Improved Image-Based Localization Using SFM and Modified Coordinate System Transfer

Mahdi Salarian, *Member, IEEE*, Nick Iliev, *Member, IEEE*, Ahmet Enis Çetin, *Fellow, IEEE*, and Rashid Ansari, *Fellow, IEEE*

***Abstract*—Accurate localization of mobile devices based on camera-acquired visual media information usually requires a search over a very large GPS-referenced image database collected from social sharing websites like Flickr or services such as Google Street View. This paper proposes a new method for reliable estimation of the actual query camera location by optimally utilizing structure from motion (SFM) for three-dimensional (3-D) camera position reconstruction, and introducing a new approach for applying a linear transformation between two different 3-D Cartesian coordinate systems. Since the success of SFM hinges on effectively selecting among the multiple retrieved images, we propose an optimization framework to do this using the criterion of the highest intraclass similarity among images returned from retrieval pipeline to increase SFM convergence rate. The selected images along with the query are then used to reconstruct a 3-D scene and find the relative camera positions by employing SFM. In the last processing step, an effective camera coordinate transformation algorithm is introduced to estimate the query's geo-tag. The influence of the number of images involved in SFM on the ultimate position error is investigated by examining the use of three and four dataset images with different solution for calculating the query world coordinates. We have evaluated our proposed method on query images with known accurate ground truth. Experimental results are presented to demonstrate that our method outperforms other reported methods in terms of average error.**

***Index Terms*—Image-based localization, BOF, retrieval, GPS uncertainty.**

I. Introduction

FINDING the accurate location of an image generated by a mobile device is crucial in a variety of different applications such as navigation, location-based services, and augmented reality. It also improves the quality of travel experience for online users who are searching for landmarks.

Even though traditional approaches that utilize the GPS data or distance from cellular towers are useful for performing this task, the adequacy of this approach depends mostly on satisfactory access to the satellite signal. In practice, GPS information is usually reliable when the device has a clear view of the sky to get the signal from at least four satellites. However, it is difficult to obtain accurate localization using a GPS-equipped device carried by a pedestrian who is moving on a street sidewalk in a dense urban area such as downtown Chicago. For instance, consider the raw GPS track extracted from a smart phone in the downtown area of Chicago shown in red and the actual track shown in blue in Fig. 1. It is evident that the level of localization error may place the pedestrian on a completely different street. It has been observed that, the GPS errors of a mobile phone are usually no greater than 100 meters [1]. Such a large error may not be acceptable in many applications. As a result, significant research effort has been directed at finding solutions to the problem of improved localization. This effort has sought to exploit other information from the sensors available in mobile devices [2], [3]. A large part of this effort has focused on using the camera-acquired visual media information that is available in any smart phone or mobile device. It relies on the notion of getting an accurate position of a query image generated by the camera by searching over a very large GPS-referenced image dataset collected from social sharing websites like Flickr [4] or services such as Google Street View (GSV) using image retrieval methods. The search space can also be limited by extracting and leveraging additional media information from other sensors available in a device to improve the results.

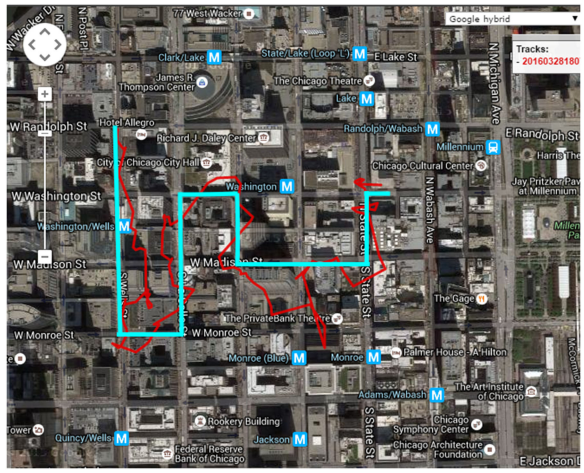


Fig. 1. Sample image with the pure GPS track extracted from a smart phone in downtown Chicago is shown in red and the actual track shown in blue.

Manuscript received September 18, 2017; revised January 26, 2018; accepted April 19, 2018. Date of publication May 23, 2018; date of current version November 15, 2018. This research was supported in part by a grant from the Elizabeth Morse Genius Charitable Trust. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Huadong Ma. *(Corresponding author: Mahdi Salarian.)*

M. Salarian, N. Iliev, and R. Ansari are with the Department of Electrical and Computer Engineering, University of Illinois at Chicago, Chicago, IL 60608 USA (e-mail: msalar2@uic.edu; niliev4@uic.edu; ransari@uic.edu).

A. E. Çetin is with the University of Illinois at Chicago, Chicago, IL 60608 USA, on leave from Bilkent University, Ankara 06800, Turkey (e-mail: aecyy@uic.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TMM.2018.2839893

Existing image retrieval approaches have turned out to be successful in finding correct matches especially when images have adequate textures. The key tools employed in these methods are features such as SIFT [5], SURF [6], BRISK [7] and FAST [8]. Although such features are powerful, the performance degrades with increasing size of the database, reducing the chances of finding a correct match. This can be improved by having prior information about the approximate coordinates which can be used to narrow the search space down. For example, the database is split into overlapping regions for the search in [9] or available sensor data such as GPS, Compass, and Estimated Positional Error (EPE), are utilized to narrow down the search space [10], [11].

Finding the best match is not the last step since it often returns multiple images along with their GPS positions. In fact, we can use those positions as our rough estimated position for the query that gives us a middle-of-the-street level accuracy. The resulting position error can be large if the query camera position is actually on the sidewalk. To achieve higher accuracy alternate methods such as those utilizing similarity matrix from two query-matching images (trifocals tensor) can be applied as done in [12]–[14]. Those methods utilize Structure From Motion (SFM) to estimate three camera positions: two matching-images camera positions and the query camera position.

Our contribution in this work is as follows: We propose a method to optimally select a subset of images from retrieved candidates with the highest intra-class similarity and distinct GPS tags to increase the convergence rate of SFM. In order to consider query features, we introduced a special similarity measure that takes into account those features common to all pairs of selected images which are shared with the query as well. Upon attaining convergence of SFM, the coordinate information obtained is fed to a new method introduced to find the transformation between camera-relative coordinate system and GPS coordinate system by adapting a cost function between two coordinate systems to control the transformation error below an acceptable level. In this context we use SFM to estimate relative camera positions for preferably four dataset and the query images. Since four relevant images may not be available for all samples, we have also examined using three retrieved images for implementing SFM. Since coordinate transformation is not possible for recovering three unknown parameters for three corresponding coordinates of images, we have reduced position vector to 2D using PCA as is used in [15]. Later a heuristic is used to estimate the query's z value of the position vector. All scenarios are experimentally compared in terms of accuracy showing significant improvement over reference reported methods. We have shown our proposed method can be employed to compensate localization error in available mobile devices.

The rest of the paper is organized as follows. In the next section related work on localization based on image retrieval techniques is described. Then, in Section III the algorithm used for the image retrieval pipeline is covered and the procedure for both pure retrieval and in combination with considering maximum GPS position error are presented. Section IV demonstrates our proposed method for query tag estimation based on SFM and followed by a specific 3D coordinate transformation. Sections V and VI show how our proposed method improves the performance in terms of accuracy when compared with other available methods.

II. Review of Related Work

Recent computer vision advances have made it possible to search for similar image in social sharing websites like Flickr or user generated datasets with sufficient reliability and for many applications [3], [16], [17]. A noteworthy application of this capability is searching a massive number of Geo-tagged images on the internet to find the location of a query image [3], [18]–[20]. A variety of methods have been proposed to do this. For instance, Reitmayr and Drummond [21] utilized an edge-based method to get street facades based on a 3-dimensional method.

The most efficient and accurate approach uses Content-Based Image Retrieval (CBIR) techniques relying on features such as SIFT and its variants. Some effective approaches frequently used in CBIR systems are Bag of Features (BOF) [22]–[24], Fisher Vector (FV) [25], [26] and vector of locally aggregated descriptors(VLAD) [27]. In these methods all feature descriptors are quantized to visual words with a clustering algorithm like K-Means. An image is represented by a histogram of a number of visual words and each image in a database has its own histogram. For finding the best match, the histogram of a query image is compared with all histograms in database.

There are different measures for finding similarity such as the inner product of two BOF vectors or specific distance functions [28], but a widely used procedure is the inverted file [29]. Some researchers have focused on clustering to find an efficient quantization technique for assigning each feature descriptor to a visual word. For example, Soft Assignment (SA) instead of Hard Assignment (HA) has been proposed to compensate for incorrect assignment of a sample feature [30]. Others have tried to select more distinctive features [31]–[34] while others have evaluated how repetitive structures influence the ultimate result [35], [36]. This is not necessarily the last step in the retrieval. Most of the methods select more than one candidate for a match in this step. An additional step, called homography verification performed by applying algorithm such as RANSAC [37] and its variants, [38], [39] are used to re-rank candidates. In fact, this step compensates for the weakness of image retrieval schemes based on BOF where the geometric information of images is ignored. Some other studies such as [2], [3] have proposed a method of using inertial sensor information and BOF to get more accurate results. Specifically in [3] prior knowledge of the approximate location from the cell towers is used to limit the search to the cellular area. In [10] and [11], uncertainty in the GPS location estimate is extracted and used to limit the search space. These approaches seek to exploit the available information to narrow down the search space. As a result, the accuracy and success rate of the retrieval is higher. This means more relevant candidate images are going to be returned in the retrieval step.

