Efficient Global 2D-3D Matching for Camera Localization in a Large-Scale 3D Map

Liu Liu [1,2], Hongdong Li[2,3] and Yuchao Dai[1,2]
[1] Northwestern Polytechnical University, Xi'an, China
[2] Australian National University, Canberra, Australia
[3] Australia Centre for Robotic Vision
nwpuliuliu@mail.nwpu.edu.cn, {hongdong.li, yuchao.dai}@anu.edu.au

Abstract

Given an image of a street scene in a city, this paper develops a new method that can quickly and precisely *pinpoint at which location (as well as viewing direction) the image was taken, against a pre-stored large-scale 3D point-cloud map of the city. We adopt the recently developed* 2D-3D direct feature matching framework *for this task [23, 31, 32, 42–44]. This is a challenging task especially for large-scale problems. As the map size grows bigger, many 3D points in the wider geographical area can be visually very similar–or even identical–causing severe ambiguities in 2D-3D feature matching. The key is to quickly and unambiguously find the correct matches between a query image and the large 3D map. Existing methods solve this problem mainly via comparing individual features' visual similarities in a* local *and* per feature *manner, thus only local solutions can be found, inadequate for large-scale applications.*

In this paper, we introduce a global method *which harnesses global contextual information exhibited both within the query image and among all the 3D points in the map. This is achieved by a novel global ranking algorithm, applied to a Markov network built upon the 3D map, which takes account of not only visual similarities between individual 2D-3D matches, but also their global compatibilities (as measured by co-visibility) among all matching pairs found in the scene. Tests on standard benchmark datasets show that our method achieved both higher precision and comparable recall, compared with the state-of-the-art.*

1. Introduction

Getting accurate estimation of 6-DoF camera pose from an image is essential for many computer vision applications such as robot navigation [15, 37], augmented reality [35, 55], and image-based 3D reconstruction [4,18]. While more and more consumer-grade cameras are equipped with in-built GPS sensors which can provide some rough location estimation, the accuracy is rather coarse (at tens of meters [12,57]) and is inadequate for many critical applications.

This paper proposes a new method for image-based camera localization (or IBL in short), against a pre-computed 3D point-cloud map. Our method follows the recently proposed framework of *direct 2D-3D matching* [23,31,32,42–44]. Under this framework, camera pose is computed by directly matching 2D image features (*e.g*. SIFT [34]) from the query image to 3D points in the map, then solve a standard camera *absolute pose* problem via PnP (perspective-n-points). If the 2D-3D matches found are contaminated by some small portion of outliers (*i.e*. wrong matches), RANSAC is conventionally applied to clean up the matches. However, this "PnP+RANSAC" scheme only works for small or moderately large problems. When the 3D map is very large, for example, covering a wide geographical area of an entire city or even a country, there may have tens of thousands or millions of 3D map points, which poses two major challenges to the problem: (1) how to quickly search (match) within a massive database of millions of 3D points; and (2) how to accurately find correct matches without suffering from ambiguity. The latter is more critical because, as the 3D map grows larger, more and more 3D features (*e.g*. SIFT) can become visually very similar or even identical due to repetitive structures. As such, one is facing an extremely difficult task of "*finding a particular needle in a huge haystack containing many other similarly-looking needles*". Applying RANSAC to this situation is doomed to fail, because the inlier ratio in the putative matches can be as low as *e.g*. ≤ 0.01 [51,52].

To solve this scalability issue, existing direct 2D-3D methods often adopt advanced retrieval techniques, such as "vocabulary tree" and "ratio test" to remove ambiguous matches [40, 42–44, 47]. However, they do this largely in a local, sequential fashion on individual *per feature* basis. In their methods, the *match-or-not* decision is only made

2380-7504/17 $31.00 © 2017 IEEE
DOI 10.1109/ICCV.2017.260

locally based on comparing individual feature match's visual similarity. When the 3D map is very large, it may have many repeated structures, spurious or ambiguous matches are almost inevitable. As a result, the matches found by a local method may be overwhelmed by outliers, leading to wrong localization.

In this paper, we introduce a principled new method which finds optimal 2D-3D matches *in a global manner*. Contrast to existing methods which rely on *local (per feature basis)* visual similarity comparison, we advocate a global scheme which exploits global contextual information exhibited not only within 2D features from the query image, but also among all matched 3D points in the map. More specifically, our new method no longer treats each individual 2D-3D match in isolation, but takes account of the compatibilities (or coherencies) among all 2D-3D matches. To measure such compatibility, we do not consider 3D points in the map as unordered "clouds of points". Instead, every 3D point joins with other neighboring 3D points via *co-visibility* relationship. A precise definition of the co-visibility relation and how to use it for global matching will be described in detail later in the paper.

Tested on standard benchmark datasets for 2D-3D IBL, our new method shows superior performance, outperforming state-of-the-art methods in many aspects. Compared with [43,44], it halves the median localization error (higher precision), while maintaining a comparable level of recall. More importantly, our method is a principled global approach, thus allows for versatile extensions.

2. Related Work

In this section, we give a brief review of previous papers closely related to our new method. We focus on the task of localizing a single image against a large-scale 3D map, hence omitting a large body of works on video-based camera localization such as that for visual-SLAM (*e.g.*, FAB-map [14]). For space reason, we also leave out those (machine) learning-based methods (*e.g.* regression forest [49], deep PoseNet [26, 27]). Interested readers are referred to these literatures for more details.

Image-retrieval based methods Instead of building large-scale 3D map using Structure-from-Motion (SfM) methods [4,18,48], the image-retrieval based methods (*e.g.*, [6, 8, 11, 19, 25, 28]) try to identify the similar databased images depicting the same landmarks as the query image. Often, the retrieved database images are ranked subject to some similarity metrics (*e.g.*, L_1 norm distances between Bag-of-Words vectors [19], L_2 norm distances between compact representations/vectors [6, 7]), and the position of the best database image is deemed as that of the query image or the top N images are fused to get the position of the query image [46,47,50].

