Location Estimation of Urban Images Based on Geographical Neighborhoods

Jie Huang\textsuperscript{1, a} and Sio-Long Lo\textsuperscript{1, b}

\textsuperscript{1} Faculty of Information Technology, Macau University of Science and Technology, Macau S.A.R., China
\textsuperscript{a} must.jhuang@gmail.com; \textsuperscript{b} sllo@must.edu.mo

Abstract. Estimating the location of an image is a challenging computer vision problem, and the recent decade has witnessed increasing research efforts towards the solution of this problem. In this paper, we propose a new approach to the location estimation of images taken in urban environments. Experiments are conducted to quantitatively compare the estimation accuracy of our approach, against three representative approaches in the existing literature, using a recently published dataset of over 150 thousand Google Street View images and 259 user uploaded images as queries. According to the experimental results, our approach outperforms three baseline approaches and shows its robustness across different distance thresholds.

1. Introduction

Location estimation of images is a challenging computer vision task, with the aim of estimating the geographical location where a query image was captured. The increasing amount of images available on the Internet provide an opportunity for research in location estimation of images. Attention from the research community has been drawn to the location estimation of images [1, 2, 3, 4, 5, 6] in the recent decade, due to the prospect of various applications in landmark recognition [7], city reconstruction [8], and robot vision [9].

One of the pioneering approaches was proposed by Hays and Efros in [3], where they adopted a data-driven approach to estimating the location of a query image, when given a large reference database of various images captured from all across the world. Succeeding research efforts have been devoted to addressing the problem of localizing social network images [10, 11], street view images [12, 13, 14], etc.

There are two related tasks for image geo-localization. One is finding a matching scheme, with the purpose of finding correspondences between image descriptors of the query image and the reference images in the database. Sattler et al. formulated the problem of image geo-localization, as a descriptor matching problem [15]. The other is location estimation of the query image, based on the matching results. However, we are more interested in the latter, i.e., how to make a location estimate of the query based on the matches returned by a matching scheme, for example, image retrieval.

Our goal in this paper is to explore the possibilities of improving the location estimation accuracy of images captured in urban environments. We note that the distribution of reference images in the database has an impact on the estimation accuracy, i.e., if there are sufficient reference images taken at, or near the query location in the database, it is more likely for the query image to be geo-localized correctly. Moreover, the visual content of query images also has an influence on the estimation accuracy. Location estimation of generic images (for example, images of the sky, plants, and indoor scenes) are beyond the scope of this paper. We confine our discussion to the problem of location estimation of urban images captured outdoors.
To pursue the goal mentioned above, we propose an approach that consists of three major steps. In the first step, visual candidates retrieval, we retrieve multiple most similar images (or visual candidates) from the reference database, for each query image. Then we build a neighborhood for every candidate, which includes images of the shortest geographical distance to the candidate, in the second step, geo-neighborhood construction. Finally, for each neighborhood, we compute the average visual similarity to the query, based on which we rank all the visual candidates. The location of the highest ranked candidate will be propagated to the query image, as the location estimate. The novelty of our approach lies in addressing the shortcomings of existing approaches including [3, 11], and improvement in accuracy of location estimation compared to these approaches.

The remainder of this paper is structured as follows. Section 2 reviews related work on location estimation of images. Section 3 introduces our approach to location estimation of urban images in more detail. The setup of our experiments is presented in Section 4. Section 5 demonstrates the performance of our approach, against three baseline approaches. We conclude the paper in Section 6.

2. Related Work

In this section, we will review a number of retrieval-based approaches related to our research work. Conventionally, image retrieval returns one or multiple images with matching descriptors to the query image. And retrieval-based approaches estimate the query location based on the retrieved images. Hays and Efros first proposed adopting first-nearest neighbor (1-NN) in estimating the location of a single image [3]. Essentially, when there is only one candidate image for the query, its location will be directly propagated to the query image. This approach assumes that if two images are visually similar, then they are likely to be in the proximity of each other geographically. One weakness of directly propagating the location of the 1-NN to the query is that it is likely to fail in cases when two images are visually similar, but continents away from each other.