After a small set of best-matching images has been collected for a given query image, the next task is to estimate the query's location. Multi-view Structure From Motion (SFM) for the reconstruction of 3D camera poses from 2D-2D correspondences, or from 3D-2D correspondences can be used in this case [40].

Recent state-of-the-art approaches in this field such as [12], [13], and [15], find a similarity matrix from two query-matching images. These approaches utilize SFM to estimate three camera positions: two positions of two cameras corresponding to the two best-matching images and the query camera position, yielding a triplet of reconstructed 3D camera positions. Numerous triplets are typically generated (multiple matches with the query) and are subsequently processed by a least-squares fitting routine in order to compute the similarity matrix and generate a unique estimate of the query's location. They also reduce the 3D position vectors to 2D position vectors by dimensionality reduction techniques such as PCA. Based on their results the ultimate error range is still high which makes its use difficult in navigation. For example, we noticed that for some queries in different intersections, the estimated positions are found to be on the opposite side of the street from the actual position which makes navigation hard. A key limitation of currently used methods is using multiple SFM processing on pair of images returned by the retrieval pipeline along with the query which is computationally expensive. Our focus is using a single SFM on a subset of images from the retrieval with the highest similarity. So we formulate the image selection as an optimization problem. Then we proposed a method to directly find camera coordinate transformation parameters between camera relative centers from SFM to real world coordinates as described in the following sections.

III. Problem Formulation for Optimal Selection of Images for SFM

We now consider the framework for formulating the problem of optimally selecting a subset of retrieved images as input to SFM process. We first briefly describe the method we use for image retrieval to obtain N matching images from which a trimmed subset of k images is optimally selected for SFM implementation. Typically N may range between 10 to 50 whereas the choice of k is either three or four.

A. Retrieval of N Images

We first obtain N images that best match a query image. For this purpose several image retrieval methods may be employed. The main component of most image retrieval methods is the Bag Of Features (BOF) technique. In this approach, each image is represented with a vector containing the occurrence frequency of features (visual words). There are a variety of features such as SIFT, SURF or a normalized version of the SIFT called RootSIFT that have shown better performance. The query vector should be compared with all dataset vectors to find the most similar image. It is important to mention that the goal of our research is primarily on finding a better estimate of query position extracted from multiple matches from the dataset, and, not on improving the image retrieval engine itself. Any suitable method with good retrieval performance can be used for this stage.

As mentioned earlier, images can be represented by visual words, but the importance of the words varies. This importance is captured in the assigned weights using the Term Frequency-Inverse Document Frequency (TF-IDF). The weight of the visual word α in image i is

$$t_{\alpha,i} = f_{\alpha i} \times \log\left(\frac{N_{db}}{N_\alpha}\right) \tag{1}$$

where $f_{\alpha i}$ is the frequency of term α in image i, N_{db} is the number of images in the dataset and N_α is number of images containing visual word α. For each visual word α, note that the Inverse Document Frequency (IDF) is defined as

$$IDF(\alpha) = log(N_{db}/N_\alpha) \tag{2}$$

The value of IDF depends on multiple parameters such as the number of images in the dataset, the number of visual words, and the average number of features in images. As can be inferred more distinctive visual words receive higher weights. Let η be the number of visual words and $F_q = [f_1^q f_2^q \dots f_\eta^q]$ and $F_{db} = [f_1^{db} f_2^{db} \dots f_\eta^{db}]$ be the frequency of visual words $\alpha_1, \alpha_2, ..., \alpha_\eta$ for query and a dataset image, respectively. The jth, element F_q or F_{db} are the number of times feature descriptors of the query and a dataset image have been assigned to visual word α_j. The similarity between query and a given dataset image (vectors) can be computed by (3).

$$SIM(I_q, I_{db}) = \frac{\sum_{\alpha=1}^{\eta} IDF(\alpha)\, \min(f_\alpha^q, f_\alpha^{db})}{(\sum_{\alpha=1}^{\eta} IDF(\alpha) f_\alpha^q)(\sum_{\alpha=1}^{\eta} IDF(\alpha) f_\alpha^{db})} \tag{3}$$

The above similarity measure is different from the commonly used Cosine similarity measure. It is experimentally observed that it produces more robust results than the Cosine similarity measure. Our procedure for implementation of the basic image retrieval engine consists of the following steps:

1) Find the RootSIFT features for all images in a database.
2) Cluster features using the Approximate Nearest Neighbor algorithm (ANN) into η clusters (visual words).
3) Find the closest visual word (cluster center) for each feature in database images and represent each image by a vector showing the frequency of each visual word.
4) Apply TF-IDF using (1) and normalize the vectors.
5) Find the best N matches based on the score obtained for the dataset image using distance criteria.
6) Re-rank N closest images based on homography verification by applying RANSAC.

In order to achieve higher recall we used the Adaptive Assignment algorithm [36]. This algorithm, which assigns different number of visual words to different features improves recall. It is worth mentioning that any method could be used for the image retrieval pipeline. We can further improve the result by considering prior knowledge of the location from GPS as described in Section III-B.

B. Considering Prior Knowledge of Location From GPS

As mentioned in previous sections, the result of retrieval should be fed to our proposed method for query geo-tag estimation. One option to achieve a better result is taking prior knowledge from the query position into account. Some reported studies have used noisy location data. For example in [9] coarse location data from cellular tower and triangulation are used to

Algorithm 1: Image Candidate Selection Algorithm

1: **Input:** Set of N_{db} images $\mathbf{S} = \{I_1^{db}, I_2^{db}, \ldots, I_{N_{db}}^{db}\}$, $R_{\max}$, I_q
2: **Output:** $\tilde{\mathrm{S}}$ which is the set of all images in the region limited by $R_{\max}$
3: **for** $I_i^{db} \in \mathbf{S}$ **do**
4: **if** $D_{Geo}(I_i^{db}, I_q) < R_{\max}$ **then**
5: add I_i^{db} to $\tilde{\mathrm{S}}$
6: **end if**
7: **end for**
8: **Output:** find $\tilde{\mathrm{S}}$ as the set of image matching candidates for the query image I_q

limit the search region. Another option for narrowing down the search space is considering maximum error of the GPS which is denoted here as $R_{\max}$. Suppose the GPS coordinates of two images I_1 and I_2 are given by (θ_1, ϕ_1) and (θ_2, ϕ_2) where θ_i and ϕ_i are the latitude and the longitude for image i. The Geo-distance between locations of these two images is computed by (4).

$$D_{Geo}(I_1, I_2) = \cos^{-1}(\sin(\theta_1)\sin(\theta_2) + \cos(\theta_1) \cos(\theta_2)\cos(\phi_2 - \phi_1)) \times R_e \tag{4}$$

where R_e is the radius of the earth that is approximately 6371 kilometers. The search space can be limited to those images located in a circle with the radius of $R_{\max}$. The procedure to limit the search space for the query image I_q is described in Algorithm 1.

For the San Francisco dataset the maximum reported error is 300 meters. Based on our experience in Chicago, the error in the position estimated by a smart phone such as iPhone 5, iPhone 6, Nexus 6, or Galaxy S6, is typically less than 100 meters. This is because those phones benefit from other sources of data such as cellular towers and inertial measurement unit (IMU). The search space therefore turns out to be smaller for real-world applications.

The final common step in most of image retrieval algorithms is the application of geometry verification based on RANSAC to re-rank the limited number of candidate based on the number of inlier features. This step mitigates the weakness of systems based on BOF which ignore the geometric information of features. To go forward and estimate the actual position of the query, more than one image is needed. This is because estimating the camera position by using just a single image and considering fundamental matrix between the query and the best match, even when models for both cameras are available, would not be accurate enough for our purpose. For our proposed 3D coordinate transformation method at least three candidates with distinct GPS tags are required. To acquire candidates that are most similar to a query, criteria such as the number of inliers between the query features and the candidate features can be considered for the re-ranking and removing irrelevant candidates. Along with this criterion, another suitably devised step should be applied to ensure return of the best candidate images with distinct GPS tags and highest intra-class similarity. The procedure for optimally selecting candidates is discussed in the next sub-section.

C. Optimum Selection Among the Retrieved Images

Suppose N images with location coordinates $g_i, i = 1, ..., N$ are selected after re-ranking. We wish to select k images out of N, where k is preferably four. If that is infeasible, then k equal to three images may be selected if possible. A simple way to select k images is to find images with distinct GPS tag and select k images with the highest number of inliers. Such a set of images is not necessarily the best choice for SFM processing since the selection relies only on the number of pairwise matches (inliers) between the query and all candidates while the number of matched features between each pair of candidate images is not taken into account.

It is important to note that a set of candidate images is the best choice when each member of this set shares the highest number of common features with the other members. In our case, while multiple images per location exist, we seek a method that optimally selects the set so that each member of the set has the highest consensus on common features with other members as well as with the query image. The solution is facilitated by defining a pairwise dissimilarity measure, w_{ij}, between distinct image i and j. An undirected graph $G = (V, E, w)$ with vertices $V = 1, 2, ..., N$ corresponding to image $I_1, I_2, ..., I_N$ with location $g_1, g_2, ..., g_N$, the set E of edges, and the set w of weights can then be created. By this definition, the more similar images will have the lower w_{ij}. Now the problem is to find a subset $G^\star = (V^\star, E^\star, w^\star)$, $V^\star \subset V$, $E^\star \subset E$, with k vertices, $k < N$, that minimize the total weights:

$$V^{k\star} = \underset{V^k \subset V}{\operatorname{argmin}} \sum_{\substack{i,j \in V^k \\ g_i \neq g_j \\ i \neq j}} w_{ij} \tag{5}$$

Here V can be partitioned into clusters with distinct GPS-tags. We now devise a solution to the problem of optimal selection of k images using the framework just described.