Direct 2D-3D matching. Irschara *et al.* [23] first proposed the 2D-3D matching method for camera localization. To overcome limited viewpoints in the database images, they artificially synthesized novel view image to augment the database. Alcantarilla *et al.* [5] learned a similarity metric between images based on poses to predict which 3D points are visible at the current camera pose. By assuming known gravity direction, Svarm *et al.* [51, 52] and Zeisl *et al.* [58] developed methods to handle outliers. Sattler *et al*. [40] performed matching via a fine-grained vocabulary search. Feng *et al*. [17] proposed to use binary feature descriptors to speed up the search. All these methods are *local methods* in the sense that they seek to find one-to-one feature matches based on local similarity comparison; they seldom exploit global information, hence can only find a local, hence sub-optimal solution. To distinguish true matches from spurious matches (*i.e*. outliers), they adopted Lowe's *ratio-test* [34], yet the results are not satisfactory for large-scale maps ([16]).

Co-visibility. The idea of using co-visibility (co-occurrence) for IBL is not brand new. It has been adopted by several previous works (*e.g*. [13, 31, 32, 45]), though in a local heuristic manner (for example, to improve local search efficiency via *query expansion*, to prioritize candidate matches, or to filter out false 3D points via geometric validation [40,43,44]). Since their processes are often performed at individual match level, they often need a good initialization [13,31,32]. In comparison, our method performs global ranking of 2D-3D matches based on global information, without the need to re-compute priorities of 3D points. Moreover, we do not reject promising matches pre-maturely.

3. An Overview of the Proposed Method

Contrast to previous works, in this paper we propose a global method which exploits *global contextual information* to resolve the matching ambiguity. Specifically, we harness two types of global contextual information. For one thing, instead of focusing on matching each individual 2D feature, we treat the entire set of features in the query image jointly. For the second, we no longer consider each possible 2D-3D match in isolation, but consider all tentative 2D-3D matches together. We obtain set-to-set matches instead of finding one-to-one matches in the first place and defer the disambiguation task until a later stage of the computation. Figure-1 gives an overall pipeline of our method. Figure-2 illustrates the conceptual difference between traditional local methods and our new global method.

To define the global contextual information among 3D points, we use *co-visibility* relationship. The central mechanism of our method is a probabilistic inference procedure applied to a Markov graph, built upon the 3D map points as

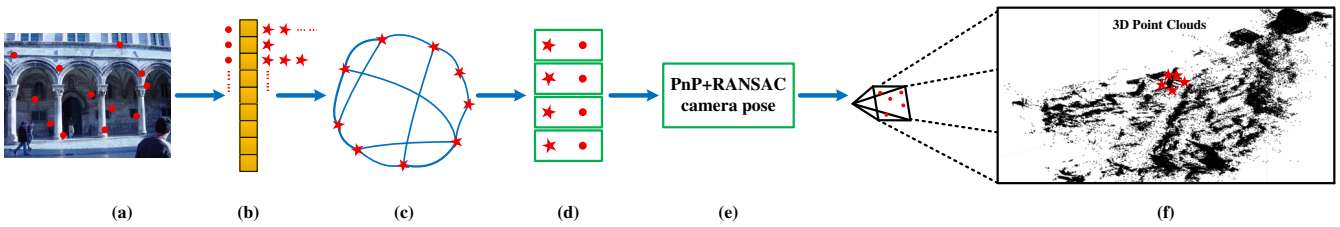


Figure 1. We solve a large-scale image-based localization problem by leveraging global contextual information manifested as co-visibility relationship between pairs of 3D map points. (a) Image features extracted from the query image; (b) Assign 2D features to visual words to obtain candidate 3D matches; (c) The matches are ranked based on global contextual information; (d) One-to-one 2D-3D matches are disambiguated; (e) PnP+RANSAC is used for 6-DoF camera pose recovery against the 3D map (f).

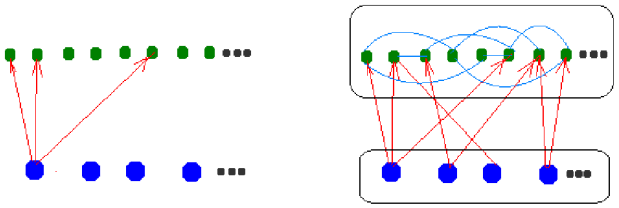


Figure 2. Left: traditional local method, in which the decision is made locally and sequentially; Right: the proposed global match scheme, where we seek optimal set-to-set match. Blue nodes: 2D features in the query image. Green node: 3D points in the 3D map. Blue links indicate co-visibility relationship among 3D points.

graph-nodes, connected by graph-edges representing inter-point co-visibility relationships. The inference is done globally, taking account of all 2D query features, all 3D map points, as well as the co-visibility encoded as graph-edges.

To solve the inference task on a Markov graph, we resort to the Random Walk with Restart (RWR) algorithm [53]. Google's PageRank [9] algorithm is in fact a well-known variant of RWR algorithm [53]. The probability distribution of graph nodes in a Markov graph evolves in a stochastic manner via random walk. When it converges, the stable state of a node measures the "relatedness" (*i.e.* "matchability") between the node and the set of query features. Recall that one of the key innovations of this paper is to seek global optimal (set-to-set) matches, rather than local one-to-one matches.

4. The Proposed Method

We go directly to explain key steps of our new method.

4.1. Step-1: Build a Map-Graph with pairwise co-visibility edges

Traditionally, a 3D map is often encoded as a set of unordered point clouds [42–44]. In this work, we aim to bring order to the clouds, by connecting (organizing) all 3D points in the map in a well-structured **map-graph**, and denote it as G(V,E), with V indicating the set of graph-nodes, each corresponds to a 3D point (as well as its associated descriptors (visual words)); and E the set of graph edges. A pair of 3D points are connected with a graph edge if and only if they can be seen simultaneously from the same viewing point. Like many other works, we assume our 3D map was pre-computed via Structure-from-Motion [4, 18, 48] technique using a large set of database images. Therefore, the co-visibility relationship among 3D points can be obtained using the database images.

We require G(V,E) to be *weighted* and *bi-directed*. Thus, for every pair of co-visible 3D points in the graph (i and j), there are two edges (e_{ij} and e_{ji}) connecting them, with non-negatively-valued weights of c_{ij} and c_{ji}. The weights measure how strong the co-visibility relationships between the two points are, *as per* the following sense: *If point-j is seen by an image set, the value of c_{ij} measures how likely the point-i can also be seen from the same image set.* c_{ji} can be defined conversely, and $c_{ji} \neq c_{ij}$ in general.

