City Scale Image Geolocalization via Dense Scene Alignment

Semih Yagcioglu Erkut Erdem Aykut Erdem
Department of Computer Engineering
Hacettepe University, Ankara, TURKEY
semih.yagcioglu@hacettepe.edu.tr {erkut, aykut}@cs.hacettepe.edu.tr

Abstract

Predicting where a photo was taken is quite important and yet a challenging task for computer vision algorithms. Our motivation is to solve this difficult problem in a city-scale setting by employing a data-driven approach. In order to pursue this goal, we developed a fast and robust scene matching method that follows a coarse-to-fine strategy. In particular, we combine scene retrieval via global features and dense scene alignment and use a large set of geo-tagged images of downtown San Francisco in our evaluation. The experimental results show that the proposed approach, despite its simplicity, is surprisingly effective and achieves comparable results with the state-of-the-art.

1. Introduction

In this paper, we focus on solving the city scale scene geolocalization problem. Image geolocalization and its variants are among the most popular problems investigated by the computer vision community in recent years. With the increase in the amount of visual data and the publicly available datasets, it is now possible to address this problem from a data-driven perspective by matching a given scene with a dataset of geo-tagged images and predicting the location of the query image via the matched scenes. The underlying challenge here is to come up with a technique for detecting the exact location of a scene that is robust in the presence of occlusions, illumination, seasonal and structural changes, and yet efficient in dealing with large data (Figure 1).

To this end, we propose an accurate and computationally efficient method that both benefits from global image representations and dense scene matching techniques for fast image retrieval in a city scale setting. The proposed method follows a simple yet effective coarse-to-fine approach. Using the global and local cues exist in the query image, *i.e.* shapes, colors, skylines, landscapes, buildings, *etc.* allows us to make a fairly accurate geolocation estimation through considering the similarity between the query scene and the dataset images at different levels, and accordingly using the geolocation information of the matched scenes.

Figure 1: **Scene Matching.** Sample query image (left) and a set of matched geo-tagged images (right).

In the literature, there are two common approaches to collect geo-tagged image data. One way is to construct a dataset by performing queries in public photo sharing websites such as Flickr as in [3, 8, 9]. In regard to city-scale geolocalization, the downside of this approach is that the collected data is usually disorganized, unevenly distributed and undersampled to represent a city, and might have noisy GPS tags. Other and more structured way is to collect the data by using surveying vehicles such as the Google Street View cars. This approach offers a good representation for this problem in terms of dense and evenly distributed data. However, such data is usually not publicly available on the web. To the best of our knowledge, the only large-scale datasets publicly available are the ones by Chen *et al.* [4] (1.06M images) for downtown San Francisco (Figure 2), and by Google [1] (102K images) for downtown and neighboring areas of Pittsburg, PA, Orlando, FL and partially Manhattan, NY. In this work we evaluated our results using the dataset by Chen *et al.* [4] due to its large coverage (Figure 7).

Contributions. The main contribution of this study is to approach the city-scale geolocation problem from a perspective that rely on a coarse-to-fine strategy so that the suggested approach scales up well to very large datasets. More specifically, we propose to refine the predictions based on GIST [14] and Tiny Image [17] global image descriptors with a dense matching technique called deformable spatial pyramid (DSP) matching [10]. While the former step reduces the search space to a large extent, the latter scene alignment step greatly increases the recall rate. Moreover,

Figure 2: **The downtown San Francisco dataset [4].** 16 sample images from the query set, and 52 sample images from the reference dataset

for each of these steps, we introduce a simple outlier removal procedure to further improve the results.

The remaining parts of this paper is as follows. In Section 2 we briefly review the related work. Section 3 gives the detailed description of the problem. Section 4 presents our proposed approach. In Section 5, we evaluate our method and present the experimental results. In Section 6, we conclude the paper with a summary, some discussions on the proposed method and possible directions for future work.