IV. Implementing Solution to Optimal Image Selection for SFM

The problem of finding an optimal subset from a set has been studied extensively during last years [41], [42]. Since there likely to be a chance of multiple images per location, the algorithm should only select one image per location. We therefore employ the General Minimum Clique Problem (GMCP) to select one image in each cluster containing images with identical GPS-tags. In the following subsection we describe how our problem is formulated and solved by GMCP.

A. Candidates Selection by GMCP

In order to formulate and solve our optimal selection problem using GMCP, we start with N best retrieved images with world coordinates (GPS-tag) $g_j, j \in \{1, ..., N\}$, not necessarily distinct. Let h be the total number of distinct or unique location coordinates. The N candidates are then grouped into clusters $\{V_1, \ldots V_h\}, h \leq N$ with an identical GPS tag. So an arbitrary

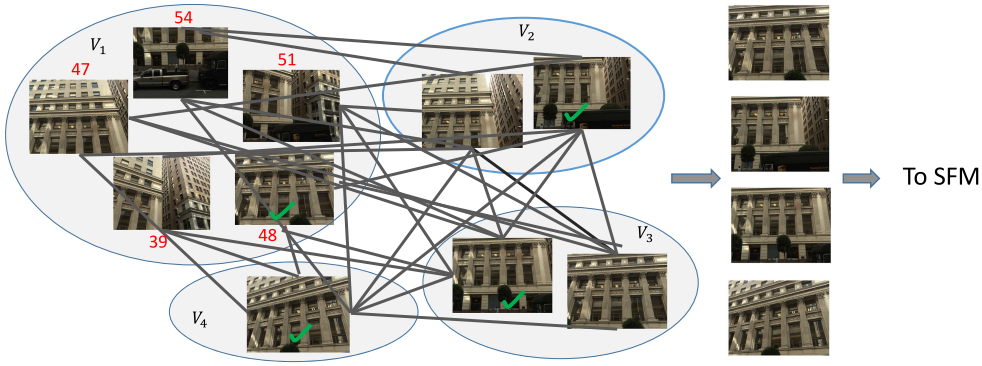


Fig. 2. Candidate selection by GMCP. Images with similar GPS-tags are placed to the same cluster. For cluster V_1, the number of inliers between each member and the query is shown in red. For each cluster only one image is returned by GMCP and is shown with a green check mark (edge weights are not shown in this figure). Note that for the cluster V_1, in our approach the two images with higher number of inliers (54 and 51) were not selected unlike the scenario in which the maximum number of inliers is the only criteria for image selection in each location.

cluster V_r, $1 \leq r \leq h$ contains different number of images associated with world coordinate g_r. With this partition of images, some clusters may contain only a single image meaning that the retrieval returned only one image for that location. Also h is usually larger than k (k is preferably 4) which exceeds our need of images for the next step. One possible solution is to keep first $k = 4$ clusters and find all images with the highest similarity. We choose to keep more clusters and then select only k images with the highest score from the result of GMCP. In order to solve our problem, for each member of all clusters, a similarity measure between image $i \in V_x$ and $j \in V_y$ where $x \neq y$ should be calculated. The number of inliers between a pair of images derived from geometry verification is a strong indicator of similarity. In the last steps, we only found the number of inliers between the query and limited number of candidate images. Applying geometry verification between each pair of candidates would be practically infeasible since it would require an unacceptable amount of time. In order to avoid this time complexity we propose the use of vectors containing frequency of visual words of images as defined in Section III. It is also important to incorporate the query visual words in computing the similarity between two images. This is because images selected in this stage along with the query should be fed to the SFM pipeline. So a desirable similarity measure should take into account those visual words that are common to two images as well as to the query image. We therefore introduce a query-contextualized image similarity measure. Suppose the vector of visual words for image I is represented by $F_I = \{f_1^I, f_2^I, ..., f_\eta^I\}$. In order to incorporate query visual words in computing similarity, the indices of non-zero visual words of the query are extracted and represented by $I_{nz}^q = \{u_1, ..., u_d\}$ where d is the number of non-zero visual words. We define the similarity between any pair of images i and j by (6):

$$\psi_{ij} = \sum_{k=1}^{d} \Delta(f_{u_k}^i)\Delta(f_{u_k}^j) / \left(\sum_{k=1}^{d} \Delta^2(f_{u_k}^i)\right)^{1/2} \times \left(\sum_{k=1}^{d} \Delta^2(f_{u_k}^j)\right)^{1/2} \tag{6}$$

where, for $x \in \mathbb{R}$,

$$\Delta(x) = \begin{cases} 1 & \text{if } x \neq 0 \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

Since $\Delta^2(x) = \Delta(x)$ the denominator in (6) can be reduced to

$$\left(\sum_{k=1}^{d} \Delta(f_{u_k}^i) \sum_{k=1}^{d} \Delta(f_{u_k}^j)\right)^{1/2} \tag{8}$$

This measure calculates the similarity between two images while taking into account the non-zero features of the query. In the next step, all selected images along with the query image should be fed to SFM step. The complexity of computing (6) is low since vectors are already available and summation is applied for the non-zero features of the query. A convenient measure of dissimilarity between image i and j can be defined by (9).

$$w_{ij} = 1 - \psi_{ij} \tag{9}$$

The next step is to find the subgraph $G^\star = (V^\star, E^\star, w^\star)$ with nodes $V^\star = \{v_1^\star, ..., v_h^\star\} \subset V$ where only one node is selected from each cluster, for instance $v_1^\star$ from V_1 and $v_h^\star$ from V_h, and subset of edges $E^\star \subset E$ that minimizes the total dissimilarity that for a feasible solution is:

$$T_{Dissimilarity}(V^\star) = \sum_{m=1}^{h} \sum_{l=m+1}^{h} w_{V^\star(m)V^\star(l)} \tag{10}$$

Fig. 2 shows the process of clustering images with only four clusters where the costs of edges are not shown. For the members of cluster one, V_1, the number of inliers between the query and each member is shown in red. In this case, clusters contain different numbers of images. The result of GMCP is shown with green check marks. As shown in cluster V_1, an image with 48 inliers with the query is selected as a best candidate. Note that for the cluster V_1, the two images with a higher number of inliers (54 and 51) were not selected by our proposed method based on GMCP. This is different from the scenario in which the maximum number of inliers is the only criteria for image selection in each location. Without use of GMCP, the candidate selected

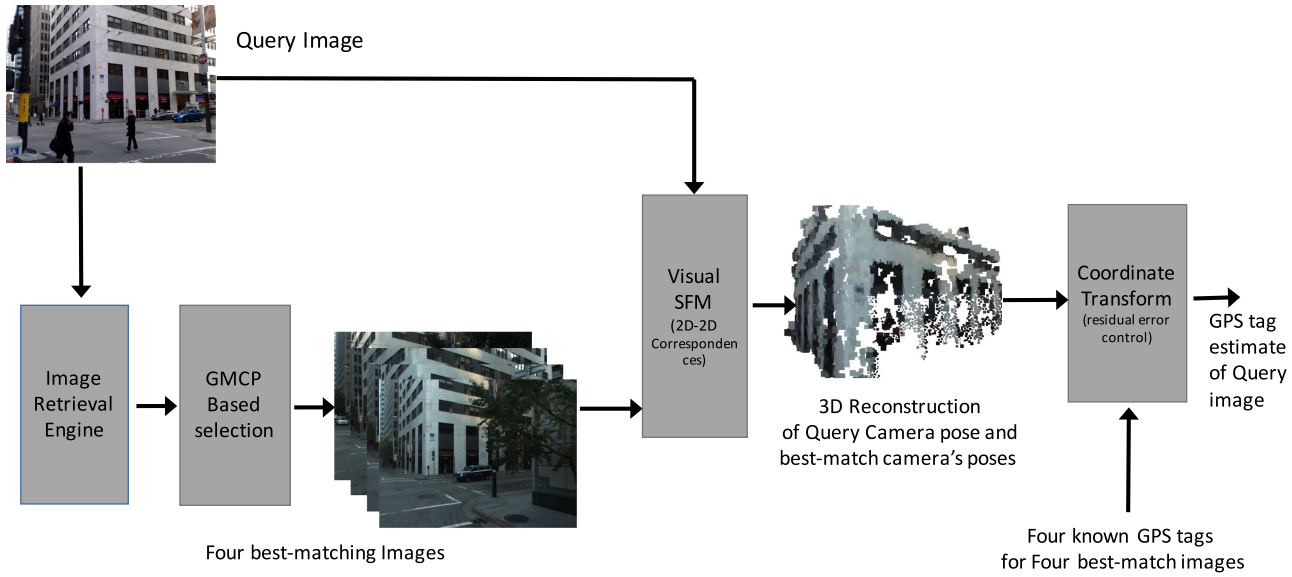


Fig. 3. Proposed pipeline for image-based localization (using four matching images from dataset).

from the cluster V_1 is the image with 54 inliers. We compare the SFM convergence rate in Section VI for both methods.

