Image Geo-Localization Based on Multiple Nearest Neighbor Feature Matching Using Generalized Graphs

Amir Roshan Zamir, *Member, IEEE* and Mubarak Shah, *Fellow, IEEE*

Abstract—In this paper, we present a new framework for geo-locating an image utilizing a novel multiple nearest neighbor feature matching method using Generalized Minimum Clique Graphs (GMCP). First, we extract local features (e.g., SIFT) from the query image and retrieve a number of nearest neighbors for each query feature from the reference data set. Next, we apply our GMCP-based feature matching to select a single nearest neighbor for each query feature such that all matches are globally consistent. Our approach to feature matching is based on the proposition that the first nearest neighbors are not necessarily the best choices for finding correspondences in image matching. Therefore, the proposed method considers multiple reference nearest neighbors as potential matches and selects the correct ones by enforcing consistency among their global features (e.g., GIST) using GMCP. In this context, we argue that using a robust distance function for finding the similarity between the global features is essential for the cases where the query matches multiple reference images with dissimilar global features. Towards this end, we propose a robust distance function based on the Gaussian Radial Basis Function (G-RBF). We evaluated the proposed framework on a new data set of 102k street view images; the experiments show it outperforms the state of the art by 10 percent.

Index Terms—Geo-location, image localization, Generalized Minimum Clique Problem (GMCP), generalized minimum spanning tree (GMST), feature matching, multiple nearest neighbor feature matching, feature correspondence, generalized graphs

1 Introduction

Recently, large scale image geo-localization methods which employ techniques similar to image matching have attracted much interest [1], [2], [3], [4]. In these methods, it is assumed that a reference data set consisting of geo-tagged images is available. Then, the problem is to estimate the geo-location of a query image by finding its matching reference images. There are several known methods in this context: Schindler et al. [3] developed a method for city scale localization based on the bag of visual words model [5] using a data set of street side images. They proposed a greedy algorithm for improving the accuracy of searching a vocabulary tree. Knopp et al. [4] presented an approach to generating a codebook which discards the words which are identified to be non-discriminative for geo-localization purposes. Hays and Efros [1] developed a method for extracting coarse geographical information from a query image using a data set of Flickr images. We proposed a framework [2] which utilized Google Street View images as the reference data set; a feature pruning method which incorporates geo-spatial information was employed to discover incorrectly matched features. Sattler et al. [6] developed a framework similar to [7] for identifying 2D-to-3D correspondences between the query and the reference data set with a large number of user shared images. In [8], they presented an efficient method for the same purpose based on both 2D-to-3D and 3D-to-2D matching.

Most of these methods only utilize local features which ignore the global context of the image and make them inherently prone to mismatches. Therefore, several procedures for embedding contextual information in local descriptors have been developed. Mortensen et al. [9] proposed an extension to SIFT by augmenting it with global curvilinear shape information. Mikolajczyk et al. [10] leveraged local feature and edge based information along with a geometric consistency verification for object class recognition. Cao et al. [11] present an approach similar to [9] to make SIFT affine invariant. Hao et al. [12] and Zhang et al. [13] proposed two methods for incorporating the geometry of the scene in image matching using bundles of local features generally termed "visual phrases." In addition, a number of approaches for dealing with the repetitive visual patterns in the data sets have been developed. Such patterns, e.g., recurrent architectural structures, exacerbate the susceptibility of local features to mismatches caused by ignoring the global context. Torii et al. [14] proposed a weight modification method in order to have a better representation of the repeated structures. Jegou et al. [15] developed a method which removes multiple matches along with reducing the weights of repeated features in a bag of visual words framework.

In this paper, we propose an approach to image localization which finds one or a few strongly matching reference images to a query by robustly discovering local feature correspondences. In order to address the weakness of local features in leveraging the global context, our method considers multiple reference nearest neighbors

• *The authors are with the Center for Research in Computer Vision, University of Central Florida, 4000 Central Florida Blvd., Harris Corporation Engineering Center, Orlando, FL 32816.*
E-mail: aroshan@cs.ucf.edu, shah@eecs.ucf.edu.