2. Related Work

Geolocalization from a single image is a well-studied topic in the literature. Robertson and Cipolla [15] proposed the first image based method using 200 images taken in an urban environment. Zhang and Kosecka [20] later developed an approach to retrieve geo-tagged images visually similar to a query image using SIFT [13] features and consequently to find the camera position by triangulating the GPS positions of the retrieved scenes. Schindler *et al*. [16] performed location recognition by using a vocabulary tree and SIFT descriptors on a 20 km. of urban streetside imagery with a 30K GPS-tagged set to model large parts of a city. Hays and Efros [8] aimed at geolocalization by using 6 million GPS-tagged images from the Internet in a world scale setting. Kalogerakis*et al*. [9] addressed the problem of image geolocalization based on human travel priors with time-stamps on image sequences. Gammeter *et al*. [6] focused on an object-level annotation of holiday photos by retrieving similar geo-tagged images. Baatz *et al*. [2] dealt with geolocating images from mountainous areas by using a digital elevation map. Zamir and Shah [19] developed an image localization approach by using a 100K set and storing SIFT features within a tree structure. They then proposed another image gelocalization approach that uses both global and local features in a 102K set [18].

The closest works to our study is the work of Zamir and Shah [19], which deals with single image geolocalization based on Google Street View. Our work differs from their approach in terms of the way the image retrieval is carried out. They [19] use a local approach and match the images via SIFT descriptors, whereas we follow a coarse-to-fine strategy that benefits from both the advantages of global and local features in images. In particular, we propose an efficient scheme with GIST [14] and Tiny Images [17] for initial retrieval of similar scenes and then DSP matching [10] for dense scene alignment. The latter work of Zamir and Shah [18] use global features such as GIST and color histogram, to overcome mismatches due to the use of only local cues in scene matching, and propose a method that combine global and local features which considers multiple reference nearest neighbors as potential matches for scene retrieval. We show that our coarse-to-fine matching scheme based on dense scene alignment with DSP [10] outperforms the SIFT-based scheme suggested in [18].

As our experimental results (Section 5) show, our work is highly competitive in terms of performance and, to the best of our knowledge, the first that efficiently solves the city-scale geolocalization problem at this scale with over 1.06M dataset images. As comparison, the work of Schindler *et al*. [16] uses a 30K set, and the work of Zamir and Shah [18] uses a 102K set.

3. Problem Definition and Challenges

The main goal of this study is to estimate the geolocation information of a single photograph of a scene taken in an urban area by considering only the available visual cues such as color and shape information. In order to geolocalize an image, we match the given scene with a dataset of geo-tagged images. The idea is to make a prediction based on the locations of the similar images in the dataset, with an underlying assumption that these retrieved images are in close vicinity to the query image.

The challenge here is that the retrieved scenes from the dataset might be found visually similar to the query image, but in fact, they could be far from the real location of the query image. Another challenge is that we want the error threshold be within a desired range, *e.g*. we used a

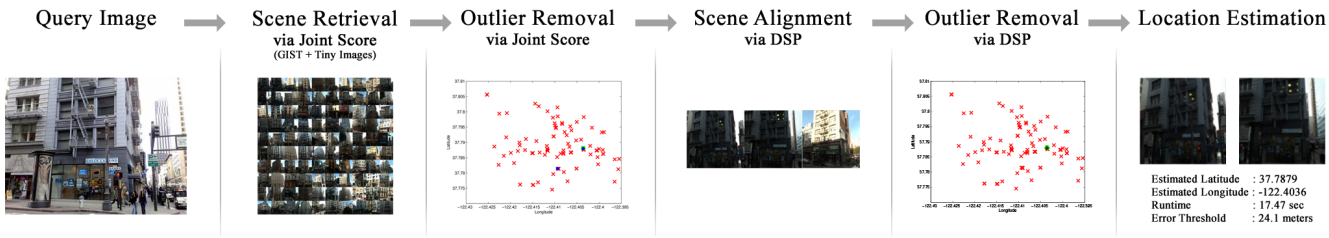

Figure 3: **System Overview.** For a query image, our coarse-to-fine approach first finds the top matches in the dataset using GIST and Tiny Image descriptors, which is followed by an outlier removal procedure. This initial set of images is then refined by densely aligning the retrieved images using DSP matching and accordingly removing the outliers this time by consider the alignment scores.