B. Generalized Minimum Clique Problem (GMCP)

Generalized Minimum Clique Problem (GMCP) can be used when the costs of edges are non-negative and graph is $|K|$-partite complete. Unlike a minimum clique problem, GMCP substitutes nodes with cluster of nodes. In this problem nodes of a given graph are partitioned into disjoint clusters. The goal is to find a subgraph with minimum cost while selecting only one node from each cluster. Each cluster furnishes only one of its nodes to the subgraph. This algorithm has been used recently in Computer Vision for multi-object tracking [43]. Suppose we are given a graph $G = (V, E, w)$ with nodes $V = \{v_1, ..., v_N\}$ and these N nodes are grouped into h sets of nodes called clusters $V_1, V_2, ..., V_h$. Note that $V = V_1 \cup V_2 \cup ... \cup V_h$ and $V_x \cap V_y = \varnothing$ for all $x, y \in \{1, ..., h\}$ where $h \in \mathbb{Z} : 1 \leq h \leq N$ and $x \neq y$. As mentioned earlier, a cost w_{ij} is assigned to the edge between nodes $i \in V_x$ and $j \in V_y$, for $x \neq y$. Now the objective is to find a subgraph $G^\star = (V^\star, E^\star, w^\star)$ with nodes $V^\star = \{v_1^\star, ..., v_h^\star\} \subset V$ which is composed of only one node from each cluster together with associated subset of edges $E^\star \subset E$ that is minimized the total edge cost. For such a problem GMCP can find a feasible solution with minimum cost which is in fact the total weights of all edges in $E^\star$. So based on the formulation of our problem in Section IV-A, GMCP can return the subset with highest intra-cluster similarity which leads to a higher convergence rate in the SFM step. In the next section we discus how the selected candidate images are used to estimate the query camera position.

V. Query Camera Position Estimation

A. Estimate Query Location by Four Dataset Images

The image retrieval process selects multiple matching images for a specific query. Each of the matching images has a known GPS tag which is used in our novel procedure for estimating the query camera's location. The proposed method is illustrated in Fig. 3. A key concept in our approach for query GPS tag estimation is the selection of a subset of images with the highest inter-class similarity using GMCP as described in IV-A and then obtaining a $3D - 3D$ coordinate transformation from one $3D$ coordinate system (eg. camera centers in camera $3D$ space as reconstructed from multi-view SFM) to another $3D$ coordinate system (eg. GPS tags in absolute world $3D$ Cartesian space for the same cameras). Fig. 4 illustrates the concept with four cameras (images), with centers C_1, C_2, C_3, and C_4, using four images obtained in the previous step, with the query camera as the fifth camera with center at C_5. C_1 to C_5 represent camera 3D center coordinates which have been reconstructed by multi-view SFM. We use the VisualSFM package [44] for this task and extract the coordinates C_1 to C_5 based on four matching images (dataset images) for the given query image. The details of camera center localization with SFM are as follows. Assume that for a given query image I_q a set of h images, $T = \left\{I_{v_1^\star}, I_{v_2^\star}, \ldots, I_{v_h^\star}\right\}$ $h >= 4$ is returned by the GMCP. Here $I_{v_1^\star}$ is image corresponding to node $v_1^\star$. The corresponding GPS tags for those h images are denoted with the set of locations $L = \left\{P_{v_1^\star}, P_{v_2^\star}, \ldots, P_{v_h^\star}\right\}$.

The set $\{I_{v_1^\star}, I_{v_2^\star}, I_{v_3^\star}, I_{v_4^\star}, I_q\}$ should then be processed with VisualSFM. Upon convergence to five camera center locations, $C_1, ..., C_5$, the quintuplet $C = \{C_1, .., C_5\}$ is used to obtain absolute world coordinate locations. If fewer than five relative camera centers are returned, SFM does not converge. It is worth mentioning that there would be a possibility to re-run the process using three best candidates as described in Section V-B.

In the following, without loss of generality, we have adopted the convention that C_5 in C corresponds to the camera center location for query image I_q. Locations $C_1, ..., C_4$ correspond to the cameras for the matching dataset images. Each camera center location in C is specified with $3D$ Cartesian coordinates in camera referenced space. Before computing the transformation

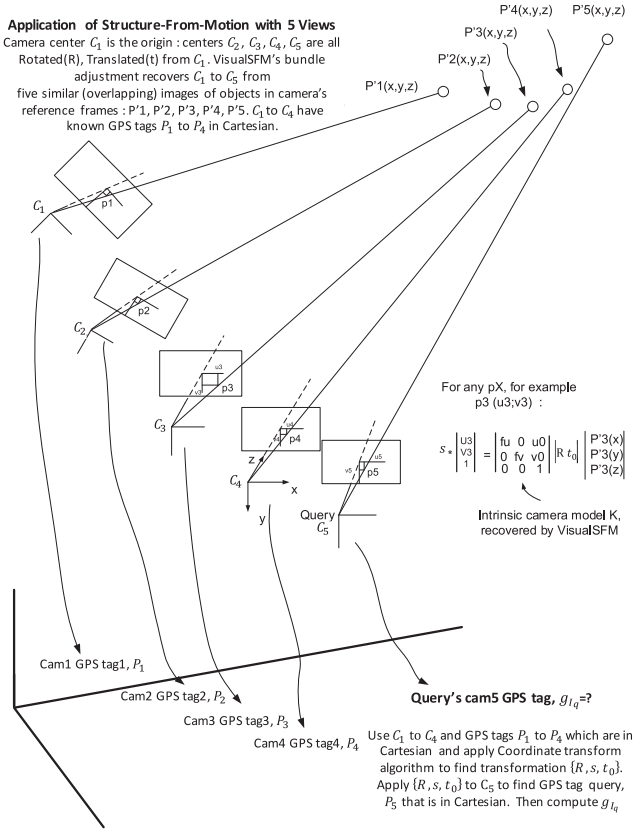


Fig. 4. Transformation from camera-referenced 3D coordinate system based on SFM to real world-referenced 3D Cartesian location using four dataset images.

the GPS tag of dataset images should be converted to Cartesian coordinates. The conversion equations are as follows. Assuming the GPS tag contains latitude and longitude pair (θ, ϕ), the coordinates x, y, z are computed by:

$$\begin{aligned} x &= R_e \cos(\theta)\cos(\phi) \\ y &= R_e \cos(\theta)\sin(\phi) \\ z &= R_e \sin(\theta) \end{aligned} \tag{11}$$

where R_e is the radius of Earth. Suppose the GPS tags for the four dataset images used in SFM are represented as $P_1, \ldots, P_4$ in Cartesian coordinates. Algorithm 2 is used for deriving the transformation from camera-referenced to absolute reference coordinates. It uses the values for the matching dataset images, P_1 to P_4 , and their relative locations C_1 to C_4 derived from SFM.

Two final steps are applied after Algorithm 2:

1) compute query's location P_5 (GPS tag in Cartesian coordinates) as $P_5 = t_0 + sRC_5$.

and

2) convert P_5 back to GPS latitude/longitude g_{I_q}.

In step 2 of algorithm 2, points C_1 to C_4 can be considered as points in the left coordinate system. This is a 3D Cartesian coordinate system for all reconstructed camera centers with origin at C_1. We label these as $y_{l,i}$ with $i = 1$ to 4. Locations P_1 to P_4 can be considered as points in the right coordinate system.

Algorithm 2: Camera Referenced Coordinate System to World Coordinate System Transformation

1: **Input:** Camera center coordinates $C_1, ..., C_4$ from quintuplet C and their corresponding GPS tags in 3D spherical coordinates
2: Convert GPS tags $P_{v_1^\star}$ to $P_{v_4^\star}$ for dataset images $I_{v_1^\star}$ to $I_{v_4^\star}$ to 3D Cartesian coordinates P_1 to P_4 by (11)
3: Use C_1 to C_4 and P_1 to P_4 as inputs in computing the rotation matrix R, translation vector t_0, and scaler s.
4: Compute the residual error evaluated for the current values of R, t_0, and s. If the error is less than a desired threshold, the localization error is acceptable.
5: **Output:** Matrix R, column vectors t_0 and s, defining the linear transformation from the camera referenced coordinate system to the world coordinate system.

This is a 3D Cartesian coordinate system representing the GPS tags for the same cameras. We label these as $y_{r,i}$ with $i = 1$ to 4. The transformation we seek, from the left to right coordinate systems, is given by:

$$y_r = sRy_l + t_0 \tag{12}$$

where $s \in \mathbb{R}$ is a scale factor, $t_0 \in \mathbb{R}^3$ is the translational offset, and $R \in \mathbb{R}^{3\times3}$ is a rotation matrix applied to 3×1 column vector y_l.