Manuscript received 13 Nov. 2012; revised 13 Dec. 2013; accepted 27 Dec. 2013. Date of publication 12 Jan. 2014; date of current version 10 July 2014.
Recommended for acceptance by D. Forsyth.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TPAMI.2014.2299799

(NN) as the potential matches for each query feature and selects the correct ones by examining the consistency among their global features. The utilized global consistency is based on the following proposition: *parent images of the reference features matched to a particular query image should have similar global features as they are expected to be of the same scene.* We performed our experiments using different types of global features, such as GIST, color histogram, and image geo-tag; all were shown to improve the performance while the geo-tags yielded the best overall results.

We use the Generalized Minimum Clique Problem (GMCP) [16] at the core of our feature matching method. GMCP is useful in situations where there are multiple potential solutions for a number of subproblems, as well as a global criterion among the subproblems to be satisfied. In our framework, each subproblem is matching a query feature to the reference features, the potential solutions are the NNs, and the global criterion is the consistency of global features of the NNs. Therefore, we utilize GMCP in performing our multiple nearest neighbor feature matching, and a voting scheme on the matched features is employed to identify the strongly matching reference image(s) and estimate the geo-location.

Despite the shared similarities in the high level goal, the current methods for leveraging the global context are fundamentally different from ours in four aspects: 1- Unlike most of the existing approaches which capture one particular type of contextual information [9], [13], [17], our method is capable of leveraging arbitrary global features such as the global color histograms or geo-location. 2- We do not embed the global context in the local feature vector. Therefore, the space in which local and global features are matched are kept separate, and different metrics can be used for each. 3- Our method matches all the features of one image simultaneously which essentially means they contribute to each others' match. This is different from the existing methods which perform feature matching on an individual basis [9], [10], [11]. 4- A number of methods perform geometric verification by fitting the fundamental matrix to a set of initially discovered correspondences in order to remove the incorrect matches [18], [19]. Such methods are different from ours as we use global features in establishing the initial correspondences rather than pruning a set of already found correspondences. Moreover, the type of contextual information leveraged in such methods is limited to the spatial geometry of features.

Robust estimation methods, such as RANSAC, are commonly used in computer vision for performing a robust model estimation where the input data includes outliers. Such methods were adopted for discovering feature correspondences [20] and have been robustified by modified cost functions [21], [22]. However, despite the similarity in the overall goal, there is a difference between such methods and ours: we nominate multiple NNs as the potential matches for a query feature. By definition, GMCP enforces picking *one and only one* candidate for each query features, whereas in the basic RANSAC formulation, the aim is to select the inlier correspondences given a set of *one-to-one* matches.

Image matching methods which involve clustering of features, such as the bag of visual words model, have been widely used because of their efficiency in dealing with a large corpus of data [3], [18]. However, they have the disadvantage of losing information in the quantization step. The quantization loss becomes critical for the data sets which possess extensively repeated features. Several methods, such as soft assignment of words [23], were developed in order to alleviate this problem. However, such methods lose their superior performance on data sets where the repetition and similarity of features happen substantially. One example of such data sets are images of urban areas, as most of the man-made structures have similar architectural features [2]. The issue of excessive quantization loss is not applicable to our method as the matching is performed on raw local features. In addition, the proposed method is well-suited for being coupled with fast and approximate NN search methods, e.g., [24], [25], to handle the large amount of data in a timely manner; this is because our approach does not strictly assume the first retrieved NN is the correct one.[1] In fact, GMCP is capable of identifying the correct NN as long as it appears among the top retrieved NNs which can partially alleviate the suboptimal performance of the approximate NN search methods.

Our reference data set consists of over 102k Google Street View images which cover the downtown and neighboring areas of Pittsburgh, PA; Orlando, FL and part of Manhattan, NY. We tested the proposed approach using an unconstrained test set of 644 GPS-tagged images uploaded by random users to Flickr, Panoramio and Picasa. The data will be made available to the public for further research.

The main contributions of this paper can be summarized as

- A multiple nearest neighbor feature matching method based on Generalized Minimum Clique Graphs.
- A novel framework for incorporating both local and global features in image geo-localization.
- A new data set of high resolution street view images.

The rest of the paper is organized as follows: the proposed framework is described in Section 2 with our pruning method in Section 2.1, GMCP-based feature matching in Section 2.2 and location estimation in Section 2.3. The algorithm for solving GMCP and experimental results are provided in Sections 3 and 4, respectively.