Formally, suppose there are N nodes and M edges in the graph. We devise the following procedure to compute c_{ij}, using database images used in the map-construction stage. For the i-th 3D point, denote the set of database images that contain this point as A_i. If two distinct 3D points i and j are co-visible, they will cast "support" or "endorsement" to each other, and the strength of the "endorsement" from point-j to point-i (*i.e.* c_{ij}) is defined as:

$$c_{ij} = \frac{|A_i \cap A_j|}{|A_j|}, \tag{1}$$

where $|A_j|$ is the cardinality of set A_j, and $A_i \cap A_j$ denotes the set-intersection operator. This equation can be understood as follows. Since point-j is known to be visible, the probability that point-i and point-j are co-visible is proportional to the total number of database images that contain both points, normalized by the total number of images that contain point-j. Conceptually this is similar to the idea of *tf-idf* (term-frequency/inverse document frequency) as commonly used in information retrieval [2].

Collecting all c_{ij}s into a square matrix $\mathrm{C} = [c_{ij}]$ of size $N \times N$, and normalizing each column to have unit norm,

we are able to represent the entire graph G by its C matrix. The reason of normalization is to make C a left stochastic matrix [1], and every c_{ij} can be interpreted as a probability. We call C the **state (probability) transition matrix**, for reasons that will be clear in the rest of the paper. Note also C is often a sparse matrix especially for a large graph. Note that C is based on database images and the 3D map only, hence it is *query-independent* and can be pre-computed off-line.

The top part of the graph in Figure-3 illustrates a toy-sized map-graph with 15 nodes (*i.e.* nodes colored in green) and some bi-directed co-visibility edges (*i.e.* links colored in blue).

4.2. Step-2: Compute query vector q

Given a query image, we first detect a set of 2D feature points, along with their view-invariant descriptors (*e.g.* SIFT [34]). Next, for every 2D feature we find a set of tentative matches from the 3D graph nodes, by comparing their descriptor similarity via an efficient *vocabulary-tree* search mechanism [36]. Instead of seeking a one-to-one 2D-3D matches, here we only look for one-to-many matches; the reason is to avoid local matches (which may be pre-mature) by deferring the one-to-one disambiguation process until a later stage.

Vocabulary-tree search. We assign all the 3D points to a pre-trained Bag-of-Words vocabulary-tree using Voronoi vector quantization [38]. We use the same *integer-mean* assignment method suggested by [42–44] to obtain 2D-3D matches. Note that one 3D point may be assigned to multiple visual words, and conversely one visual word may correspond to multiple 3D points (ref. [43, 44]). However, for each 2D feature from the query image, we only assign one visual word to it for efficiency.

Query vector. We use the *Hamming embedding distance* $H(f,i)$ to measure the similarity between a 2D query feature f and a 3D map point i. For brevity we refer the reader to [24] for a precise definition of Hamming embedding distance. Next, inspired by [8, 24, 41, 47], we define the similarity between 2D feature-f and 3D point-i as $w_{fi} = \exp(-H^2(f,i))/\sigma^2, \forall i \in [1..N]$, where σ is typically set to $1/4$ of the dimension of the descriptor according to [8]. Note that the similarity is computed at per visual word basis.

By summing up all the similarities from the entire set of query features at every 3D point, and stacking the results into a single vector, we obtain a vector $\mathbf{q} \in \mathbb{R}^N$ whose i-th element is:

$$q_i = \sum_{f \in \mathcal{O}(i)} \frac{\sqrt{w_{fi}}}{N_i} \cdot \log\left(\frac{N}{N_f}\right), \qquad (2)$$

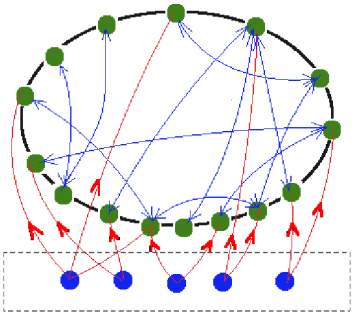


Figure 3. An illustration of a toy-sized map-graph G(V,E). Green nodes are the 3D point clouds in the map. Blue edges are co-visibility links. The blue nodes on the bottom represent 2D query features which assign initial probabilities to the 3D points based on the query-vector $\mathbf{q}$ computed by Eq.2.

where $\mathcal{O}(i)$ is the set of 2D query features which are (tentatively) matched to point-i; N_i is the size of $\mathcal{O}(i)$, N is the total number of 3D points in the map, and N_f is the number of 3D points which are tentatively matched to feature-f. Once $\mathbf{q}$ is obtained, we normalize it to have unit norm, *i.e.*, $q_i \leftarrow \left(q_i / \sum_{i=1}^{N} q_i\right), \forall i \in [1..N]$. We call such a (normalized) $\mathbf{q}$ the **query vector.** Intuitively, the i-th entry of a query vector (*i.e.*, q_i), measures the probability of point-i belongs to the optimal sub-set of 3D points that can be matched to the set of 2D query features – based on their visual word similarity only.

A remark. Existing methods for direct 2D-3D matching are primarily built upon the comparison of local 2D-3D feature similarity (*e.g.*, perform 2D-3D ratio-test at per visual word basis [42–44]); they fail to capture global information among all the matches.

4.3. Step-3: Random walk on map-graph

Given a map-graph G(V,E) along with a state transition matrix C (*Sec.* 4.1). we formulate it as a Markov Network (aka. Markov Random Field). Suppose we are present with a query image, we first compute its query vector $\mathbf{q}$ with respect to the graph G (*Sec.* 4.2).

Our idea to seek a global match between 2D query image and 3D map is to run a **Random Walk** algorithm on this graph, *conditioned on the input query vector* $\mathbf{q}$. When the random walks converge, we then deem that the steady-state probability obtained at each 3D node on the graph actually measures how well it is matched (or matchable) to the query image. The higher a node's steady-state probability is, the more probable that it belongs to the correct 3D point set.