The idea of extracting multiple visual nearest neighbors (NNs) of the query, then clustering the location of the NNs using mean-shift clustering was proposed in [3]. Eventually, the mode location of the largest cluster will be assigned to the query image. The assumption behind the clustering approach is that if a location cluster has more images than the others, the mode of which is more likely to be the closest to the query location. However, the assumption does not hold for cases when the location closest to the query location does not have sufficient reference images in the database, to become the largest cluster.

Li et al. proposed and matured the idea of “geo-visual ranking” (GVR) [10, 11] as an alternative approach to location estimation, overcoming the weakness of both approaches proposed in [3]. The GVR approach first extracts $k$ nearest neighbors of the query, then assigns $k$ images into different clusters based on their geo-location. Finally, the top $m$ clusters, where $m < k$, are ranked based on the aggregate visual similarity to the query. The location estimate of the query, will be the centroid of GPS coordinates of all the images in the highest ranked cluster [11]. The shortcoming of the ranking approach [11] is that it requires pre-defining a number of parameters during the location estimation process.

3. Location Estimation Based on Geo-Neighborhoods

Our approach assumes that, images in the proximity of one another geographically, have visual elements in common thus making them appear visually similar. Hence, we propose to first retrieve multiple visual candidates of a query, then evaluate each candidate based on the visual similarity of its neighborhood to the query. Geo-neighbors in a candidate’s neighborhood are defined as the images of the shortest geographical distance to a candidate. The rationale is the neighbors of a candidate provide additional visual information of a neighborhood where a candidate is located. If the visual similarity of a neighborhood to the query is high, this can serve as supportive evidence that a candidate is visually similar to the query. There are three steps in our approach, namely, 1) visual candidates retrieval, 2) geo-neighborhood construction, and 3) visual candidates ranking. In the succeeding subsections, we will elaborate on each of the steps in detail.
3.1. Visual candidates retrieval

When given a query image $q$, the purpose of this step is to retrieve images that are most likely to have been captured at the same location as the query, solely based on the visual content. Hence, we compute the visual similarity between the query $q$ and each image in the reference database $D = \{I_i\}_{i=1}^{N}$, and further retrieve the images of the highest similarity. The k-nearest neighbor (kNN) classifier will be applied to retrieve visually similar images from the databases $D$, given that there exist multiple views of a single location in the database adopted in our experiments. As a result, $K$ visual nearest neighbors of the query will be retrieved from the database as $N_K(q)$

$$N_K(q) = \{(I_i^{(q)}, d_i^{(q)})\}_{i=1}^{K}$$  \hspace{1cm} (1)

Where $I_i^{(q)} \in D$, and $d_i^{(q)} \in \mathbb{R}_{>0}$ is the visual distance, or dissimilarity between a database image and the query. In addition, 2-norm (L2) distance is applied to computing the dissimilarity between two images when using the kNN classifier

$$\text{dist}_{\text{vis}}(p, q) = \|f_p - f_q\|_2$$  \hspace{1cm} (2)

Where $f_p$ and $f_q$ are the feature representation of image $p$ and $q$. To represent an image with a feature vector, gist descriptors [16, 17] are adopted to describe an image on the whole, due to its usefulness in semantic image retrieval [18]. In addition, we accommodate the gist descriptors of an image in a 600-dimensional feature vector.

3.2. Geo-neighborhood construction

After retrieving $K$ visual candidates of the query image, the aim of the second step is to create a geographical neighborhood for each candidate $r \in N_K(q)$, by extracting from the reference database $D$, images of the shortest geographical distance to each candidate. For a visual candidate $r \in N_K(q)$, its ground-truth geo-location can be found in the corresponding GPS database $G : D \rightarrow L \in R^2$, $l_i = G(I_i)$, and $l_i = (lat_i, lon_i)$ for $i = 1, 2, \ldots, N$ are the latitude and longitude of image $I_i$ in the reference database $D$. The $M$-geographical neighbors of a candidate $r$ are defined as images of the shortest geographical distance to $r$ as

$$R_M(r) = \{(I_i^{(r)}, h_i^{(r)})\}_{i=1}^{M}$$  \hspace{1cm} (3)

Where $I_i \in D$, $h_i^{(r)} \in \mathbb{R}_{>0}$ is the geographical distance to the location of candidate $r$, and the value of $M$ varies among visual candidates. The Haversine formula [19], denoted $H(p, l_i)$, is applied to computing the geographical distance between GPS coordinates of a candidate $p = G(r)$ and those of images in the reference database $l_i = G(I_i)$ where $I_i \in D$. Images $I_i \in D$ of the shortest Haversine distance to the candidate $r$ are extracted and form a neighborhood of the candidate $r$.