2 Geo-Localization by Image Search

We preprocess the reference data set by computing a set of local features (in our implementation SIFT) from each image. We aggregate these features of all the reference images and organize them in a k-means tree [25]. We refer to the extracted local features, their corresponding reference images, and the built tree as *reference features*, *parent images*, and *reference tree*, respectively. Additionally, we find a

1. Throughout the paper, what we refer to by "correct" NN is the one which indeed matches the query NN, and not the first NN in terms of the descriptor distance. The correct NN does not necessarily have to be the first NN.

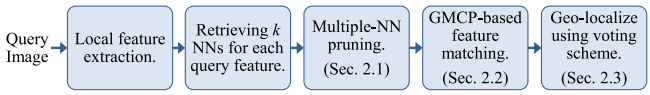

Fig. 1. Block diagram of the proposed Image Geo-localization Method.

global feature, e.g., color histogram or GPS location, for each reference image.

The block diagram of our framework for geo-locating a query image is shown in Fig. 1. First, we extract local features from the query image; we refer to them as *query features*. We search the reference tree using the query features and retrieve k nearest neighbors for each query feature. Next, we apply our multiple-nearest neighbor pruning (Section 2.1) to coarsely remove the query features which do not have distinctive NNs. In the next step, we employ a robust function for computing the distance between global features which is particularly essential when multiple reference images with dissimilar global features match the query image (Section 2.2.2). Unlike the traditional feature matching methods, such as the k nearest neighbor classifier which performs voting using all of the k NNs, we consider the k NNs as potential matches for their query feature and identify the correct one using the GMCP-based feature matching method (Section 2.2). Lastly, a voting scheme on the matched features is employed to find the reference image which most strongly matches the query. We use the location of the strongest match as an estimation of the location of the query image (Section 2.3).

Detailed explanation of each step is provided in the rest of this section.

2.1 Multiple-Nearest Neighbor Pruning

Let M be the number of local features detected in the query image. Let v_n^j denote the nth NN for the jth query feature where $n \in \mathbb{N} : 1 \leq n \leq k$ and $j \in \mathbb{N} : 1 \leq j \leq M$.

Many of the interest points found in the query image, such as the ones detected on foliage, ground or moving objects, do not convey any useful information for geo-localization purposes. It will be helpful if such features can be coarsely identified and removed prior to performing the feature matching. For this purpose, we utilize the following pruning method which is based on examining how distinctive the first and $(k+1)$th NN are:

$$\begin{cases} \text{remove } q^i, & \text{if } \frac{\|q^i - \zeta(v_1^i)\|}{\|q^i - \zeta(v_{k+1}{}^i)\|} > 0.8 \\ \text{retain } q^i, & \text{otherwise,} \end{cases} \tag{1}$$

where $\zeta(.)$ represents an operator which returns the local feature descriptor of the argument node. q^i is defined as the local descriptor of the ith query feature, and $\|.\|$ represents the distance between the features. Equation (1) states the ith query feature should be pruned if its first and $(k+1)$th NNs are more than 80 percent similar. This formulation is consistent with our multiple-NN feature matching scenario as we assume the correct match is among the top k NNs. Therefore, we disregard the top k NNs and compare the first NN to the $(k+1)$th one. If the first NN is not distinctive, even compared to the $(k+1)$th NN, the corresponding query feature is pruned since it is expected to be uninformative. The threshold value of 0.8 is empirically found to be optimal for comparing SIFT features by Lowe [26]. Note that the difference between Lowe's criteria [26] and the criteria in Equation (1) is that we utilize multiple NNs instead of using the first two only, which makes our pruning consistent with our multi-NN formulation.

