s(d_j)\} \\ f(w_{ij}; \lambda) &= \lambda - ||w_{ij}||_1 \\ s.t. &||s(q)||_2 = 1 \quad \forall q \\ & ||s(d)||_2 = 1 \quad \forall d \\ & s(q_i), s(d_j) \in [0, 1] \\ & y \in \{+1, 0\} \end{aligned}$$

where λ is a hyper-parameter to control the pace of learning in the self-paced procedure.

In summary, our method can be described as Algorithm 1.

Algorithm 1 Joint Saliency Estimation and Matching Optimization

input: training dataset D_{train}, pre-trained matching model parameters Θ, self-paced function f and stepsize μ
output: saliency score S and refined matching parameters Θ^*

```
Initialize S, λ and the matching model Θ* = Θ
while not converged do
    Estimate saliency score with Eq (9)
    Generate region level pseudo labels using image level labels
    Select "easy" samples by Eq (12)
    Refine Θ* with the loss function Eq (10)
    if λ is small then increase λ by μ
end while
return S, Θ*
```

4.2 Inference

After refining the matching model, we propose a corresponding region-based metric to measure the similarity between two images q and d, which is

$$sim_R(q,d) = -\sum_{q_i \in q} \min_{d_j \in d} \frac{|q_i|}{|q|} w_{ij} ||f(q_i) - f(d_j)||_2 \qquad (14)$$

where $|q_i|$ and $|q|$ is the size of region q_i and image q, $f(q)$ and $f(d)$ are the vector representations of the two images in the feature space.

Our method of video geo-localization is based on such fact: if a video has not been edited, the geo-location of it should not vary a lot in a short time. To find the geo-location of a video, we first retrieve K most similar images in the database for each frame, and collect $K \times F$ images (may overlap) in the database. Then, we cluster the images in the database into C clusters, based on their representative vectors in the feature space. The video will be localized to the average GPS coordinate of the largest cluster.

5 Experiments

In this section, we first give introduction to the two datasets we used to test our model; and then introduced the details in our implementation in experiments; next, we briefly introduce the evaluation metrics and baseline we applied. We report the experimental results on joint model based geo-localization and saliency estimation in the last two sub-sections, and give some analysis.

5.1 Datasets

We use two challenging datasets, Boston Marathon 2013[6] and Tokyo Time Machine [1]. Table 2 shows the summary of the two datasets we used.

5.1.1 Boston Dataset Chen et al. [6] collected video clips which are related to the event of Boston Marathon bombing from the internet. This is a dataset which focus on a real event. In the videos, the streets look very different from daily appearance due to the event. Figure 4 shows some example videos. In this dataset, they also provide 10,000 satellite environment images in the surrounding region of the event location. The region contains 562 buildings in total. We uniformly sampled 2,500 GPS locations in this region. For each GPS coordinate, we collected 4 images from 4 different perspectives of view (10,000 images in total). We also collected the same number of images as queries for training at randomly sampled GPS locations in the same way to collect database. For testing, we labeled the ground truth of geo-location for 100 sampled frames from 37 45-degree-view video clips. Some example frames are shown in Figure 4. Each frame may have multiple relevant images in the database.

Figure 4: Sample video clips collected from the internet.

5.1.2 Tokyo Time Machine Dataset Tokyo Time Machine (TokyoTM) dataset was released by Arandjelovic et al. [1], generated from downloaded Time Machine panoramas. In this dataset, the appearances of buildings also change a lot, since the pictures were taken at different time in different days.

Dataset	Database	Query set
Tokyo Time Machine-train	49,104	7,277
Tokyo Time Machine-val	49,056	7,186
Tokyo 24/7 (-test)	75,984	1,125
Boston-train	8,000	8,000
Boston-val	2,000	2,000
Boston-test	10,000	37 video clips

Table 2: The size of the dataset we used in the experiments. The train/val(idation)/test dataset is mutually disjoint geographically.

5.2 Implementation Details

In the following experiments, we use Caffe [10] as the deep learning framework.

We divide an image into a "pyramid", where the first layer is the image itself, the second layer contains 2×2 sub-images and the third layer contains 3×3 sub-images. Thus, each image has 14 regions when we apply our saliency guided geo-localization method. The saliency score of each region is estimated separately. Figure 5 shows how we generate regions for an image in the experiments.

Figure 5: The regions of an image in our experiments.

To compare with Eq (14), we also report the experimental results with another metric of similarity

$$sim(q,d) = -||f(q) - f(d)||_2 \quad (15)$$

which is directly the additional inverse of the Euclidean distance between two representative vectors in the feature space.