300 m. error threshold in our experiments. Local cues such as SIFT-like features, are found to be effective in finding a match between two scenes. However, it is extremely hard to obtain near real-time performance when local features are used for searching for visually similar scenes within a very large dataset. Hence, solving this problem in near real-time is also another challenging task. In order to overcome these issues, we developed an algorithm that retrieves and scores the scenes from the dataset in two stages. This efficient coarse-to-fine algorithm ensures that retrieved scenes do accurately match via a final dense scene alignment step. We give the details of our proposed method in the next section.

4. Our Approach

In order to find the location of an image taken in an urban area, we compare the query image with the entire dataset by following a coarse-to-fine strategy. Our motivation here is to find best possible candidates without compromising the query time in a large scale dataset. Figure 3 presents the system overview of our proposed algorithm. Moreover, the algorithmic details are provided in Algorithm 1.

In particular, we undertake the task of image geolocalization in a coarse-to-fine algorithm, which we call Retrieve-Align-and-Predict (RAP), consisting of three different stages described below:

- **Scene retrieval (Section 4.1):** Given a query image, find a set of images that are visually similar in terms of scene context. Remove the outliers with the worst matching scores and the furthest away to the top matched scene (Section 4.3).
- **Scene alignment (Section 4.2):** Refine the initial set of images by densely aligning them with the query image. Then similarly remove the remaining outliers with the worst alignment scores and the furthest away to the top aligned scene.
- **Geolocation prediction (Section 4.4):** Predict the geolocation of the query image by using the locations of the remaining top-aligned scenes.

Algorithm 1 RAP (Retrieve, Align and Predict)

Input: image i, weights w_g, w_t
Phase I: Scene Retrieval
for each item x in reference dataset R **do**
    $S_g(i,x) \leftarrow$ GIST score
    $S_t(i,x) \leftarrow$ Tiny Image score
    $S_{joint}(i,x) \leftarrow w_g.S_g(i,x) + w_t.S_t(i,x)$
Phase II: Scene Alignment
$N \leftarrow$ best candidates in R based on $S_{joint}(i,x)$
$K \leftarrow FNR(N, S_{joint})$
for each item y in K **do**
    $S_{match}(i,y) \leftarrow$ Match score via DSP
$L \leftarrow FNR(K, S_{match})$
Phase III: Geolocation Prediction
$Loc^* \leftarrow predict(L)$
return Loc^*

4.1. Scene Retrieval

Scene retrieval is the key component of our whole method as the final prediction accuracy depends on the quality of the initial retrieval set. To retrieve similar scenes, we first match the reference images in the dataset with the query image by estimating the similarity between them in terms of the popular global image descriptors, namely GIST [14] and Tiny images [17]. If two scenes are found to be similar, we have a high score, otherwise the score is low. We retrieve scenes that look most similar to the query scene based on these similarity scores.

Scene matching in a large-scale dataset is a computationally time consuming task, which is of great importance if near real-time performance is desired. We solved this problem by dramatically reducing the search space via a coarse-to-fine strategy consisting of two stages. In the first coarse matching phase, we compute GIST [14] and Tiny image [17] descriptors for the query image and compute the normalised Euclidean distances between the precomputed descriptors of images in our dataset.

GIST. The GIST descriptor [14] is a global image feature that holds several characteristics about a scene such as the

(a) Query Image (b) 1K (c) 10K (d) 100K (e) 1M

Figure 4: **Size matters.** The number of the images in the dataset is important. As the size increases, the quality of the nearest two matches increases considerably.

strength and the amount of vertical and horizontal lines in a scene. GIST feature help us to understand the similarity between two scenes by looking at the cues such as the buildings, textures, skylines and other objects and have been shown to work very well in terms of retrieving similar scenes [7].

Tiny Images. We additionally used Tiny Images [17] to further improve the recall rates. It is a fast to compute color based descriptor which is simply estimated by the smaller versions of images, such as 32×32 and shown to be effective for object recognition and scene classification especially for larger datasets.

To obtain visually similar images in the dataset we compute a weighted sum of GIST and Tiny Image scores. Combining these two different metrics for the initial retrieval step ensures our method to perform fast in a large scale dataset, which in turn reduces the search space dramatically while yielding fairly good results. It should be noted that the retrieval quality is significantly affected by the dataset size. As illustrated in Figure 4, the quality of the retrieved scenes improves drastically as the dataset size increases.