2.2 Feature Matching Using Generalized Minimum Clique Graph

Let L be the number of local features surviving the pruning step. We define the graph $G = (\mathbf{V}, E, \varpi, w)$, where $\mathbf{V}$, E, ϖ and w denote the set of nodes, edges, node costs and edge weights, respectively. The set of nodes, $\mathbf{V}$, is divided into L disjoint clusters. Each query feature point is represented by one cluster, and the nodes therein represent the corresponding k nearest neighbors. C_i, where $i \in \mathbb{N} : 1 \leq i \leq L$, denotes the ith query feature (≡cluster), and v_m^i denotes the mth candidate (≡node) for the ith query feature. The edges are defined as $E = \{(v_m^i, v_n^j) | i \neq j\}$ which signifies all the nodes in G are connected as long as they do not belong to the same cluster. We define the node cost, $\varpi : \mathbf{V} \rightarrow \mathbb{R}^+$, as

$$\varpi(v_m^i) = \|q^i - \zeta(v_m^i)\|. \tag{2}$$

The node cost specifies how similar the local features of v_m^i and its corresponding query features are. Edge weight, $w : E \rightarrow \mathbb{R}^+$, is defined as

$$w(v_m^i, v_n^j) = \|\rho(v_m^i) - \rho(v_n^j)\|, \tag{3}$$

where $\rho(.)$ represents an operator which returns the global descriptor of the parent image of the argument node. The edge weight, $w(v_m^i, v_n^j)$, is a measure of similarity between the nodes v_m^i and v_n^j in terms of the global features of their parent images. Low values for an edge weight and its node costs signify a high global consistency between corresponding nodes and vice versa. $\|.\|$ in Equations (2) and (3) denotes the distance between the local and global features of the argument nodes, respectively; however, the type of distances employed in these two equations do not have to be the same.

The task of matching the query features to the reference features requires identifying the correct NN for each one. Therefore, a feasible solution to this problem can be represented by a subgraph of G in which one node (≡NN) is selected from each cluster(≡set of nominated NNs for one query feature). Such a subgraph, $G_s = (\mathbf{V}_s, E_s, \varpi_s, w_s)$, consists of a set of nodes with the general form $\mathbf{V}_s = \{v_a^1, v_b^2, v_c^3, \ldots\}$ which indicates the ath node from first cluster, bth one from second cluster, and so on are selected to be included in $\mathbf{V}_s$. By definition, $E_s = \{E(p,q) | p, q \in \mathbf{V}_s\}$, $\varpi_s = \{\varpi(p) | p \in \mathbf{V}_s\}$, and $w_s = \{w(p,q) | p, q \in \mathbf{V}_s\}$. We use $\mathbf{V}_s$ to denote a feasible solution hereafter since the set of nodes $\mathbf{V}_s$ is essentially enough to form G_s. The cost of the feasible solution $\mathbf{V}_s$ is defined as

$$C(\mathbf{V}_s) = \frac{1}{2}\sum_{i=1}^{L}\sum_{\substack{j=1,\\ j\neq i}}^{L}\left(\frac{1}{2}\alpha\overbrace{\left(\varpi(\mathbf{V}_s(i)) + \varpi(\mathbf{V}_s(j))\right)}^{\text{local features}} + (1-\alpha)\underbrace{w(\mathbf{V}_s(i), \mathbf{V}_s(j))}_{\text{global features}}\right), \tag{4}$$

which is the cost of the complete graph induced by the nodes in $\mathbf{V}_s$. $\mathbf{V}_s(i)$ denotes the ith element of $\mathbf{V}_s$, and α is the mixture constant that ranges between 0 and 1 and balances the contribution of local and global features. A larger α corresponds to more contribution from local features in the overall cost and vice versa; $\alpha = 0.5$ corresponds to equal contributions from both features. Equation (4) (note the constants) is defined in a way that the number of summed terms corresponding to the nodes and edges are always equal. Hence, the balance between the contribution of local and global features to the cost does not change with L.

Equation (4) assigns a cost to a feasible solution utilizing both local and global features. This is done by incorporating the agreement between the global features of the parent images of reference features. Therefore, the potential wrong matches resulting from the limited scope of local features are minimized. By finding the feasible solution with the minimal cost, i.e., $\text{argmin}_{\mathbf{V}_s} C(\mathbf{V}_s)$, the subset of NNs with the highest agreement is found. In the following section, we explain that the definition of Generalized Minimum Clique Graph ideally fits the formulation of our problem and can be used for solving the aforementioned optimization task.

2.2.1 Generalized Minimum Clique Problem

Generalized Graphs, also known as Generalized Network Design Problems [16], are a category of graph theory problems which are based on generalizing the standard subgraph problems. The generalization is done by extending the definition of a node to a cluster of nodes. For example, in the standard Traveling Salesman Problem (TSP) the objective is to find the minimal cycle which visits all the nodes exactly once. In the *Generalized* Traveling Salesman Problem, the nodes of the input graph are grouped into disjoint clusters; the objective is to find the minimal cycle which connects all the clusters while exactly one node from each is visited [16].