5.3 Evaluation Metrics and Baselines

We test the performance of our model with adopting two common metrics: mean average precision (MAP) and recall on top$N$:
1. *MAP* For a query image, the average precision of an image retrieval task can be computed by $AveP = \frac{\sum_{k=1}^{n} P(k)\times rel(k)}{number\ of\ relevant\ images}$, where $P(k)$ is the precision at cut-off k in the list; $rel(k)$ is an indicator function equalling 1 if the image at rank k is a relevant image, 0 otherwise. Then $MAP = \frac{\sum_{i=1}^{|Q|} AveP(i)}{|Q|}$, where $|Q|$ is the number of queries. This metric shows the general performance for a model.
2. *Recall on Top N*. We retrieve N most similar images for each query. Therefore $RecallontopN = \frac{\sum_{i=1}^{|Q|} \#RetrievedRelevantImage(i)}{\sum_{i=1}^{|Q|} \#RelevantImage(i)}$, where # represents number. This metric is helpful to reveal the application value for a method, since we often retrieve top N images for image retrieval tasks, like geo-localization.

We applied our method introduced in Section 4 with VGG-CNN-M model as a base CNN model and we introduce the following baselines:

- VGG-CNN-M-*sim*: it calculates similarity *sim* by Eq (15) using the $fc7$ layer output of the model released by Chatfield et al. [5].
- VGG-CNN-M-sim_R: it estimates the saliency score using the VGG-CNN-M model, and then compute the similarity between image q and d by Eq (14).
- IBFT-*sim*: it applies image-level based fine-tuning to the matching model, and use Eq (15) as the metric of similarity.
- JSEM-sim_R: it is the full joint saliency estimation and matching model with the similarity measurement Eq (14) proposed in this paper.
- Manually-labeled-sim_R: For Boston dataset, we also manually label the building contour in the image using LabelMe [20]. We treat building areas as foreground and other areas as background. We manually set the saliency value of a region based on its intersection percentage with the foreground, and use this saliency value from additional building contour labeling in similarity measurement sim_R. It could be treated as a kind of upper bound.

5.4 Evaluation of Geo-Localization

Model	Boston	Tokyo TM
VGG-CNN-M-*sim*	28.85	26.83
VGG-CNN-M-sim_R	35.67	24.34
IBFT-*sim*	37.50	23.91
JSEM-sim_R	**58.33**	**37.01**
Manually-labeled-sim_R	63.40	-

Table 3: MAP (×100) on the two datasets.

Boston: The result reported in Table 3 shows that our model has significantly improved the mean average precision (MAP) on the geo-localization task. For the convenience of view, Table 4 shows some case study on this task, from which we can easily see that the rankings of ground truth images raise a lot. In CNN, we resize the frames into a 224×224 square shape. Thus, we show the frames and database in such square shapes rather than rectangular ones in Figure 4. Besides, the "recall on top K" result on Boston dataset is reported in Table 5.

Query Frame				
Ground Truth				
Ground Truth Ranking in All Database Images				
VGG-CNN-M-*sim*	6	3	10	108
JSEM-sim_R	1	1	1	12

Table 4: Case study on Boston Dataset.

TokyoTM: We also run experiments on the Tokyo TM dataset and show results in Table 3 and Table 6.

The experimental result had proved the efficiency of our model, showing that our model obtains the best recall for every N, and beats the baselines on the metric MAP. That is, our model not only performs well generally, but also has application value for real image retrieval tasks. However, it's interesting that the model IBFT-*sim* performs similar to VGG-CNN-M-*sim* on Tokyo TM dataset. It suggests that fine-tuning based only on the limited training dataset could easily cause over-fitting. This also shows that saliency score is an important part to improve the performance of the model, and make the model more robust.

5.5 Analysis of Learned Saliency

To give further supporting evidences for the effectiveness of our method, we analyze the learned saliency qualitatively and quantitatively. We first conduct case study on some salient and non-salient regions, and then report the Pearson correlation with manually labeled saliency score on some randomly selected samples.

We applied the method introduced in Section 4 to estimate the saliency score for each region of a given image. Figure 6 shows the 8 most salient and 8 most non-salient regions together with their saliency scores from the Tokyo TM database. The salient regions are all buildings with distinctive structure($a - h$) while the non-salient regions are images from road($i - k$), trees(m and n), sky(o) and indistinctive building regions(l and p).