In essence, a random walk algorithm simulates a randomly-moving walker traversing through the graph. At every time tick, the walker moves to a randomly chosen neighboring node based on the probability stored in matrix C. Probability $p_v(t)$ is defined as the probability of finding

the random walker at node v at time t. Therefore, when t goes to infinity, $p_v(\infty)$ gives the probability that the random walker eventually ends at node v.

In the 2D-3D matching context, our idea is to capture both local feature appearance similarity information and global points co-visibility information. For this purpose we require the random walk to respect both visual-word similarity (prescribed by the query vector $\mathbf{q}$) and global co-visibility structure (provided by the transition matrix C). We design the following Random Walk with Restart (RWR) iterative procedure, where the random walker has a tendency (though at a small probability) to return to the set of *starting nodes* prescribed by the query-vector $\mathbf{q}$.

Random Walk with Restart:

$$\mathbf{p}(\mathbf{t}+\mathbf{1}) = \alpha \mathrm{C}\mathbf{p}(\mathbf{t}) + (1-\alpha)\mathbf{q}. \tag{3}$$

Here $\mathbf{p}(\mathbf{t}) \in \mathbb{R}^N$ is the graph's state probability vector at time t. α is a trade-off parameter, often chosen empirically between 0.8–0.9. The second term in the right hand of the iteration equation is known as the *restarting term*, which gives the random walker a small probability of returning to the states prescribed by the query vector. For readers who are familiar with MRF, we can say that: the first term of the right-hand side of the equation is basically the "prior" or "smoothness" term which describes how the network behaves if no external query signal is presented. The second term is the "data term" which encourages the result to respect the input query signal.

To start the iteration, we initially concentrate all the probability mass uniformly over all 2D query features (*e.g.* in Figure-3, the 2D query features are colored in blue), *i.e.*, all the 2D query features have the same probabilities to be matched to 3D points. We then connect these query features to the 3D graph nodes by one-way directed edges (Vocabulary-tree search), and inject probability mass to the graph based on the probabilities stored in $\mathbf{q}$, *i.e.*, the original probabilities of 3D points are initialized by $\mathbf{q}$. Once the iteration converges, we sort this steady-state probability vector $\mathbf{p}(\infty)$ in descending order, which gives the final "matchability" of every 3D point to the set of 2D query features.

Remarks. As proved in [22], convergence of the above iteration is guaranteed when C is *aperiodic and irreducible*. In our particular map-graph, both conditions are satisfied, because aperiodicity is true since the state transition probabilities in Eq.(1) are different for distinct pair of 3D points, and the irreducibility is true since our graph is (two-way) bi-directed connected. There are no so-called *dangling nodes* as all 3D map points were computed from SfM triangulation from two or more views, therefore they cannot exist alone without co-visible neighbors. Also, the above iteration is intimately related to Google's PageRank [20]. This is not surprising, because the task that we are solving in this paper is a typical information retrieval (IR) task, and PageRank is a well-known IR tool efficient in solving large-scale IR problems. However, despite this, to the best of our knowledge, random-walk has not been applied to Camera Localization. Moreover, there are important differences between our method and PageRank, which make our method particularly relevances for IBL: (1) We use bi-directed graph with two-way weights to capture co-visibility neighborhood relations, in contrast to Google's undirected "Web graph" with binary (1/0) hyperlink neighbors. (2) We do not use Google's uniform teleportation vector, and replace it with a query vector. In spirit ours is akin to a personalized version of PageRank [21]. (3) Our state transition matrix and similarity query vector have taken into account of the special structure of the direct 2D-3D method.

4.4. Step-4: Camera pose computation

Recover one-to-one correspondences. The steps so far have only achieved set-to-set global matching. To facilitate camera pose computation, ultimately we still need one-to-one matches. Since after our previous random walk algorithm, positive 3D points will likely be ranked highly, making it amenable to a simple ratio-test [34] to resolve the one-to-many ambiguity. Other more sophisticated matching methods (such as Hungarian assignment [29]) are also applicable. We do not insist to find perfect putative one-to-one matches at this point, because the matches will be fed into the subsequent PnP-RANSAC for further outlier removal. We simply use the ratio-test to retrieve one-to-one matches. The ratio-test is performed by comparing the descriptor distances between the 3D points (one-by-one in the ranking list after random walk iterations) and 2D feature points when they are at the same visual words in the vocabulary tree. The one-to-one match is accepted when it passes the ratio test.

RANSAC camera pose. The obtained one-to-one correspondences are fed directly to a RANSAC absolute pose routine. We use the P4P approach [10, 33] to solve the unknown focal length, camera position and orientation.

5. Experiments

Benchmark datasets. We conducted extensive experiments to validate the efficacy of the proposed global method. We evaluate its performance against four standard publicly available benchmark datasets for city-scale localization ([12,43]): (1) Dubrovnik, (2) Rome, (3) Vienna and (4) San Francisco (SF-0), where the first 3 have about millions of 3D map points, but the last one is much bigger in size (*e.g.*, by 1 or 2 orders of magnitude larger in terms of total number of 3D points or database images). Information about the 4 datasets is summarized in Table-1.

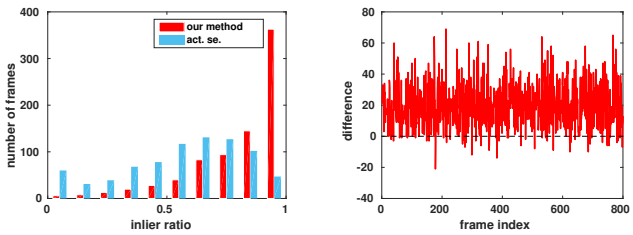
Table 1. Statistics of the benchmark datasets: the numbers of database images, 3D points and query images.