Because of measurement errors, we are unlikely to find a scale factor, a translation vector, and a rotation matrix such that the transformation equation above is satisfied for each point exactly. Instead there will be a residual error given by:

$$e_i = y_{r,i} - sRy_{l,i} - t_0 \tag{13}$$

For general coordinate transformation problem and two sets of k points in left and right, the problem can be formulated as a Least Squares problem. The objective is to find a match matrix or correspondences m which represents the corresponding points in the left and right coordinates and transformation parameters R, s, t_0 which minimize mapping error from one set of points y_l onto another set of points y_r.

$$(t_0^\star, s^\star, R^\star, m^\star) = \underset{t_0,s,R,m}{\operatorname{argmin}} \sum_{i=1}^{n}\sum_{j=1}^{n} m_{ij} \|y_{r,i} - sRy_{l,j} - t_0\|^2 \tag{14}$$

In our application we know the GPS-tags of all images (for example four images) in dataset which are fed to SFM. Upon convergence of SFM, the camera centers corresponding to those four images but in camera referenced coordinate system can be extracted. Therefore correspondences are known from left to right systems which obviates the need to keep match matrix in (14). So we have:

$$(t_0^\star, s^\star, R^\star) = \underset{t_0,s,R}{\operatorname{argmin}} \sum_{i=1}^{n} \|y_{r,i} - sRy_{l,i} - t_0\|^2 \tag{15}$$

In our method we ideally use four (or three if four is infeasible) corresponding points. Each point has three variables. Therefore

more than eight equations are available which makes it feasible to find the transformation parameters with a closed form method described in [45] in $O(n)$ time. So we directly calculate R, s, t_0 as shown below. The first step according to [45] is computing the centriod of y_l and y_r.

$$\bar{y_l} = \frac{1}{n}\sum_{i=1}^{n} y_{l,i} \quad \bar{y_r} = \frac{1}{n}\sum_{i=1}^{n} y_{r,i} \tag{16}$$

Then points should be shifted with respect to the centroids:

$$y_{l,i}^{'} = y_{l,i} - \bar{y_l} \quad y_{r,i}^{'} = y_{r,i} - \bar{y_r} \tag{17}$$

Now by using $y_{l,i}^{'}$ and $y_{r,i}^{'}$ in the error e_i we have:

$$e_i = y_{r,i}^{'} - sRy_{l,i}^{'} - t_0^{'} \tag{18}$$

$$t_0^{'} = t_0 - \bar{y_r} + sR\bar{y_l} \tag{19}$$

The square of error in (15) can be minimized when $t_0^{'}$ is equal to zero. This yields

$$t_0 = \bar{y_r} - sR\bar{y_l} \tag{20}$$

Now for finding the translation, t_0, s and R should be computed. From [45] s can be computed as follows:

$$s = \sqrt{\sum_{i=1}^{n} \|y_{r,i}^{'}\|^2 / \sum_{i=1}^{n} \|y_{l,i}^{'}\|^2} \tag{21}$$

Now R can be calculated using the steps below. First compute M :

$$M = \sum_{i=1}^{n} y_{r,i}^{'} (y_{l,i}^{'})^{\intercal} \tag{22}$$

which is a 3×3 matrix. Then compute $B = (M^{\intercal}M)$ and find the eigenvalues $\lambda_1, \lambda_2, \lambda_3$ and eigenvectors $\hat{v_1}, \hat{v_2}, \hat{v_3}$ and express B using eigen-decomposition as follows:

$$B = \lambda_1 \hat{v_1}\hat{v_1}^{\intercal} + \lambda_2 \hat{v_2}\hat{v_2}^{\intercal} + \lambda_3 \hat{v_3}\hat{v_3}^{\intercal} \tag{23}$$

$$R = M\left(\frac{1}{\sqrt{\lambda_1}}\hat{v_1}\hat{v_1}^{\intercal} + \frac{1}{\sqrt{\lambda_2}}\hat{v_2}\hat{v_2}^{\intercal} + \frac{1}{\sqrt{\lambda_3}}\hat{v_3}\hat{v_3}^{\intercal}\right) \tag{24}$$

By substituting s and R from (21) and (24) into (20), the translation vector, t_0 can be computed. Now each point from left coordinate including the query point can be transformed to right side by:

$$y_r = sRy_l + t_0 \tag{25}$$

The total residual error (E_{Total}) resulting from the transformation is equal to

$$E_{Total} = \sum_{i=1}^{n} \|e_i\|^2 \tag{26}$$

We describe the result and the related error range for some samples in Section VI. Notice that [15] and [13] utilized multiple estimates of the query position derived from multiple running of SFM. Then an optimization approach (Random Walk) is employed to estimate query location. In order to avoid time complexity of multiple SFM, our proposed method runs SFM only once to compute the coordinate transformation parameters as mentioned above. Since four relevant images with distinct GPS-tags may not always be available for all queries, we have also examined the use of only three matching images. In order to adapt algorithm 2 to three dataset images, two methods have been proposed as described below.

B. Estimate Query Location Using Three Dataset Images

Four relevant images may not be found in every case. We therefore seek to recover the query GPS-tag using a smaller number of images. Until now we considered the use of four images since three unknown coordinate variables should be determined. In general for computing transformation parameters between m-dimensional vectors using the method described above, $m + 1$ corresponding points are required. So if three images are used, only two unknown coordinates such as x and y can be recovered. The advantage of this approach is that it can be applied to more query images since some of them do not have four relevant candidates with distinct GPS-tags. Although finding the transformation between the camera coordinate system and the real-world system in Cartesian coordinate is almost the same for four or three images, transfer from Cartesian coordinates to GPS tag (Lat, Long) is not possible without having corresponding 3D position vectors. To address that, we propose two different methods described below.

1) Finding Third Component by Averaging z: Since z values would be close for the query and dataset images, we seek to only recover x and y by computing coordinate transformation. We use x and y in left and right coordinates (camera-referenced coordinates and real-world coordinates) for three images to compute the transformation parameters, R, s, t_0. Then the transformation should be applied to the query location in left system to obtain the query location in real-world coordinate (only x and y). By having x and y , only calculating longitude is possible since z is required for calculating latitude. In order to have a reasonable estimate, an average of z, as shown in (27) below

$$Z_q \approx \frac{\sum_{i=1}^{3} Z_{db_i}}{3} \tag{27}$$

for those three candidates is computed. This is because we assume that there would not be an abrupt change in the z coordinate values among the selected images and the query. All three components of the location vector of the query, $[x, y, z]$, can be used for computing query's latitude and longitude.

2) Position Vector Reduction: Another method employed is dimension reduction. We applied Principal Component Analysis (PCA) to the 3D position vectors for transfer to 2D space. A coordinate transformation is then applied between those 2D vectors and their associated GPS tags. The query GPS tag can then be calculated directly. To examine how the whole process affects the ultimate accuracy, the same query images as used with the four images approach in Section V have been used for evaluation. Also, the experimental results for those queries which were not evaluated due to the lack of four relevant images are covered in the Section VI.

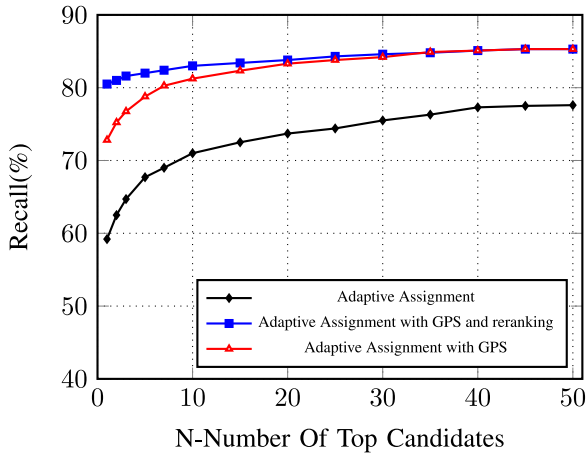


Fig. 5. Recall versus the number of top candidates for San Francisco dataset for different scenarios. Limiting the search area using Algorithm 1 improves recall.

VI. Performance Evaluation

In this research, we evaluated the performance of our method using the publicly available San Francisco dataset from [46] containing more than one million images.

The reason for using this dataset is that the location error for its queries is high since the images are captured mostly in downtown San Francisco. Also, it contains more images per area which is necessary in our research which is based on atleast three images with distinct GPS-tag in SFM process. We have used only perspective central images, PCIs, from the dataset since they are less likely to cause distortions during the VisualSFM 3D reconstruction. The San Francisco dataset provides a set of 803 query images, usually taken from a pedestrian's perspective at street level. Each query image is also annotated with a ground truth GPS tag which is noisy. Since accurate ground truth is required for evaluating the final results, we have used ground truth from [12] and compared our result with the results provided in this article. We also used Adaptive Assignment [36] while $\eta = 200k$ for the image retrieval engine. To assess the performance, recall as used in [46], [47], and described in (28) has been used. To further improve recall rate, rough position and maximum GPS error are used for narrowing down the search space. For the San Francisco dataset $R_{\max} = 300$ is reported. By considering $R_{\max}$ in Algorithm 1, recall has improved as shown in Fig. 5. This figure also covers the recall curves after re-ranking. Note that we have used San Francisco 2011 ground truth which does not cover all the query images. As a result perfect recall is not attainable, where

$$recall = \frac{\#of\ relevant\ retreived\ images}{\#of\ retreived\ images} \tag{28}$$

We found that relevant images typically have more than 20 inliers. So candidate images with fewer than 20 inliers have been filtered out directly. From 803 original queries, our retrieval pipeline finds candidates which have at least 20 inliers for 453 queries . For 398 queries, more than four images are found. Although retrieval curves for $N = 50$ are shown, we have selected 15 images for the GMCP ($N = 15$). The reason is that the recall is almost flat for the $N > 15$. Subset of four images is then selected with two different approaches discussed in Section III-C. For queries for which the number of retrieved candidates is less than 15, all retrieved images proceed in the next step. Fig. 6 shows a query with multiple candidates returned from the retrieval pipeline and four images opted by two approaches. Although images appear to be similar in both sets, the set returned by GMCP converged in SFM processing while the other did not. Fig. 7 represents a sample which did not converge for both methods while they contain different images. In Fig. 7, G represents images selected by GMCP and U by distinct GPS tag. Images which are not selected are shown by NS.