The rationale of this step is, the neighbors $R_M(r)$ of a visual candidate $r$ provide additional visual information regarding the geo-location of $r$. If the visual similarity of the neighborhood of a candidate $r$ is high, this can serve as additional evidence that $r$ is visually similar to the query. The approach by Li et al. [10], where they also consider extracting all the images within a certain radius (for example, 1 kilometer), in order to build the neighborhood of a visual candidates, is prone to an unbalanced number of neighbors for each candidate when constructing a neighborhood. Instead, we consider extracting only images of the shortest geographical distance to the candidate, and compute the average similarity of all the images in the neighborhood, including the candidate itself, in order to overcome the bias caused by the unbalanced number of neighbors for each candidate as proposed in [10].
3.3. Visual candidates ranking

To obtain a reasonable geo-location estimate, we first calculate the visual distance between the neighborhood of every candidate \( r, R(r) = R_M(r) \cup \{ (r, 0) \}, r \in N_K(q) \), and the query image \( q \). We define the average visual distance \( \text{dist}_{\text{rank}}(R(r), q) \) between the neighborhood \( R(r) \) of a candidate \( r \), and the query image \( q \) as

\[
\text{dist}_{\text{rank}}(R(r), q) = \frac{1}{|R(r)|} \cdot \sum_{\alpha \in R(r)} \text{dist}_{\text{vis}}(\alpha, q) \tag{4}
\]

Where \( |R(r)| \) is the number of images in \( R(r) \), and \( \alpha \in R(r) \) represents every image belonging to the neighborhood \( R(r) \). We rank all the visual candidates \( r \in N_K(q) \) based on the average visual distance of their corresponding neighborhood \( R(r) \).

\[
r^* = \arg \min_{r \in N_K(q)} \text{dist}_{\text{rank}}(R(r), q) \tag{5}
\]

Finally, GPS coordinates \( l_{r^*} \) of the candidate \( r^* \), which has the highest neighborhood similarity, will be propagated to the query \( q \), as the location estimate, \( l_q = l_{r^*} = G(r^*) \).

4. Experimental Setup

4.1. Dataset

To perform a fair comparison, we evaluate the location estimation accuracy of our approach against three baseline approaches on a Google Street View Dataset contributed by Zemene et al. in their recent work [14], comprising of 152,536 images taken in 14 cities from different parts of the world, along with a separate query dataset of 259 images, both after elimination of faulty images. The distribution of reference and query images in the UCF Google Street View Dataset [14] is shown below in Table 1.

| Continent/country | City         | # of reference images | # of query images |
|-------------------|--------------|-----------------------|-------------------|
| Europe (5 cities) | Amsterdam    | 15,608                | 17                |
|                   | Frankfurt    | 8,788                 | 0                 |
|                   | Rome         | 15,729                | 13                |
|                   | Milan        | 15,986                | 2                 |
|                   | Paris        | 16,588                | 28                |
| Australia (2 cities) | Sydney     | 15,986                | 16                |
|                   | Melbourne    | 14,190                | 10                |
| USA (7 cities)    | Las Vegas    | 5,950                 | 54                |
|                   | Los Angeles  | 12,610                | 13                |
|                   | Phoenix      | 4,233                 | 13                |
|                   | Houston      | 10,361                | 11                |
|                   | San Diego    | 3,079                 | 15                |
|                   | Dallas       | 4,740                 | 18                |
|                   | Chicago      | 8,688                 | 49                |
| Total number      | 152,536      | 259                   |
Figure 1 shows sample reference images and queries from the UCF Google Street View Dataset [14]. Each location is represented with at most five reference images in the database. As demonstrated in Figure 1a, the reference images depict each location from different perspectives, and are complementary to one another. In Figure 1b, twelve sample query images are demonstrated. The visual content of query images varies, however, most queries are ground-level images.

(a) Twelve sample reference images.  
(b) Twelve sample queries.

Figure 1. Sample reference images and queries from the UCF Google Street View Dataset [14].