Similarly, in the Generalized Minimum Clique Problem the vertices of the input graph are grouped into disjoint clusters. As shown in Fig. 2, the objective is to find a subset of the nodes that includes exactly one node from each cluster while the cost of the complete graph that the subset forms is minimized [16]. A similar formulation is utilized to solve the Frequency Assignment Problem [27] in telecommunications. It has been utilized for maximizing the number of comparisons between human detections in different video frames in order to perform data association as well [28].

The input to GMCP with vertex cost is defined as the graph $G = (\mathbf{V}, E, \varpi, w)$ where $\mathbf{V}$, E, ϖ and w represent the set of nodes, edges, node costs and edge weights. $\mathbf{V}$ is divided into L disjoint clusters, i.e., $\mathbf{C}_i \cap \mathbf{C}_j = \emptyset$ $(1 \leq i \neq j \leq L)$ and $\mathbf{C}_1 \cup \mathbf{C}_2 \cup \ldots \cup \mathbf{C}_f = \mathbf{V}$. A feasible solution of GMCP is denoted by a subgraph $G_s = (\mathbf{V}_s, E_s, \varpi_s, w_s)$,

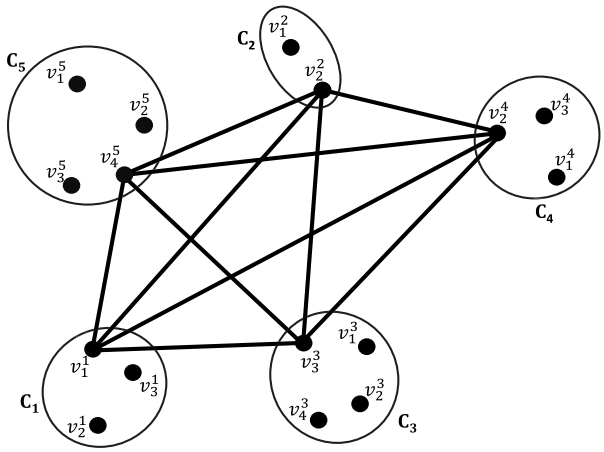

Fig. 2. An example GMCP. A feasible solution is shown where one node from each cluster is selected. The complete subgraph, G_s, which the selected nodes form is shown using the edges.

where $\mathbf{V}_s$ and ϖ_s denote a subset of $\mathbf{V}$ which includes only one node from each cluster and its corresponding node costs, respectively. E_s and w_s are the subset of E which V_s induces and the corresponding edge weights. The cost of a feasible solution is defined as the sum of all the edge weights and node costs along the solution subgraph. Note that the subgraph G_s is complete, making any feasible solution of GMCP a clique.

As can be inferred from the formulation of our multiple NN feature matching problem, GMCP can be essentially used for solving the same optimization problem. Therefore, by solving GMCP for the graph G, the optimal solution which has the most agreement in terms of global and local features will be found via:

$$\hat{\mathbf{V}}_s = \underset{\mathbf{V}_s}{\text{argmin}}\ C(\mathbf{V}_s). \tag{5}$$

Note that our GMCP-based method differs from basic graphical models in several aspects. Our input graph and feasible solutions are complete as we consider the relationships among all possible pairs of local features. This makes our formulation different from the graphical models which have specific assumptions on the structure of the graph, e. g., being acyclic. Additionally, a graphical model equivalent to our input graph would include a large number of loops as it would have to be complete; the condition under which the inference methods similar to belief propagation [29] converge for graphs with loops is still unknown [30]. On the contrary, we employ a combinatorial approach to solving our optimization problem whose performance does not deteriorate by including loops in the input graph. The details of how to solve GMCP are discussed in Section 3.

Fig. 3 demonstrates the process of feature matching using GMCP.[2] (a) shows a query image and two best matching reference images on the left and right, respectively. Discovered correspondences by GMCP are shown in green. (b) shows all the nodes in $\mathbf{V}$ in the global feature

2. In Figs. 3, 4 and 6 the nodes which are located exactly on the same spot in the global feature space, i.e., belong to the same reference image, are shown slightly apart in order to demonstrate the density properly.