For the 277 queries from 398, both approaches, returned identical subsets. Among those sets, 141 of them converge and produce 3D coordinates. For the reminding 121 queries we got different subsets with 42 convergences for the method based on finding distinct GPS tag and 61 convergences for the GMCP based approach. It is worth mentioning that GMCP based selection converged for all samples which distinct based method converged. We also found that localization error is low and acceptable for our application when the SFM converges with five images including the query. This is because the amount of error introduced by the approach we have used for coordinate transformation is low. Therefore the total location error is acceptable upon convergence of the SFM. Although some queries could find more than three candidates, the number of candidates with distinct tag is less than four. We have evaluated our method based on three candidates as discussed in V-B and found success in 47 more cases.

Table I illustrates several query images from the San Francisco dataset, the corresponding four matching images for 3D camera pose reconstruction, and the residual error $\sum_{i=1}^{n} \|e_i\|^2$ in squre kilometers. The reconstructed best-matching camera center positions from SFM are listed for each query. Note that we rely on VisualSFM [44] for convergence i.e. estimate camera locations for cameras including the query camera. When convergence is not achieved with four images, the same approaches with three images can be applied. For each query, the amount of residual error obtained from closed-form formula for the transformation is listed in the rightmost column. Those values of errors which are introduced by coordinate transformation are acceptable for all quintuplets considered in testing. This error is sum of errors for coordinate transformation of four images from camera coordinates to real world coordinates system through the computed R, s, and t_0 and is acceptable for our application specially when we know the precision of the ground truth is in the range of a meter. The specific parameters of the corresponding transformation, scale factor s, translational offset t_0, and rotation matrix R are listed in the adjacent column. Details are presented in the Section VI.

Table II depicts an illustrative random subset of query images and the distance error in meters between the estimated GPS tag and the ground truth tag for each query when four dataset images are used. When VisualSFM does not converge using four candidate images, we considered the result for that query

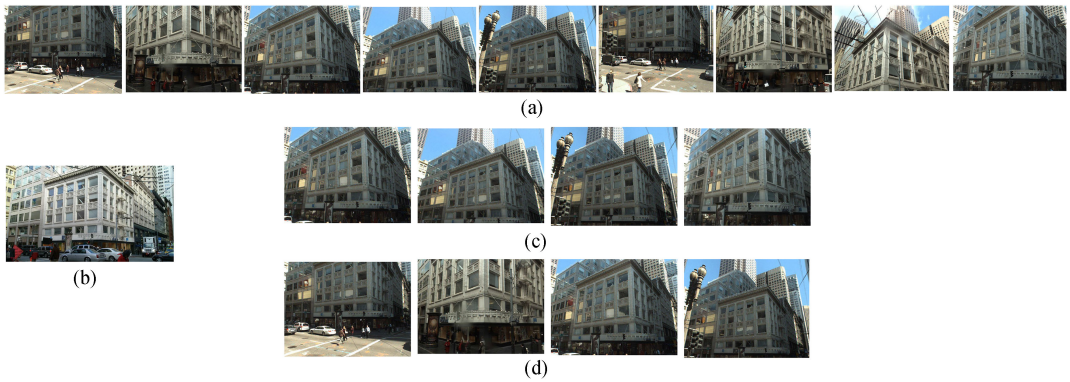


Fig. 6. (a) Sample set of images returned by retrieval pipeline for query image shown in (b). (c) Images selected by proposed method based on GMCP. (d) Images selected by finding images with highest number of inliers with query and distinct GPS-tag. Although images in two sets (c) and (d) look similar, only the set returned by GMCP let to convergence in the SFM pipeline.

Fig. 7. Sample set of images returned by retrieval pipeline in a case where neither GMCP nor distinct GPS-tag led to convergence. Images selected by GMCP and/or distinct (unique) GPS-tag are denoted as G and U, respectively, while images that were not selected are denoted as NS.

TABLE I
QUERY IMAGES AND CORRESPONDING FOUR MATCHES FOR SPARSE 3D CAMERA POSE RECONSTRUCTION

Query ID	PCI_ sp (best-matching) image ID				Query image	Transform: R, t_0, s	$E_{Total} = \sum_{i=1}^{n} \|e_i\|^2 (km^2)$
14	9276	9277	9279	9275	14		
3D cam positions	0.316227 -0.010382 1.740068	1.024494 -0.011895 1.0341255	2.418441 -0.000215 -0.439624	-0.391183 -0.000558 2.459574	1.148572 -0.000714 -1.207395	-0.892 -0.573 -0.137 0.403 -0.881 -0.655 -0.194 0.852 -0.747 t_0= [-2.697 -4.250 3.904] e+03 s = 0.0078	1.7399 e-06
26	4535	4534	4533	19128	26		
3D cam positions	-0.615103 -0.000913 2.005482	-1.189003 -0.001757 2.662914	-1.737672 -0.001110 3.313662	0.3774560 .040469 1.761960	-1.085159 -0.000942 -0.231594	0.602 -0.006 -0.678 0.297 0.056 0.681 0.740 0.044 0.274 t_0= [-2.697 -4.250 3.904] e+03 s = 0.0046	1.8707 e-07
320	13255	13256	13257	4819	320		
3D cam positions	0.003372 -0.005780 0.200270	0.733618 -0.035297 0.6269572	1.396704 -0.045083 -1.526042	1.332279 -0.063366 -1.287443	0.410683 0.349956 -3.633897	-0.27 0.01 -0.86 0.73 0.01 0.09 0.61 0.03 -0.49 t_0= [-2.697 -4.250 3.904] e+03 s = 0.0035	4.1369 e-07

TABLE II
GEO DISTANCE ERROR BETWEEN GROUND TRUTH AND ESTIMATED GPS TAGS FROM FIVE IMAGES

Geo Distance Error	Query Number						
	14	26	52	115	189	233	524
meters	1.81	7.37	5.1	1.29	0.97	2.2	1.89

to be unsuccessful. It is however possible to consider using three candidates for that query.

The coordinate transformation pipeline was found to converge with acceptable error for all successful cases of convergence in SFM. According to the Table I, the maximum residual error for transferring four points from left to right coordinate system was about three meters in the worst case.

It is worth mentioning that for all of the samples we got less than this level of error for residual error and it was an order of magnitude times smaller for most of the cases. The resulting estimation error of each query camera's GPS tag is shown in the Fig. 8.

As can be seen, the best result is obtained using the method with four dataset images. Moreover, the plot shows that 59.4% of the query estimated locations have an error of less than 5 meters and 32.6% have an error between 5 and 10 meters. For that scenario and for some samples shown in Table II, the ultimate localization error for most of the samples is less than three meters. This level of accuracy is not achievable for other two methods based on three dataset images. In fact for three images, PCA-based method is slightly better while it is inferior in a scenario with four images. Also, these results represent a marked improvement over the errors reported in [12], [2]. In [12] only errors less than 20 meters are reported while no statistics for errors less than five meters or between five and 10 meters is

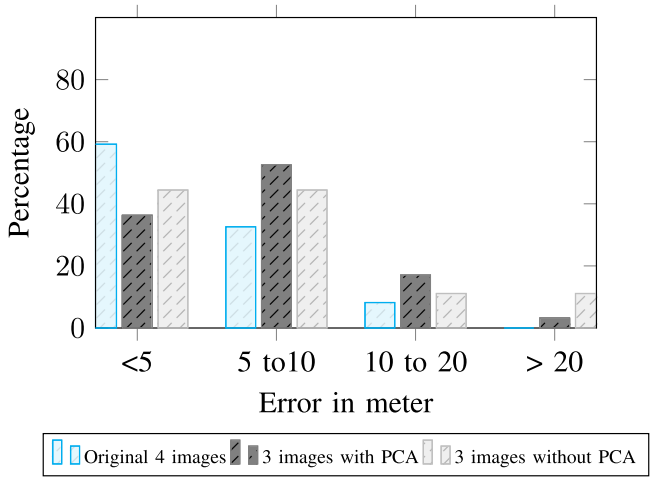


Fig. 8. Distribution of estimation error in meters of query camera's GPS tag using our proposed method for the cases of four original images (blue), three images without PCA (dark gray), three images with PCA (light gray). Localization error for about 59% of query images is less than 5 m using four images (blue).

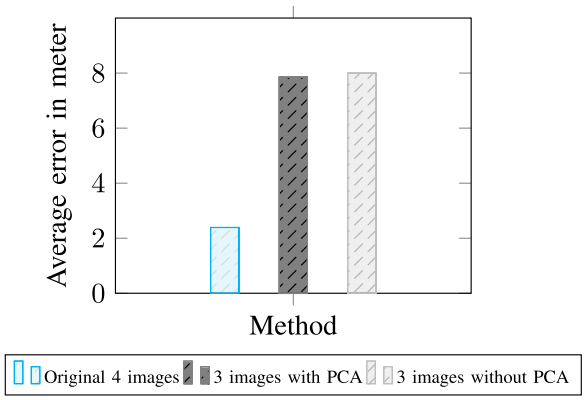


Fig. 9. Overall average of estimation error (in meters) of query camera's GPS tag using our proposed method for the cases of 4 original images (blue), 3 images with PCA (dark gray), 3 images without PCA (light gray).

presented. In [2] only 15% of the errors are less than five meters compared to 59.4% achieved with our approach. Achieving an error in the range of 20 meters was not our goal since this level of error can be obtained by just considering the location of the best match from retrieval for most of the queries. The average of estimation error of the query camera's GPS tag using our proposed method is shown in Fig. 9 for the cases of four original images (blue), three images with PCA (light gray) and three images without PCA (dark gray). It is important to note that we do not incur any increase in computational burden in our method with four or three images. This is because the time required for retrieving and re-ranking images for an arbitrary query is almost the same for all approaches. Also, image selection based on GMCP with only $N = 15$ nodes does not require a large amount of computation and adds up less than 10% to the time required for a single SFM. Moreover, the computational cost for coordinate transformation based on the proposed closed-form approach is even less than 1% of required time for a single SFM. So, the total running time is dependent mainly on the number of times SFM is executing. Unlike other mentioned research, we

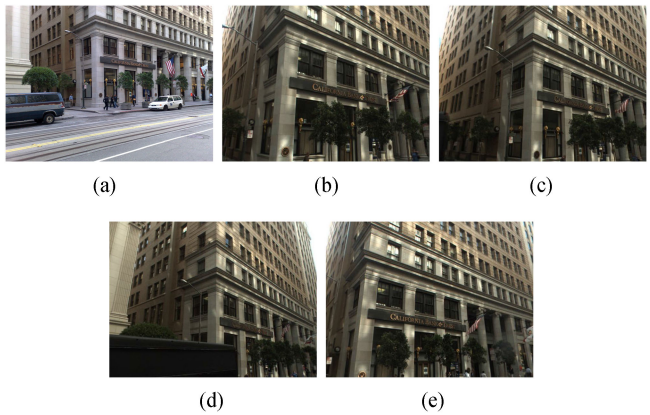


Fig. 10. Sample image set 1 of query and 4 best matching images considered in the position estimation shown in Fig. 11.