4.2. Experimental design and evaluation metric
To compute the visual similarity between images, we represent each image with a feature vector, then compute the distance between two feature vectors. The shorter the distance, the more similar two images are. In our experiments, we build a feature vector of gist descriptors[16, 17] for every image, with the spatial resolution of 5 by 5, where every bin stores the average response of the image region to filters at 6 orientations and 4 scales. We arrive at a 600-dimensional vector for every image and apply L2 distance to computing the distance between feature vectors. The settings are identical to those used in [3].

In order to evaluate the accuracy of location estimation, we define a distance threshold, $\theta_d$, which controls the evaluation accuracy and tolerance to noise in ground-truth GPS coordinates [11]. An image is considered to be geo-localized correctly, if its estimated coordinates, denoted $l_q^*$, fall within the distance $\theta_d$, of the ground-truth location $l_q$. As formally expressed in Eq.6, the correctness of geo-localization of an image can be obtained using the function $f_{\theta_d}$, with respect to the distance threshold $\theta_d$, where $\text{dist}(l_q, l_q^*)$ refers to the geographical distance between $l_q$ and $l_q^*$, calculated using the Haversine formula [19].

$$f_{\theta_d}(l_q, l_q^*) = \begin{cases} 
\text{correct,} & \text{if dist}(l_q, l_q^*) \leq \theta_d \\
\text{incorrect,} & \text{otherwise}
\end{cases} \tag{6}$$

4.3. Baseline approaches
The three baseline approaches, against which we compare our approach, in terms of geo-localization accuracy, are all retrieval-based approaches, i.e., approaches that first retrieve visually similar images(or visual nearest neighbors) of a query, then make an estimate of the query location based on the location of retrieved images.

- 1-NN: Following [3], we retrieve from the reference database, the most similar image to the query using the k-nearest neighbor classifier, then assign the location of the retrieved image to the query, as the location estimate.
- Clustering: Instead of retrieving the most similar image, $k$ images of the highest visual similarity are retrieved [3]. These retrieved images will be assigned to the corresponding cluster using mean-shift clustering algorithm. Eventually, the cluster with the most images will be selected, and GPS coordinates of the cluster centroid will be propagated to the query.
• Ranking: The ranking approach [11], used as a baseline in our experiments, is an extension of the one proposed in [10]. $k$ nearest neighbors will first be retrieved for the query, then the neighbors will be assigned into different clusters, based on their ground-truth location. However, only the top $m$ cluster, where $m < k$, will enter the location estimation process. Each cluster will be given a ranking score, i.e., the sum visual similarity of images in the cluster to the query. The cluster with the highest ranking score will be selected, and the centroid of GPS coordinates of all the images will be the location estimate of the query.

In our experiments, when applying the k-nearest neighbor classifier, we set the number of nearest neighbors to 20, i.e., when given a query image, we retrieve 20 images of the highest visual similarity to the query. Moreover, the bandwidth is set to 0.2 for mean-shift clustering. Regarding the ranking approach [11], we also set the number of nearest neighbors to 20 during image retrieval, while the number of clusters eligible for the location estimation process to 10, following the optimal parameter combination in [11].

5. Experimental Results

5.1. Comparison with the baseline approaches

As reported in Table 2, our approach outperforms three baseline approaches in terms of location estimation accuracy, across all distance thresholds. Moreover, one of Hays and Efros’ findings about the performance gap between 1-NN and the clustering approach [3] is also validated in our experiments: 1-NN performs better when it comes to precise location estimation. In our experiments, 1-NN yields better accuracy when the distance threshold is set to 25 or 200 kilometers. While the clustering approach outperforms the 1-NN approach, in terms of accuracy, as the distances threshold grows towards 2,500 kilometers.

| Distance threshold (km) | 25    | 200   | 750   | 2,500 |
|------------------------|-------|-------|-------|-------|
| 1-NN                   | 8.49% | 9.27% | 15.06%| 27.03%|
| Clustering             | 6.18% | 6.95% | 15.44%| 27.41%|
| Ranking                | 8.88% | 8.88% | 20.08%| 35.91%|
| Ours                   | **10.42%** | **10.42%** | **20.85%** | **36.68%** |

5.2. Discussion

Compared to 1-NN [3] and the clustering approach [3], the advantage of our approach is its robustness. Hays and Efros concluded in [3] that 1-NN performs better at precise location estimation, while the clustering approach is more suitable for global-scale localization. Our approach, however, results in better performance than both baselines when measured with different distance thresholds. The estimation accuracy of our approach also exceeds that of the ranking approach [11], where we have witnessed an encouraging increase of 1.54% in estimation accuracy, when the distance threshold is 25 or 200 kilometers, and a slight increase in accuracy as well, when the threshold is set to 750 or 2,500 kilometers.