4.2. Scene Alignment

Relying on global features such as GIST and Tiny Images is a fast way to search against a large scale dataset for overall similarity of scenes but such an approach does not guarantee the exact matches since global descriptors often not robust to occlusion, viewpoint and orientation changes. That is the query image might have a different orientation or viewpoint compared to reference images but a robust method should still be able to find good matches in the presence of such situations. Hence, as a refinement step, we compute dense correspondences between the query image and the retrieved set of similar images.

To align the short list of relevant images with the query image, we propose to use the so-called deformable spatial pyramid (DSP) matching method [10]. The DSP algorithm is a fast and accurate dense matching method, which simultaneously operates at multiple spatial extents, *i.e.* from the entire image to coarse grid cells, up to every single pixel. In this way, we can efficiently discard the qualitatively bad matches, resulting in a more accurate set of nearest neighbor images.

The DSP matching method first defines a spatial pyramid and divide the entire image to four rectangular grids and keep dividing until a desired number of pyramid levels is reached. Finally, it adds an additional level where each cell is of a pixel wide. The approach define spatial nodes of different sizes for each level of pyramid, larger nodes for greater regularization and smaller ones for fine matching. In DSP, each grid cell is described as nodes of a graph. The matching cost is computed for each node by using multiple descriptors. The goal is to find the best translation for each node from first image to second image, which is modelled as the following energy minimization problem:

$$E(t,s) = \sum_{i} D_i(t_i, s_i) + \alpha \sum_{i,j \in N} V_{ij}(t_i, t_j) + \beta \sum_{i,j \in N} W_{ij}(s_i, s_j) \quad (1)$$

where t_i is the translation for node i in first image to second image, D_i is the appearance matching cost of node i at translation t_i, V_{ij} is a smoothness term, W_{ij} denotes a scale smoothness and s_i is a scale variable for node i and α and β are constant weights.

This dense alignment approach has huge advantage in scenarios where the query image and the reference image is taken from different viewpoints or have different field of views.

4.3. Outlier Removal

To make a precise geolocalization, eliminating outliers is as important as finding the best matches in both scene retrieval and alignment steps. The introduction of this stage is to simply filter the unlikely matches that might exist in the candidate lists. Here, we propose a novel two stage outlier removal procedure which we call Furthest Neighbor Removal (FNR). In this algorithm, we first employ similarity based outlier removal by considering visual dissimilarity scores, and then a distance based outlier removal by examining 2D distances computed from the geo-tag information.

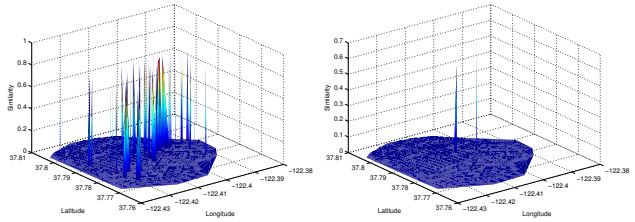

Figure 5: **Similarities.** Joint similarity scores of the initial retrieval list (left) and the matching scores after outlier removal via FNR (right).

Algorithm 2 FNR (Furthest Neighbor Removal)

Input: candidate list C, similarity metric S
Phase I: Removal by dissimilarity
for each item x in candidate list C **do**
  if $S(x) \geq (1+\epsilon) \cdot min(S)$ **then**
    $remove(x)$ from C
Phase II: Removal by 2D distance
for each item x in candidate list C **do**
  $D_i(x) \leftarrow$ average pairwise distance to item i, $i \in C$
  if $D_i(x) \geq \phi$ **then**
    $remove(x)$ from C
return C

We first utilize a ratio test based on the parameter ϵ to select the nearest neighbors in an adaptive way. Mathematically speaking, the neighbors that do not fall within a disk defined by the nearest similarity scores (either the joint scores via global features or the match scores via dense alignment) are removed. This procedure is generally named ϵ-NN set, and returns all the neighbors (1+ ϵ) times the minimum distance from the query:

$$N(x) = y_i \mid dist(x, y_i) \leq (1+\epsilon) \cdot dist(x, y_{min}) \\ y_{min} = min(dist(x, y_i)) \tag{2}$$

We note that this simple inlier selection strategy have proven to be effective for data-driven semantic segmentation [12] and texture synthesis [5, 11].