Dataset	#(images)	#(points)	#(query images)
Dubrovnik [32]	6,044	1,975,263	800
Rome [32]	15,179	4,067,119	1,000
Vienna [23]	1,324	1,123,028	266
SF-0 [12]	610,773	30,342,328	803

Experiment setting. To evaluate the algorithm performance, we mainly use (a) recall-rate (*i.e.* how many images have been successfully localized), (b) precision (*i.e.* camera localization errors), (c) accuracy (*i.e.* what is the inlier ratio in the final matched 2D-3D feature pairs after applying RANSAC), as well as (d) scalability (*i.e.* by testing on the largest dataset of SF-0 containing over 30 Millions map points). All our experiments were conducted on a single CPU thread based on a regular laptop with Intel i7-6700K CPU at 4GHz. Note for localization precision comparison, we only report results on Dubrovnik dataset, because it is the only dataset among the four which has metric ground-truth information to 6 DoF camera locations. For SF-0 we only found rough estimations of camera positions given out by the GPS-tag of each query image. In implementing the visual-vocabulary-tree search, we use a pre-trained vocabulary of 100K visual words [42], generated with approximate k-means clustering [39] of SIFT descriptors, and choose a tree branching factor at 10. We perform the 3D-2D ratio-test at level 3 to recover the one-to-one correspondences, the ratio-test threshold used to reject outlier matches is set to 0.6, which is the same as used by Active Search [43, 44]. We used P4P [10, 33] and RANSAC. In RANSAC, the re-projection error for rejecting outliers was set to 4 pixels, and the belief level at 0.99 and the expected inlier-threshold at 0.4. We used a damping parameter of $\alpha = 0.85$ and stop the algorithm after 10 iterations (enough for converging, *i.e.*, the permutation of 3D points are fixed after the iterations).

5.1. Is global search really effective?

In our first set of experiments, we want to verify (and to evaluate) whether or not the use of global contextual information (as defined by the Markov graph using pair-wise 3D point co-visibilities) is effective. For this purpose, we compare our method with the Active Search method [43, 44]– which is considered as the state-of-the-art local search methods. In other words, it is expected that results obtained by Active-Search represent what the best-performing local methods should achieve. We conducted experiments on the metric version of Dubrovnik dataset with sub-maps with reduced sizes of up to $40,000$ map points. The sub-map for each query image is generated by including the 3D points observed by its nearby database images, using image-retrieval techniques [46] or GPS data of the query/database image if available [3, 12].



(a) (b)

Figure 4. (**a**). Compare the two histograms of inlier ratios for the 800 query images of Dubrovnik. Red: histogram by our method; Light-blue: histogram by Active-Search. The average inlier-ratio obtained is 81.1%, and 57.1%, by our method and by Active-Search, respectively. (**b**). The absolute improvement in terms of inlier numbers (=#(inliers found by our method)-#(inliers found by Active-Search)) over all query images from Dubrovnik. A positive-valued 'difference' means more inliers are detected by our method. Our method consistently outperforms the local Active-Search method for almost all 800 queries.

We use the final inliers set reported by PnP+RANSAC as the found inlier matches. We keep all parameters for the RANSAC process the same for both our method and Active Search for the sake of fair comparison. After running both algorithms, we compare the histograms (distributions) of the obtained inlier ratios. The higher the inlier ratio is, the better the method. Figure-4 (a) gives the distributions of inlier ratios for the two methods. From this, one can clearly see that our global method statistically outperforms Active-Search. To evaluate whether the improvement is consistent across all query images, we plot the differences between the numbers of correctly-detected inliers (out of the top 100 ranked candidate matches), one was obtained by our method and one by the Active-Search method. Figure-4 (b) shows this result. Again, except for a few exceptions, our global method outperforms the Active Search method consistently for almost all 800 query images.

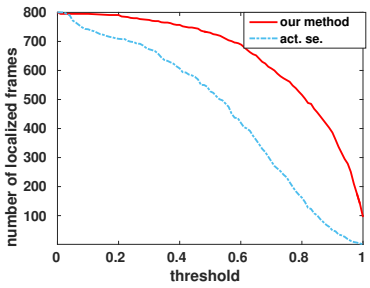


Figure 5. Recall curve: *i.e.*, the number of localized images as a function of inlier ratio threshold. The higher, the better. (see text for more details).

Recall Curve. To evaluate the recall performance (*i.e.* numbers of query images that can be successfully localized un-

Table 2. Numbers of localized images v/s inlier ratio thresholds.

Method	Inlier thresholds						
	0.2	0.3	0.4	0.5	0.6	0.7	0.8
Active Search [43, 44]	709	673	607	528	420	287	162
Our method	**791**	**774**	**757**	**730**	**690**	**607**	**516**

der different inlier-ratio-thresholds), we vary the inlier-ratio thresholds between [0–1]. We then plot the obtained two recall curves (one by our global method, one by the local Active Search method). As shown in Figure-5, our method has consistently localized more images at all threshold levels. A detailed numerical comparison for the two methods under different inlier-ratio thresholds is given in Table-2:

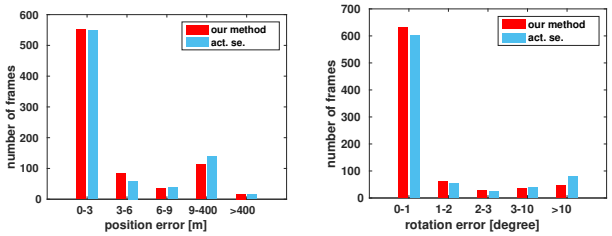


Figure 6. Localization Precision on the Dubrovnik dataset. Left: translation error histogram; Right: rotation error histogram. Results by our method in red, and by Active-Search in blue.

Localization Precision. We also compare the 6-DoF camera pose precisions obtained by the two methods. Rotation error is measured by $\epsilon = \arccos((\text{trace}(\mathbf{R}_{gt}^T \mathbf{R}_{es}) - 1)/2)$, where $\mathbf{R}_{gt}$ is the ground-truth rotation, and $\mathbf{R}_{es}$ the estimated one. Translation error is measured by the absolute distance between the estimate and the ground-truth camera center. Figure-6 gives the histogram over the position errors and rotation errors. Our method outperforms the Active Search method at almost all position/rotation levels, localizing more frames while maintaining the accuracy.

Table 3. Localization errors on metric Dubrovnik.