In addition, our approach does not require tuning any additional parameters after retrieving visual candidates of a query. Moreover, we overcome the problem of the unbalanced number of images in the neighborhood among visual candidates, introduced by the ranking approach in [10]. The problem is tackled by ranking the candidates on the basis of the average similarity of images in the neighborhood, where it is guaranteed that each neighborhood will have at least one neighbor for the candidate.
6. Conclusion
We have proposed a new approach to location estimation of urban images, based on geographical neighborhoods. Our approach consists of three steps, namely, visual candidates retrieval, geo-neighborhood construction, and visual candidates ranking. Experiments have been carried out to compare the performance of our approach against three baseline approaches. The results demonstrate the robustness of our approach across different distance thresholds. In addition, our approach has outperformed all three baseline approaches, in terms of estimation accuracy. In the future work, we plan to retrieve visual candidates of a query using additional geo-informative features, and improve on how we build the geo-neighborhood for a visual candidate.

Acknowledgment
This work was supported partially by the Macao Science and Technology Development Fund grants (No. 019/2014/A1 and No. 112/2014/A3).

References
[1] Zhang W and Kosecka J 2006 3D Data Processing, Visualization, and Transmission, Third International Symposium on (IEEE) pp 33–40
[2] Philbin J, Chum O, Isard M, Sivic J and Zisserman A 2007 Computer Vision and Pattern Recognition, 2007. CVPR 2007. IEEE Conference on (IEEE) pp 1–8
[3] Hays J and Efros A A 2008 Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on (IEEE) pp 1–8
[4] Serdyukov P, Murdoch V and Van Zwol R 2009 Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval (ACM) pp 484–491
[5] Crandall D J, Backstrom L, Huttenlocher D and Kleinberg J 2009 Proceedings of the 18th international conference on World wide web (ACM) pp 761–770
[6] Irshara A, Zach C, Fram J M and Bischof H 2009 Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on (IEEE) pp 2599–2606
[7] Chen D M, Baatz G, Köser K, Tsai S S, Vedantham R, Pylvänäinen T, Roimela K, Chen X, Bach J, Pollefeys M et al. 2011 Computer Vision and Pattern Recognition, 2011. CVPR 2011. IEEE Conference on (IEEE) pp 737–744
[8] Tuite K, Snively N, Hsiao D, Tabing N and Popovic Z 2011 Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (ACM) pp 1383–1392
[9] Meier L, Tanskanen P, Fraundorfer F and Pollefeys M 2011 Robotics and automation (ICRA), 2011 IEEE international conference on (IEEE) pp 2992–2997
[10] Li X, Larson M and Hanjalic A 2013 Proceedings of the 3rd ACM conference on International conference on multimedia retrieval (ACM) pp 81–88
[11] Li X, Larson M and Hanjalic A 2015 IEEE Transactions on Multimedia 17 674–686
[12] Zamir A R and Shah M 2010 European Conference on Computer Vision (Springer) pp 255–268
[13] Zamir A R and Shah M 2014 IEEE Transactions on Pattern Analysis and Machine Intelligence 36 1546–1558
[14] Zemene E, Tariku Y, Idriss H, Prati A, Pelillo M and Shah M 2017 arXiv preprint arXiv:1702.01238
[15] Sattler T, Weyand T, Leibe B and Kobbelt L 2012 BMVC vol 1 p 4
[16] Oliva A and Torralba A 2001 International journal of computer vision 42 145–175
[17] Oliva A and Torralba A 2006 Progress in brain research 155 23–36
[18] Hays J and Efros A A 2007 ACM Transactions on Graphics (TOG) vol 26 (ACM) p 4
[19] Robusto C C 1957 The American Mathematical Monthly 64 38–40