Following visual similarity based elimination, we employ a 2D spatial distance based elimination. However, we do not simply apply a naive thresholding, but rather compute average pairwise distances for remaining neighbors and use another ratio, ϕ, to eliminate the furthest neighbors adaptively based on their distance ratio to other neighbors.

The details of the FNR procedure are fully described in Algorithm 2. In Figure 5, we demonstrate the effect of applying the outlier removal procedure on the list of relevant images retrieved based on the global descriptors.

4.4. Geolocation Prediction

The prediction phase is quite general. Given a query image, it takes a set of candidates and returns the best possible geolocation by using the 2D position of each candidate and their visual similarity to the input scene. With the best candidates retrieved by our RAP algorithm, we make a triangulation by using a weighted averaging based on the similarity and 2D positions of the candidates as:

$$Loc^* = \sum w_i Loc(x_i) \\ w_i = \frac{S(x, y_i)}{\sum_i S(x, y_i)} \tag{3}$$

where $Loc(x_i)$ denotes the geolocation of the candidate image x_i and the weight w_i refers to the normalized similarity of the query to that image. In Figure 6, we illustrate a sample geolocation prediction performed using Equation 3.

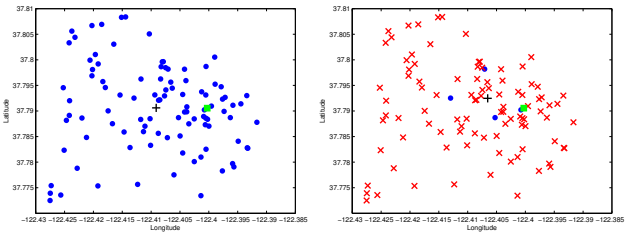

Figure 6: **Geolocation Estimation.** For a sample query image, short list of candidates (left), the inliers found after FNR on the scene alignment scores (right). The blue dots indicate matched scenes. The green square is the groundtruth position of the query scene, whereas the red crosses are the eliminated outliers. The black asterisk indicates the estimated location using the available candidates.

In the experiments, we used a fix set of parameters given in Table 1.

Table 1: Algorithm Parameters

Parameter Name	Parameter Value
Short list size	100
ϕ distance ratio for removing outliers	0.5
ϵ-NN ratio for removing outliers	0.1
Tiny image size	32×32
Tiny image weight w_t	0.1
GIST size	512×512
GIST weight w_g	0.9

5. Experimental Results

In the following, we describe the dataset we used in our experiments, together with the details of training, testing and tuning of our approach.

Dataset. As reference dataset, we used publicly available city scale georeferenced dataset of downtown San Francisco

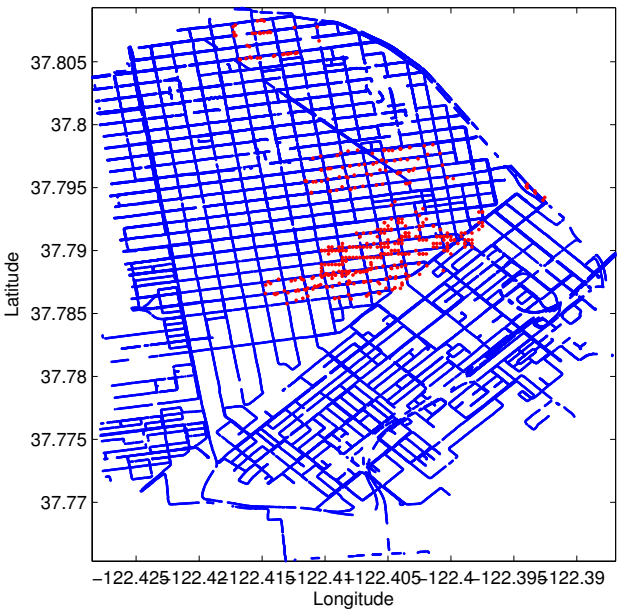

Figure 7: **Database locations.** Reference dataset (blue dots), and query set (red dots)