We also analyze the correlation between learned saliency with saliency calculate from manually labeled building contour in manually-labeled-sim_R, which could be treated as ground truth saliency score. The Pearson correlation is 0.7625, showing that our estimated saliency score has strong relation with the ground truth saliency score. Figure 7 shows the case study of our region based saliency estimation on two specific images.

figure 7a 0.5974522293 0.3690895466 0.3680404646 0.9743274348 0.4267515924 0.4075309105 0.4587860622 0.4804421132 0.4638066692

figure 7b 0.3754215062 0.4274634695 0.3560884226 0.5809666542 0.8801049082 0.4672161858 0.4428250281 0.4357437242 0.4052828775

Model	N = 1	N = 2	N = 3	N = 4	N = 5	N = 10	N = 15	N = 20	N = 25
VGG-CNN-M-*sim*	7.05	13.85	21.41	25.44	30.23	44.33	51.89	62.97	69.77
VGG-CNN-M-sim_R	8.31	15.87	23.43	30.23	35.52	53.40	73.05	80.35	84.13
IBFT-*sim*	7.56	15.62	23.93	37.03	42.82	56.93	67.25	78.84	85.89
JSEM-sim_R	16.12	27.96	39.80	51.39	58.44	78.84	84.63	89.67	95.21

Table 5: Recall on top N on Boston dataset.

Model	N = 1	N = 2	N = 3	N = 4	N = 5	N = 10	N = 15	N = 20	N = 25
VGG-CNN-M-*sim*	5.67	10.17	14.42	17.97	20.33	32.86	46.57	58.16	68.32
VGG-CNN-M-sim_R	5.91	10.87	15.84	18.68	21.04	36.17	47.28	61.47	70.92
IBFT-*sim*	7.57	11.58	15.60	18.91	23.17	34.04	46.57	55.79	68.09
JSEM-sim_R	9.22	15.60	21.28	25.77	28.84	48.70	63.83	74.23	84.87

Table 6: Recall on Top N on Tokyo TM dataset.

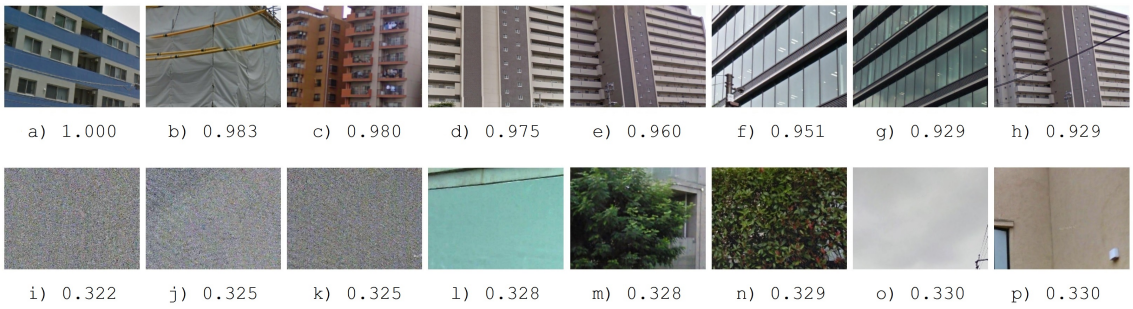

Figure 6: 8 most salient regions(top) and 8 most non-salient regions in Tokyo TM database, with their saliency score.

Figure 7: Case study of region-based saliency estimation.

6 Conclusions and Future Work

We study automatic geo-localization of online event videos, which is the key component supporting various event analysis tasks such as cross camera tracking and 3d reconstruction. The major challenge lies in large appearance change of the environment due to human activity during the event. To solve this problem, we formulate the problem as region based joint saliency estimation and matching, and propose a solution, which iteratively estimates saliency and optimizes region based matching model. For the step of saliency optimization, we derived a closed form solution, which has an intuitive explanation. For region based matching model optimization, we apply self-paced learning algorithm to learn from the pseudo labels generated by saliency value estimated by matching. We conduct experiments on Boston Marathon 2013 dataset and Tokyo Time Machine datas et to test the performance of our model. The experiments have shown that our solution significantly improves over matching on whole image and the automatically learned saliency has a strong correlation with manual labels on region saliency, proving that our model has significant application value.

In the future, we will try more sophisticated CNN based models such as Resnet and NetVLAD in our matching model to improve performance. As the cost of manually labeling all the exact matching point or region is unaffordable, weak labels are common situation in matching problems. We will generalize this problem formulation to matching problems in different tasks at different levels: interest points and regions. We also hope to design a framework based on current solution for a generalized problem formulation.