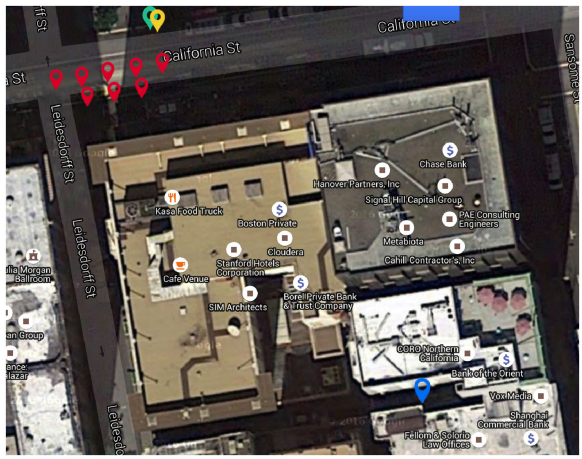


Fig. 11. Sample localization result for query image in set 1 in Fig. 10: Noisy query position from GPS (blue), position of best matches (red), actual (green) and estimated positions by proposed method (yellow).

run SFM only once that leads to significantly reduced running time. It is worth mentioning that considering four images instead of three images for the visualSFM process has a negligible effect on the processing time as discussed in [44] but reduces the mean squared reprojection error of the estimated five camera coordinates.

In order to show how our proposed method improved the localization, two samples are provided with more details. The two sample image sets considered here are shown in Figs. 10 and 12, and the positions of the retrieved images (more than four), are shown, respectively, in Figs. 11 and 13 with red icons. Also Figs. 10 and 12 show four images that are used in SFM for each query. To evaluate the performance of our proposed method, the noisy query position is shown in blue while the actual and estimated positions are shown in green and yellow, respectively. As can be seen, the actual and estimated positions are close, especially in Fig. 13 where the distance between the two is less than two meters.

Although we have considered prior knowledge of the position along with maximum GPS error for the San Francisco dataset, the localization errors for new cellphones are usually within 100 meters even in the worst case in cities such as San Francisco

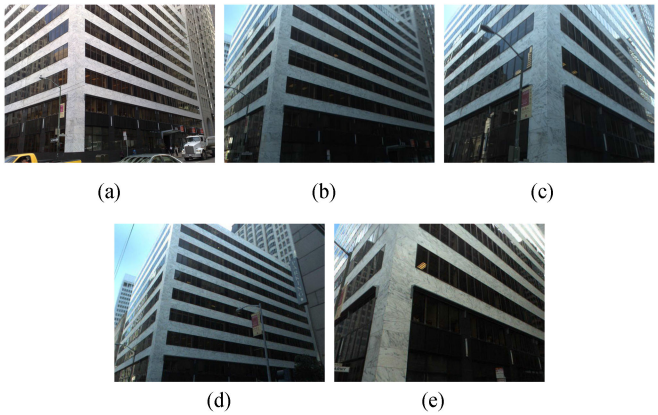


Fig. 12. Sample image set 2 of query and 4 best matching images considered in the position estimation shown in Fig. 13. (a) Query. (b) Match 1. (c) Match 2. (d) Match 3. (e) Match 4.

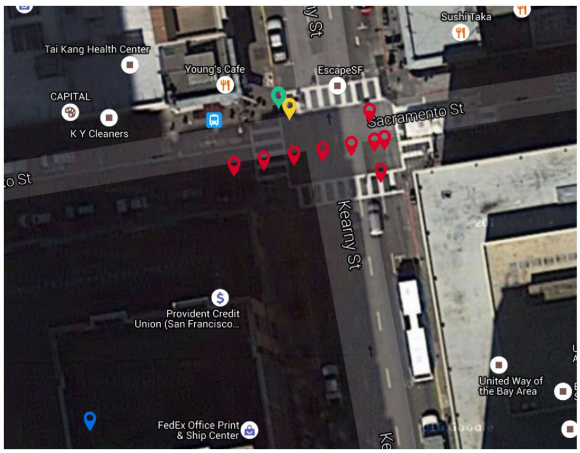


Fig. 13. Sample localization result for query image in set 2 in Fig. 12: Noisy query position from GPS (blue), position of best matches (red), actual (green) and estimated positions by proposed method (yellow).

or Chicago. So the retrieval engine can search in an even smaller region specified by the range of GPS error. By applying our proposed method this level of error would be reduced to a range of a couple of meters.

VII. Conclusion

In this paper, we first proposed a method to optimally select the best subset of images selected with the highest similarity to be used in reconstructing a 3D scene by using SFM. In order to compute the query location, we introduced a coordinate transformation between dataset images location in camera referenced coordinate system and their corresponding real-world locations. The advantage of this method is that the transformation parameters and consequently query location can be computed directly from the results obtained with only a single execution of SFM. Although four images from retrieval are employed, for most of the samples we showed that transformation parameters can also be computed with three images. Experimental results show that our approach is able to reduce the error in the estimates of query's GPS tag from more than 20 meters (distance between actual query position and best match) to less than five meters in a high percentage of the considered test cases which is suitable for localization application of interest to us. Also we observed that our proposed method will produce an improved performance (SFM convergence for a larger set of query images) if the original database has more images per location and a higher degree of overlap between images from similar locations. In future we will explore how a Convolutional Neural Network (CNN) can be employed as a core of image retrieval pipeline to improve the retrieval results.

References

[1] G. Schroth *et al.*, "Mobile visual location recognition," *IEEE Signal Process. Mag.*, vol. 28, no. 4, pp. 77–89, Jul. 2011.
[2] H. Liu, T. Mei, J. Luo, H. Li, and S. Li, "Finding perfect rendezvous on the go: Accurate mobile visual localization and its applications to routing," in *Proc. 20th ACM Int. Conf. Multimedia*, 2012, pp. 9–18. [Online]. Available: http://doi.acm.org/10.1145/2393347.2393357
[3] T. Guan, Y. He, J. Gao, J. Yang, and J. Yu, "On-device mobile visual location recognition by integrating vision and inertial sensors," *IEEE Trans. Multimedia*, vol. 15, no. 7, pp. 1688–1699, Nov. 2013.
[4] J. Sang *et al.*, "Interaction design for mobile visual search," *IEEE Trans. Multimedia*, vol. 15, no. 7, pp. 1665–1676, Nov. 2013.
[5] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.
[6] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool , "Speeded-up robust features (SURF)," *Comput. Vis. Image Understanding*, vol. 110, no. 3, pp. 346–359, 2008.
[7] S. Leutenegger, M. Chli, and R. Y. Siegwart, "BRISK: Binary robust invariant scalable keypoints," in *Proc. Int. Conf. Comput. Vis.*, 2011, pp. 2548–2555.
[8] G. Takacs *et al.*, "Rotation-invariant fast features for large-scale recognition," *Proc. SPIE Opt. Eng. Appl.*, vol. 8499, 2012, Art. no. 84991D.
[9] J. Zhang, A. Hallquist, E. Liang, and A. Zakhor, "Location-based image retrieval for urban environments," in *Proc. 18th IEEE Int. Conf. Image Process.*, 2011, pp. 3677–3680.
[10] M. Salarian, A. Manavella, and R. Ansari, "Accurate localization in dense urban area using Google Street View images," in *Proc. SAI Intell. Syst. Conf.*, 2015, pp. 485–490.
[11] M. Salarian and R. Ansari, "Improved image retrieval for efficient localization in urban areas using location uncertainty data," in *Proc. IEEE Int. Symp. Multimedia*, 2016, pp. 181–184.
[12] X. Xu, T. Mei, W. Zeng, N. Yu, and J. Luo, "AMIGO: Accurate mobile image geotagging," in *Proc. 4th Int. Conf. Internet Multimedia Comput. Serv.*, 2012, pp. 11–14. [Online]. Available: http://doi.acm.org/10.1145/2382336.2382340
[13] K. Vishal, C. V. Jawahar, and V. Chari, "Accurate localization by fusing images and GPS signals," in *Proc. IEEE Comput. Vis. Pattern Recognit. Workshops*, 2015, pp. 17–24.
[14] A. Roshan Zamir , S. Ardeshir, and M. Shah, "GPS-tag refinement using random walks with an adaptive damping factor," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 4280–4287.
[15] A. Zamir and M. Shah, "Accurate image localization based on Google Maps Street View," in *Proc. Eur. Conf. Comput. Vis.*, 2010, pp. 255–268.
[16] H. Li, Y. Wang, T. Mei, J. Wang, and S. Li, "Interactive multimodal visual search on mobile device," *IEEE Trans. Multimedia*, vol. 15, no. 3, pp. 594–607, Apr. 2013.
[17] C. Yan *et al.*, "Supervised hash coding with deep neural network for environment perception of intelligent vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 1, pp. 284–295, Jan. 2018.
[18] Y. Song, X. Chen, X. Wang, Y. Zhang, and J. Li, "6-DOF image localization from massive geo-tagged reference images," *IEEE Trans. Multimedia*, vol. 18, no. 8, pp. 1542–1554, Aug. 2016.
[19] F. X. Yu, R. Ji, and S.-F. Chang, "Active query sensing for mobile location search," in *Proc. 19th ACM Int. Conf. Multimedia*, 2011, pp. 3–12. [Online]. Available: http://doi.acm.org/10.1145/2072298.2072301
[20] M. Salarian, N. Ileiv, and R. Ansari, "Accurate image based localization by applying SFM and coordinate system registration," in *Proc. IEEE Int. Symp. Multimedia*, 2016, pp. 189–192.
[21] G. Reitmayr and T. W. Drummond, "Initialisation for visual tracking in urban environments," in *Proc. 6th IEEE ACM Int. Symp. Mixed Augmented Reality*, 2007, pp. 161–172.
[22] J. Sivic and A. Zisserman, "Video Google: A text retrieval approach to object matching in videos," in *Proc. 9th IEEE Int. Conf. Comput. Vis.*, 2003, pp. 1470–1477.