constructed by Chen *et al*. [4]. There are over 1.06 million perspective images in this dataset. The data was collected by LIDAR sensor and a panoramic camera as 150K panoramic images. Afterwards, panoromic images were converted to perspective images with each image having 60° field of view, 640×480 resolution and 50% overlap with neighboring images [4]. For the experiments, we compute and store the GIST and Tiny Image descriptors of each image in this dataset alongside their GPS coordinates.

Query Set. To simulate a real world geolocalization scenario, we used 596 challenging query images. These GPS tagged images are taken by various mobile phones and provided by [4].

The distribution of the downtown San Francisco dataset and query images are shown in Figure 7 in which the locations of reference and query images are marked with blue and red colors, respectively. Some sample query and reference images are given in Figure 2.

5.1. Evaluation Criteria

We evaluate the effectiveness of our approach in terms of three different criteria, that is accuracy, efficiency and chance. The accuracy is computed by means of the estimation error, the distance between true geolocation of the query image and the predicted one. We consider a geolocalization successful if it is within 300 m. in the vicinity of its true location. Second, we analyze the performance of our method in terms of running times. Third, we compare our results against the random selection of a geolocation from the data set that we refer to as chance. As a further

Figure 8: **Scene Retrieval.** Query images (left) and retrieved images from the dataset (right).

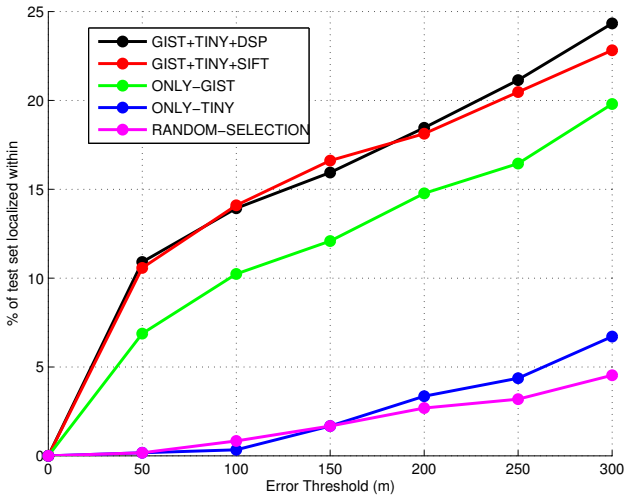

Figure 9: **Recall accuracy.** For various thresholds.

benchmark, we compared our results with the predictions via GIST and Tiny Image descriptor. Additionally, we analyze the effectiveness of the DSP approach [10] against a SIFT based local matching. The latter can be seen as a two step naïve implementation of Zamir and Shah's work [18].

Qualitative Results. Figure 8 shows some sample query images along with the dataset images which are found relevant by the proposed approach. As can be seen, the retrieval set includes images of the scenes that either depicts the query scene or visually very close ones.

Quantitative Results. In Figure 9, we plot the performances of various approaches at different recall levels. Our results indicate that the proposed algorithm can geolocalize 24% of query set within 300 m. and our method performs

11 times better than chance. All instances of query images are succesfully geolocalized within 3.9 km. Moreover, our suggested scheme (GIST + TINY + DSP) outperforms other schemes in recall rates for 300 m. threshold.

Runtime Performance. We implemented the proposed method and algorithms in MATLAB and performed our experiments on a Linux based Intel(R) Xeon(R) 2.50GHz computer on 12 cores. In terms of computational performance our method geolocalizes a scene under 160 seconds on average. However, the SIFT-based implementation performs better in terms of computational costs, it can geolocalize a scene under 135 seconds on average, with a slightly worse performance.

6. Conclusion

In this paper, we described a fast and robust method for city scale scene geolocalization. We also demonstrated the performance and the quality of proposed method by using various evaluation criteria. Our method combines global image descriptors with a dense scene alignment strategy, and successfully geolocalizes challenging query scenes taken in urban areas.