Method	quartile errors (m)			num. of images		
	1st	median	3rd	<18.3m	>400m	#(reg.)
our method	**0.24**	**0.70**	**2.67**	**743**	**7**	794
act. se. [43, 44]	0.40	1.40	5.30	704	9	**795**
all desc. [42]	0.40	1.40	5.90	685	16	783
int. mean [42]	0.50	1.30	5.10	675	13	782
P2F [32]	7.50	9.30	13.40	655	-	753
vis. prob. [13]	0.88	3.10	11.83	-	-	788

5.2. Comparisons with other State-of-the-Art

In this section, we compare our method with several other state-of-the-art local methods. Following [31, 32, 43, 44], we deem an image is localized if the best-pose found by RANSAC contains ≥ 12 inlier matches. We first compare the localization precision and the results are presented in Table-3 and Figure-7. Our method achieves the best localization accuracy at all three quartile levels of location errors, which almost halves the localization errors obtained

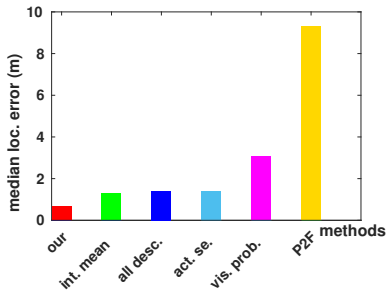


Figure 7. Compared with five other existing methods, our method achieves the best localization precision while maintaining a high recall on the metric Dubrovnik dataset. The lower the error bar is, the better the method.

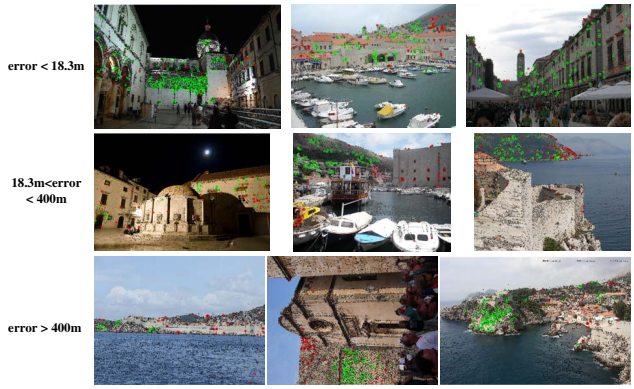


Figure 8. Sample results on metric Dubrovnik data. The detected inlier points are shown as circles in green, and outlier features in red, against different localization errors. SIFT feature points are depicted as black dots. For images with error $< 18.3m$, the inliers are evenly distributed over the image. For images with $18.3m <$ error $< 400m$, the inlier features tend to concentrate on a small region of the image.

by other methods. It localizes 743 (out of 800) query images under a localization error of $< 18.3m$. The average query time is 1.73s.

Examples of detected inlier and outlier features for some sample query images are shown in Figure-8. A point-cloud view of the estimated camera locations is given in Figure-9. Our method outperforms a very recent pose-voting method [58] in terms of location precision (pure voting with median error at 1.69m), despite [58] used IMU information (for vertical direction determination). We did not compare method in [51] (which also exploited known vertical direction information) because their results were obtained on synthetic data only.

We also experimented on the Rome and Vienna datasets. They each has about 4-millions and 1-millions 3D map points, respectively. Our method has localized 990 (out of 1000) and 213 (out of 266) query images for these two datasets, respectively. The average query time by our unoptimized code was 2.35s and 1.67s, which while slower

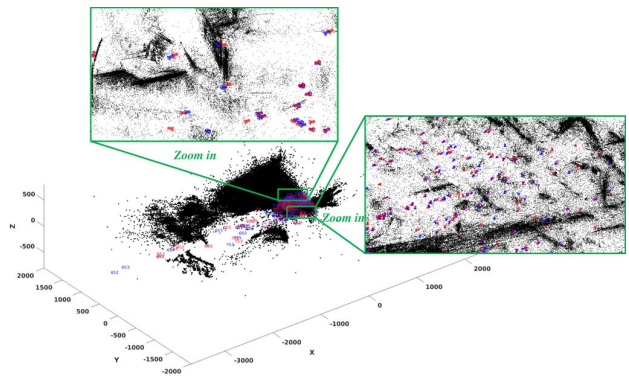


Figure 9. Estimated 6-DoF camera poses with respect to the 3D point clouds on the metric Dubrovnik dataset. The 3D points are denoted by black dots. Estimated and ground-truth camera poses are colored in red and blue, respectively. The number besides the camera model is the index of query image. (Best viewed on screen with zoom-in.) An *online demo* is available at https://www.youtube.com/watch?v=hBwdMcZhbfo&list=UUXw_IGWxWrOuhYhJ-BbmNnw

than Active-Search, are adequate for interactive applications ([11, 23, 31, 32, 42–44]). If a 3D map is very large, it is nearly inevitable to find repetitive structures. In order to evaluate our method's resilience to repetitive structure, we adopted (and modified) method of [54] for repeated feature removal. It requires points share similar visual word from lower-level of the vocabulary tree. Details and results for this test is given in supplementary material.

5.3. Test on a very large scale dataset (SF-0)

Now we attempt to test our method on the San Francisco (SF-0) dataset [12], which is the largest one among the four, containing about 30-millions map points. Besides its huge size, there are also other challenges associated with this dataset: *e.g.*, the provided images have very different characteristics (*i.e.* some were taken by cell-phone; some were cropped sub-images of Google street-view with unspecified camera model). Moreover, its total memory footprint for 3D points/descriptors is 28.5GB, exceeding our PC's memory size (24GB RAM). To be able to process the data, we used the raw GPS data provided for the query image to limit the search within a geographic region of about 600-meters in diameter, and we aim to refine GPS's coarse location estimation to very high precisions (*e.g.*, in meter or sub-meter/sub-degree)). Note that previous work on SF-0 also used GPS prior [12]. As before, if the number of inliers after RANSAC exceeds 12 points, we deem the localization is successful. Our method successfully localize 652 images (out of totally 803 query images), and the average time spent per image is 0.30s. We also ran Active Search [43, 44] on the same dataset under the same simplification using GPS. However it failed in most cases and was only able to localize 31 images. We suspect that the reason is due to the existence of large number of similar or repetitive visual descriptors in this very large-scale map (also confirmed by [31]), and Active-Search only makes local 2D-3D match decision based on visual word similarity. To the best of our knowledge, only two other methods have handled SF-0 data efficiently, but they used different local search heuristics, and none exploited global coherencies of the query set and the 3D point clouds. This test demonstrates that our method is able to handle problems of such bigger sizes, yet still under the same "random walk on Markov network" framework. In future we plan to tackle larger, *e.g.*, billion-point problems.