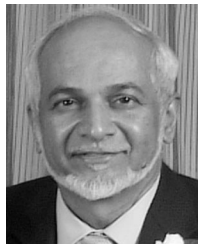
[23] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Object retrieval with large vocabularies and fast spatial matching," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2007, pp. 1–8.
[24] D. Nister and H. Stewenius, "Scalable recognition with a vocabulary tree," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2006, vol. 2, pp. 2161–2168.
[25] H. Jegou *et al.*, "Aggregating local image descriptors into compact codes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 9, pp. 1704–1716, Sep. 2012.
[26] H. Jegou, M. Douze, and C. Schmid, "Product quantization for nearest neighbor search," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 1, pp. 117–128, Jan. 2011.
[27] H. Jégou, M. Douze, C. Schmid, and P. Pérez, "Aggregating local descriptors into a compact image representation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2010, pp. 3304–3311.
[28] Y. Jing, M. Covell, D. Tsai, and J. M. Rehg, "Learning query-specific distance functions for large-scale web image search," *IEEE Trans. Multimedia*, vol. 15, no. 8, pp. 2022–2034, Dec. 2013.
[29] I. H. Witten, A. Moffat, and T. C. Bell, *Managing Gigabytes: Compressing and Indexing Documents and Images*. San Mateo, CA, USA: Morgan Kaufmann, 1999.
[30] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Lost in quantization: Improving particular object retrieval in large scale image databases," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2008, pp. 1–8.
[31] J. Knopp, J. Sivic, and T. Pajdla, "Avoiding confusing features in place recognition," in *European Conference on Computer Vision*. New York, NY, USA: Springer, 2010, pp. 748–761.
[32] P. Gronat, G. Obozinski, J. Sivic, and T. Pajdla, "Learning and calibrating per-location classifiers for visual place recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2013, pp. 907–914.
[33] Y. Chen, A. Dick, X. Li, and A. Van Den Hengel , "Spatially aware feature selection and weighting for object retrieval," *Image Vis. Comput.*, vol. 31, no. 12, pp. 935–948, 2013.
[34] P. Turcot and D. G. Lowe, "Better matching with fewer features: The selection of useful features in large database recognition problems," in *Proc. IEEE 12th Int. Conf. Comput. Vis. Workshops*, 2009, pp. 2109–2116.
[35] M. Park, K. Brocklehurst, R. T. Collins, and Y. Liu, "Deformed lattice detection in real-world images using mean-shift belief propagation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 10, pp. 1804–1816, Oct. 2009.
[36] A. Torii, J. Sivic, T. Pajdla, and M. Okutomi, "Visual place recognition with repetitive structures," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2013, pp. 883–890.
[37] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, Jun. 1981. [Online]. Available: http://doi.acm.org/10.1145/358669.358692
[38] P. H. Torr and A. Zisserman, "MLESAC: A new robust estimator with application to estimating image geometry," *Comput. Vis. Image Understanding*, vol. 78, no. 1, pp. 138–156, 2000.
[39] L. Moisan, P. Moulon, and P. Monasse, "Automatic homographic registration of a pair of images, with a contrario elimination of outliers," *Image Process. Line*, vol. 2, pp. 56–73, 2012.
[40] W. Min, C. Xu, M. Xu, X. Xiao, and B.-K. Bao, "Mobile landmark search with 3D models," *IEEE Trans. Multimedia*, vol. 16, no. 3, pp. 623–636, Apr. 2014.
[41] N. Katoh, T. Ibaraki, and H. Mine, "An algorithm for finding K minimum spanning trees," *SIAM J. Comput.*, vol. 10, no. 2, pp. 247–255, 1981.
[42] M. Fischetti, H. W. Hamacher, K. Jørnsten, and F. Maffioli, "Weighted k-cardinality trees: Complexity and polyhedral structure," *Networks*, vol. 24, no. 1, pp. 11–21, 1994.
[43] A. R. Zamir, A. Dehghan, and M. Shah, "GMCP-tracker: Global multi-object tracking using generalized minimum clique graphs," in *Computer Vision–ECCV 2012*. New York, NY, USA: Springer, 2012, pp. 343–356.
[44] C. Wu, S. Agarwal, B. Curless, and S. M. Seitz, "Multicore bundle adjustment," in *Proc. Comput. Vis. Pattern Recognit.*, 2011, pp. 3057–3064.
[45] B. K. P. Horn, H. Hilden, and S. Negahdaripour, "Closed form solution of absolute orientation using orthonormal matrices," *Opt. Soc. Amer. A*, vol. 5, pp. 1127–1135, 1988.
[46] D. Chen *et al.*, "City-scale landmark identification on mobile devices," in *Proc. Comput. Vis. Pattern Recognit.*, 2011, pp. 737–744.
[47] T. Sattler, T. Weyand, B. Leibe, and L. Kobbelt, "Image retrieval for image-based localization revisited," in *Proc. Brit. Mach. Vis. Conf.*, 2012, vol. 1, pp. 1–10.

Mahdi Salarian received the M.Sc. degree in electrical engineering from the University of Mazandaran, Babol, Iran, in 2006. He is currently working toward the Ph.D. degree with the Department of Electrical and Computer Engineering, University of Illinois at Chicago, Chicago, IL, USA.

From 2007 to 2012, he held a lecturing position with the Electrical and Computer Engineering Department, Azad University. His research interests include computer vision, image processing, multimedia, and image retrieval.

Nick Iliev received the M.S. degree in electrical engineering from the University of California—Irvine, Irvine, CA, USA. He is currently working toward the Ph.D. degree in electrical engineering with the University of Illinois at Chicago, Chicago, IL, USA. His research interests include neuromorphic signal processing algorithms and VLSI architectures for sensor localization, speaker recognition, energy systems sensing, and microrobotic systems.

Ahmet Enis Çetin (F'10) received the B.Sc. degree from Middle East Technical University, Ankara, Turkey, and the Ph.D. degree from the University of Pennsylvania, Philadelphia, PA, USA, in 1987. From 1987 to 1989, he was an Assistant Professor with the University of Toronto, Toronto, ON, Canada. From 1989 to 2016, he was on the faculty of Bilkent University, Ankara, Turkey. He is on leave from Bilkent University. He is currently a Research Professor with the University of Illinois at Chicago, Chicago, IL, USA. His research interests include signal, data, image, and video processing. He is the coauthor of the book *Methods and Techniques for Fire Detection: Signal, Image and Video Processing Perspectives* (Academic Press, 2016). He is the Editor-in-Chief for *Signal, Image and Video Processing*, Springer.



Rashid Ansari (F'99) received the B.Tech. and M.Tech. degrees in electrical engineering from the Indian Institute of Technology Kanpur, Kanpur, India, in 1975 and 1977, respectively, and the Ph.D. degree in electrical engineering and computer science from Princeton University, Princeton, NJ, USA, in 1981. He is currently a Professor and Head with the Department of Electrical and Computer Engineering, University of Illinois at Chicago (UIC), Chicago, IL, USA. Before joining UIC, he served as a Research Scientist with the Bell Communications Research and on the faculty of electrical engineering with the University of Pennsylvania. His research interests include signal processing and communications, with recent focus on image and video analysis, multimedia communication, and medical imaging.

In the past, he was an Associate Editor for the IEEE TRANSACTIONS ON IMAGE PROCESSING, the IEEE SIGNAL PROCESSING LETTERS, and the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS. He was a member of the editorial board of the *Journal of Visual Communication and Image Representation*. He was a member of the Digital Signal Processing Technical Committee of the IEEE Circuits and Systems Society and a member of the Image, Video, and Multidimensional Signal Processing Technical Committee. He was a member of the organizing and program committees of several past IEEE conferences. He was on the organizing and executive committees of the Visual Communication and Image Processing (VCIP) conferences and was the General Chair of the 1996 VCIP Conference.