There are some challenges that have to be addressed in order to get more reliable gelocalization results from our method. Our approach is based on matching geo-tagged scenes with query images taken from street level urban images with unknown camera parameters. We assume that at least one reference image is taken as that of the similar vantage point as the query image. Therefore, there is no possibility to match query images from unsampled locations. This limitation leads to poor results when there is no matched samples in the reference dataset from the query image location, *c.f*. Figure 4 for size limitations.

Acknowledgments

We would like to thank to the following researchers, Kim *et al*. [10] for making DSP code, and Chen *et al*. [4] for making their dataset publicly available.

References

[1] ICMLA 2011 StreetView Recognition Challenge. http://www.icmla-conference.org/icmla11/challenge.htm. 2011. 1

[2] G. Baatz, O. Saurer, K. Köser, and M. Pollefeys. Large scale visual geo-localization of images in mountainous terrain. In *Computer Vision–ECCV 2012*, pages 517–530. Springer, 2012. 2

[3] C.-Y. Chen and K. Grauman. Clues from the beaten path: Location estimation with bursty sequences of tourist photos. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1569–1576. IEEE, 2011. 1

[4] D. M. Chen, G. Baatz, K. Koser, S. S. Tsai, R. Vedantham, T. Pylvanainen, K. Roimela, X. Chen, J. Bach, M. Pollefeys, et al. City-scale landmark identification on mobile devices. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 737–744. IEEE, 2011. 1, 2, 6, 7

[5] A. A. Efros and T. K. Leung. Texture synthesis by non-parametric sampling. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, volume 2, pages 1033–1038. IEEE, 1999. 5

[6] S. Gammeter, L. Bossard, T. Quack, and L. V. Gool. I know what you did last summer: object-level auto-annotation of holiday snaps. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 614–621. IEEE, 2009. 2

[7] J. Hays and A. A. Efros. Scene completion using millions of photographs. In *ACM Transactions on Graphics (TOG)*, volume 26, page 4. ACM, 2007. 4

[8] J. Hays and A. A. Efros. Im2gps: estimating geographic information from a single image. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008. 1, 2

[9] E. Kalogerakis, O. Vesselova, J. Hays, A. A. Efros, and A. Hertzmann. Image sequence geolocation with human travel priors. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 253–260. IEEE, 2009. 1, 2

[10] J. Kim, C. Liu, F. Sha, and K. Grauman. Deformable spatial pyramid matching for fast dense correspondences. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 2307–2314. IEEE, 2013. 1, 2, 4, 6, 7

[11] L. Liang, C. Liu, Y.-Q. Xu, B. Guo, and H.-Y. Shum. Real-time texture synthesis by patch-based sampling. *ACM Transactions on Graphics (ToG)*, 20(3):127–150, 2001. 5

[12] C. Liu, J. Yuen, and A. Torralba. Nonparametric scene parsing via label transfer. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(12):2368–2382, 2011. 5

[13] D. G. Lowe. Object recognition from local scale-invariant features. In *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, volume 2, pages 1150–1157. Ieee, 1999. 2

[14] A. Oliva and A. Torralba. Building the gist of a scene: The role of global image features in recognition. *Progress in brain research*, 155:23–36, 2006. 1, 2, 3

[15] D. P. Robertson and R. Cipolla. An image-based system for urban navigation. In *BMVC*, pages 1–10, 2004. 2

[16] G. Schindler, M. Brown, and R. Szeliski. City-scale location recognition. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–7. IEEE, 2007. 2

[17] A. Torralba, R. Fergus, and W. T. Freeman. Tiny images. 2007. 1, 2, 3, 4

[18] A. Zamir and M. Shah. Image geo-localization based on multiple nearest neighbor feature matching using generalized graphs. 2014. 2, 6

[19] A. R. Zamir and M. Shah. Accurate image localization based on google maps street view. In *Computer Vision–ECCV 2010*, pages 255–268. Springer, 2010. 2

[20] W. Zhang and J. Kosecka. Image based localization in urban environments. In *3D Data Processing, Visualization, and Transmission, Third International Symposium on*, pages 33–40. IEEE, 2006. 2