References

[1] R. Arandjelović, P. Gronat, A. Torii, T. Pajdla, and J. Sivic. 2016. NetVLAD: CNN architecture for weakly supervised place recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*.

[2] Relja Arandjelović and Andrew Zisserman. 2013. All about VLAD. In *Computer Vision and Pattern Recognition, 2013. CVPR 2013. IEEE Conference on*. IEEE.

[3] Mayank Bansal, Kostas Daniilidis, and Harpreet Sawhney. 2012. Ultra-wide baseline facade matching for geo-localization. In *European Conference on Computer Vision*. Springer, 175–186.

[4] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*. ACM, 41–48.

[5] Ken Chatfield, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. 2014. Return of the devil in the details: Delving deep into convolutional nets. *arXiv preprint arXiv:1405.3531* (2014).

[6] Jia Chen, Junwei Liang, Han Lu, Shoou-I Yu, and Alexander G. Hauptmann. 2016. Videos from the 2013 Boston Marathon: An Event Reconstruction Dataset for Synchronization and Localization.

[7] Christopher B Choy, JunYoung Gwak, Silvio Savarese, and Manmohan Chandraker. 2016. Universal Correspondence Network. In *Advances in Neural Information Processing Systems 29*.

[8] Jochen Gorski, Frank Pfeuffer, and Kathrin Klamroth. 2007. Biconvex sets and optimization with biconvex functions: a survey and extensions. *Math. Meth. of OR* 66, 3 (2007), 373–407. https://doi.org/10.1007/s00186-007-0161-1

[9] James Hays and Alexei A Efros. 2008. IM2GPS: estimating geographic information from a single image. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. IEEE, 1–8.

[10] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. 2014. Caffe: Convolutional Architecture for Fast Feature Embedding. *arXiv preprint arXiv:1408.5093* (2014).

[11] Lu Jiang, Deyu Meng, Teruko Mitamura, and Alexander G Hauptmann. 2014. Easy samples first: Self-paced reranking for zero-example multimedia search. In *Proceedings of the 22nd ACM international conference on Multimedia*. ACM, 547–556.

[12] Lu Jiang, Deyu Meng, Qian Zhao, Shiguang Shan, and Alexander G Hauptmann. 2015. Self-Paced Curriculum Learning.. In *AAAI*, Vol. 2. 6.

[13] M Pawan Kumar, Benjamin Packer, and Daphne Koller. 2010. Self-paced learning for latent variable models. In *Advances in Neural Information Processing Systems*. 1189–1197.

[14] Keith Yu Kit Leung, Christopher M Clark, and Jan P Huissoon. 2008. Localization in urban environments by matching ground level video images with an aerial image. In *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*. IEEE, 551–556.

[15] Tsung-Yi Lin, Serge Belongie, and James Hays. 2013. Cross-view image geolocalization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 891–898.

[16] Tsung-Yi Lin, Yin Cui, Serge Belongie, and James Hays. 2015. Learning deep representations for ground-to-aerial geolocalization. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 5007–5015.

[17] Volodymyr Mnih, Nicolas Heess, Alex Graves, and others. 2014. Recurrent models of visual attention. In *Advances in Neural Information Processing Systems*. 2204–2212.

[18] James Philbin, Ondrej Chum, Michael Isard, Josef Sivic, and Andrew Zisserman. 2007. Object retrieval with large vocabularies and fast spatial matching. In *2007 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2007), 18-23 June 2007, Minneapolis, Minnesota, USA*.

[19] James Philbin, Ondrej Chum, Michael Isard, Josef Sivic, and Andrew Zisserman. 2008. Lost in quantization: Improving particular object retrieval in large scale image databases. In *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2008), 24-26 June 2008, Anchorage, Alaska, USA*.

[20] Bryan C. Russell, Antonio Torralba, Kevin P. Murphy, and William T. Freeman. 2008. LabelMe: A Database and Web-Based Tool for Image Annotation. *International Journal of Computer Vision* 77, 1 (2008), 157–173.

[21] Kevin Tang, Vignesh Ramanathan, Li Fei-Fei, and Daphne Koller. 2012. Shifting weights: Adapting object detectors from image to video. In *Advances in Neural Information Processing Systems*. 638–646.

[22] Nam N Vo and James Hays. 2016. Localizing and Orienting Street Views Using Overhead Imagery. In *European Conference on Computer Vision*. Springer, 494–509.

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

[24] Yin Zheng, Richard S Zemel, Yu-Jin Zhang, and Hugo Larochelle. 2015. A neural autoregressive approach to attention-based recognition. *International Journal of Computer Vision* 113, 1 (2015), 67–79.