6. Conclusion

Scalability and Ambiguity are two major challenges for camera localization if a direct 2D-3D matching approach is employed. In this paper, We have proposed a principled global method to address both issues in the same framework. Our key idea is, contrary to existing methods which mostly rely on local similarity search, we formulate the problem as a global inference task performed on a Markov graph of the 3D map. The special structure of the graph, in particular through its edges which encode the co-visibility relationships among all 3D points, allows the inference procedure to take account of not only individual feature match's visual similarity, but also the global compatibilities as measured by the pair-wise co-visibility. Inspired by Google's PageRank, we solved the inference task via a random walk algorithm. To the best of our knowledge, this paper represents a novel and original contribution to the literature of image-based camera localization. Since the proposed method advocates a global, holistic view to looking at the problem, we hope it will inspire other new ideas which may lead to more powerful solutions. For instance, currently we are investigating the potential usefulness of other MRF inference techniques (such as Efficient-LBP [56] or graph-cut with co-occurrence [30]) for solving even larger camera localization instances.

Acknowledgment

This work was supported by China Scholarship Council (201506290131), ARC grants (DP120103896, LP100100588, CE140100016, DE140100180), Australia ARC Centre of Excellence Program on Robotic Vision, NICTA (Data61), Natural Science Foundation of China (61420106007, 61473230, 61374023), State Key Laboratory of Geo-information Engineering (NO.SKLGIE2015-M-3-4) and Aviation Fund of China (2014ZC53030). We gratefully acknowledge the support of NVIDIA Corporation with the donation of the GPU. We thank all anonymous reviewers for their valuable comments.

References

[1] Stochastic matrix. https://en.wikipedia.org/wiki/Stochastic_matrix. 4

[2] tf-idf. https://en.wikipedia.org/wiki/Tf%E2%80%93idf. 3

[3] Lane Level Localization on a 3D Map . http://www.acmmm.org/2017/challenge/lane-level-localization-on-a-3d-map/. 6

[4] S. Agarwal, Y. Furukawa, N. Snavely, I. Simon, B. Curless, S. M. Seitz, and R. Szeliski. Building rome in a day. *Communications of the ACM*, 54(10):105–112, 2011. 1, 2, 3

[5] P. F. Alcantarilla, K. Ni, L. M. Bergasa, and F. Dellaert. Visibility learning in large-scale urban environment. In *IEEE International Conference on Robotics and Automation*, pages 6205–6212. IEEE, 2011. 2

[6] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic. Netvlad: Cnn architecture for weakly supervised place recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5297–5307, 2016. 2

[7] R. Arandjelovic and A. Zisserman. All about vlad. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1578–1585, 2013. 2

[8] R. Arandjelović and A. Zisserman. Dislocation: Scalable descriptor distinctiveness for location recognition. In *Asian Conference on Computer Vision*, pages 188–204. Springer, 2014. 2, 4

[9] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. In *Proceedings of the Seventh International Conference on World Wide Web*, pages 107–117, 1998. 3

[10] M. Bujnak, Z. Kukelova, and T. Pajdla. A general solution to the p4p problem for camera with unknown focal length. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2008. 5, 6

[11] S. Cao and N. Snavely. Graph-based discriminative learning for location recognition. *International Journal of Computer Vision*, 112(2):239–254, 2015. 2, 8

[12] D. M. Chen, G. Baatz, K. Köser, S. S. Tsai, R. Vedantham, T. Pylvänäinen, K. Roimela, X. Chen, J. Bach, M. Pollefeys, et al. City-scale landmark identification on mobile devices. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 737–744. IEEE, 2011. 1, 5, 6, 8

[13] S. Choudhary and P. Narayanan. Visibility probability structure from sfm datasets and applications. In *European Conference on Computer Vision*, pages 130–143. Springer, 2012. 2, 7

[14] M. J. Cummins and P. M. Newman. Fab-map: Appearance-based place recognition and mapping using a learned visual vocabulary model. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 3–10, 2010. 2

[15] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse. Monoslam: Real-time single camera slam. *IEEE transactions on pattern analysis and machine intelligence*, 29(6), 2007. 1

[16] R. Díaz and C. C. Fowlkes. Cluster-wise ratio tests for fast camera localization. *arXiv preprint arXiv:1612.01689*, 2016. 2

[17] Y. Feng, L. Fan, and Y. Wu. Fast localization in large-scale environments using supervised indexing of binary features. *IEEE Transactions on Image Processing*, 25(1):343–358, 2016. 2

[18] J.-M. Frahm, P. Fite-Georgel, D. Gallup, T. Johnson, R. Raguram, C. Wu, Y.-H. Jen, E. Dunn, B. Clipp, S. Lazebnik, et al. Building rome on a cloudless day. In *European Conference on Computer Vision*, pages 368–381. Springer, 2010. 1, 2, 3

[19] D. Gálvez-López and J. D. Tardos. Bags of binary words for fast place recognition in image sequences. *IEEE Transactions on Robotics*, 28(5):1188–1197, 2012. 2

[20] Google matrix. https://en.wikipedia.org/wiki/Google_matrix. 5

[21] T. H. Haveliwala. Topic-sensitive pagerank. In *Proceedings of the 11th international conference on World Wide Web*, pages 517–526. ACM, 2002. 5

[22] I. C. Ipsen and S. Kirkland. Convergence analysis of a pagerank updating algorithm by langville and meyer. *SIAM journal on matrix analysis and applications*, 27(4):952–967, 2006. 5

[23] A. Irschara, C. Zach, J.-M. Frahm, and H. Bischof. From structure-from-motion point clouds to fast location recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2599–2606. IEEE, 2009. 1, 2, 6, 8

[24] H. Jegou, M. Douze, and C. Schmid. Hamming embedding and weak geometric consistency for large scale image search. In *European conference on computer vision*, pages 304–317. Springer, 2008. 4

[25] E. D. Johns and G.-Z. Yang. Pairwise probabilistic voting: Fast place recognition without ransac. In *European conference on computer vision*, pages 504–519. Springer, 2014. 2

[26] A. Kendall and R. Cipolla. Geometric loss functions for camera pose regression with deep learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 2

[27] A. Kendall, M. Grimes, and R. Cipolla. Posenet: A convolutional network for real-time 6-dof camera relocalization. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2938–2946, 2015. 2

[28] H. J. Kim, E. Dunn, and J.-M. Frahm. Learned contextual feature reweighting for image geo-localization. In *CVPR*, 2017. 2

[29] H. W. Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955. 5

[30] L. Ladicky, C. Russell, P. Kohli, and P. H. Torr. Graph cut based inference with co-occurrence statistics. In *European Conference on Computer Vision*, pages 239–253. Springer, 2010. 8

[31] Y. Li, N. Snavely, D. Huttenlocher, and P. Fua. Worldwide pose estimation using 3d point clouds. In *European Conference on Computer Vision*, pages 15–29. Springer, 2012. 1, 2, 7, 8

[32] Y. Li, N. Snavely, and D. P. Huttenlocher. Location recognition using prioritized feature matching. In *European Conference on Computer Vision*, pages 791–804. Springer, 2010. 1, 2, 6, 7, 8

[33] M. Lourakis and X. Zabulis. Model-based pose estimation for rigid objects. In *International Conference on Computer Vision Systems*, pages 83–92. Springer, 2013. 5, 6

[34] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004. 1, 2, 4, 5

[35] S. Middelberg, T. Sattler, O. Untzelmann, and L. Kobbelt. Scalable 6-dof localization on mobile devices. In *European conference on computer vision*, pages 268–283. Springer, 2014. 1

[36] M. Muja and D. G. Lowe. Scalable nearest neighbor algorithms for high dimensional data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36, 2014. 4

[37] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos. Orb-slam: a versatile and accurate monocular slam system. *IEEE Transactions on Robotics*, 31(5):1147–1163, 2015. 1

[38] D. Nister and H. Stewenius. Scalable recognition with a vocabulary tree. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 2161–2168. Ieee, 2006. 4

[39] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE, 2007. 6

[40] T. Sattler, M. Havlena, F. Radenovic, K. Schindler, and M. Pollefeys. Hyperpoints and fine vocabularies for large-scale location recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2102–2110, 2015. 1, 2

[41] T. Sattler, M. Havlena, K. Schindler, and M. Pollefeys. Large-scale location recognition and the geometric burstiness problem. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 4

[42] T. Sattler, B. Leibe, and L. Kobbelt. Fast image-based localization using direct 2d-to-3d matching. In *International Conference on Computer Vision*, pages 667–674. IEEE, 2011. 1, 3, 4, 6, 7, 8

[43] T. Sattler, B. Leibe, and L. Kobbelt. Improving image-based localization by active correspondence search. In *European Conference on Computer Vision*, pages 752–765. Springer, 2012. 1, 2, 3, 4, 5, 6, 7, 8

[44] T. Sattler, B. Leibe, and L. Kobbelt. Efficient effective prioritized matching for large-scale image-based localization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PP(99):1–1, 2016. 1, 2, 3, 4, 6, 7, 8

[45] T. Sattler, B. Leibe, and L. Kobbelt. *Exploiting Spatial and Co-visibility Relations for Image-Based Localization*, pages 165–187. Springer International Publishing, Cham, 2016. 2

[46] T. Sattler, A. Torii, J. Sivic, M. Pollefeys, H. Taira, M. Okutomi, and T. Pajdla. Are large-scale 3d models really necessary for accurate visual localization? In *CVPR 2017-IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 2, 6

[47] T. Sattler, T. Weyand, B. Leibe, and L. Kobbelt. Image retrieval for image-based localization revisited. In *BMVC*, volume 1, page 4, 2012. 1, 2, 4

[48] J. L. Schönberger and J.-M. Frahm. Structure-from-motion revisited. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 2, 3

[49] J. Shotton, B. Glocker, C. Zach, S. Izadi, A. Criminisi, and A. Fitzgibbon. Scene coordinate regression forests for camera relocalization in rgb-d images. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2930–2937, 2013. 2

[50] Y. Song, X. Chen, X. Wang, Y. Zhang, and J. Li. 6-dof image localization from massive geo-tagged reference images. *IEEE Transactions on Multimedia*, 18(8):1542–1554, 2016. 2

[51] L. Svarm, O. Enqvist, F. Kahl, and M. Oskarsson. City-scale localization for cameras with known vertical direction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PP(99):1–1, 2016. 1, 2, 7

[52] L. Svarm, O. Enqvist, M. Oskarsson, and F. Kahl. Accurate localization and pose estimation for large 3d models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, June 2014. 1, 2

[53] H. Tong, C. Faloutsos, and J. Y. Pan. Fast random walk with restart and its applications. In *Data Mining, 2006. ICDM\'06. Sixth International Conference on*, pages 613–622. IEEE Computer Society, 2006. 3

[54] A. Torii, J. Sivic, M. Okutomi, and T. Pajdla. Visual place recognition with repetitive structures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(11):2346–2359, Nov 2015. 8

[55] J. Ventura, C. Arth, G. Reitmayr, and D. Schmalstieg. Global localization from monocular slam on a mobile phone. *IEEE transactions on visualization and computer graphics*, 20(4):531–539, 2014. 1

[56] J. S. Yedidia, W. T. Freeman, and Y. Weiss. Understanding belief propagation and its generalizations. *Exploring artificial intelligence in the new millennium*, 8:236–239, 2003. 8

[57] P. A. Zandbergen and S. J. Barbeau. Positional accuracy of assisted gps data from high-sensitivity gps-enabled mobile phones. *Journal of Navigation*, 64(03):381–399, 2011. 1

[58] B. Zeisl, T. Sattler, and M. Pollefeys. Camera pose voting for large-scale image-based localization. In *IEEE International Conference on Computer Vision*, December 2015. 2, 7