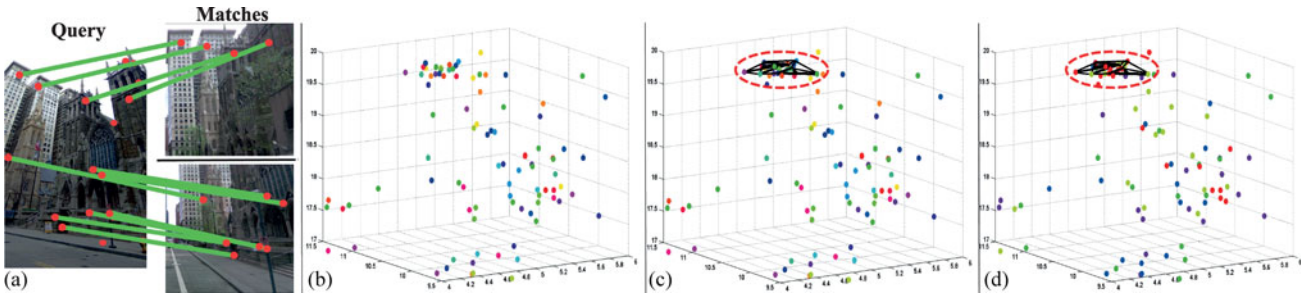

Fig. 3. Feature matching using GMCP; (a) A query image and matched reference images are shown on the left and right, respectively. Found correspondences are shown by the green lines. (b) All the nodes in $\mathbf{V}$ are shown in three-dimensional global feature space. Each node represents one NN while the color coding indicates cluster membership. (c) Same as (b) while the black lines indicate GMCP edges. (d) Same as (c) while the color coding shows the rank of the nearest neighbor. red = first, yellow = second, green = third, blue = fourth, magenta = fifth.

space. In this example, a 60-dimensional RGB color histogram is used as the global feature, and the dimensionality is reduced to 3 using PCA for illustration purposes. Membership to GMCP clusters, $\mathbf{C}_i$, is color coded meaning each color represents one value of i. (c) shows the subset of nodes included in $\hat{\mathbf{V}}_s$ depicted by the red contour. The black edges are the ones included in the subgraph $\hat{G}_s$. (d) illustrates the same plot as (c) except that the color coding represents the rank of each node when retrieved based on the local features. Red, yellow, green, blue and magenta represent first, second, third, fourth and fifth NN, respectively. A considerable number of the nodes included in $\hat{\mathbf{V}}_s$ are not marked in red which signifies the nodes with the most consistent global features are not necessarily the first NNs. Also, it is apparent in (c) and (d) that the selected nodes belong to a tight area in the feature space which indicates they share similar global features.

Another example is demonstrated in Fig. 4a. The utilized global feature is a 60-dimenstional RGB color histogram reduced to two dimensions using PCA for illustration. Each node in (a) represents one NN included in $\mathbf{V}$ shown in the global feature space. The inlier and outlier NNs are shown in red and blue, respectively.[3] The inlier NNs are the ones which belong to one of the reference images that actually matches the query image, and the outliers are the ones which do not belong to any of the matching reference images. As apparent in the figure, the global features of all the inlier NNs are similar as they are adjacent in the global feature space.

However, we commonly observe cases where the global features of matching reference images are not similar and consequently form disjoint groups in the global feature space. This dissimilarity is mainly due to variations, such as different imaging conditions or diverse camera poses, to which most of the existing global features are not invariant. One case is shown in Fig. 4b where the building in the query image is visible from two distinct locations, and the reference data set includes images taken at both of these locations. The difference in viewpoint between the query and the first and second groups of matching images is less than 30 degrees which causes the local feature to nominate NNs from both groups of images. However, the disagreement in viewpoint and imaging conditions will cause the images in the two different groups to have dissimilar global features. Thus, the inlier NNs are observed in two disjoint groups in the global feature space as shown in Fig. 4b.

The method explained earlier in this section was based upon the assumption that the global features of all the inlier NNs should be similar, i.e., they should form one joint group of inliers. In Section 2.2.2 we argue that the GMCP-based method fails to identify all of the inliers when multiple disjoint groups exist. We address

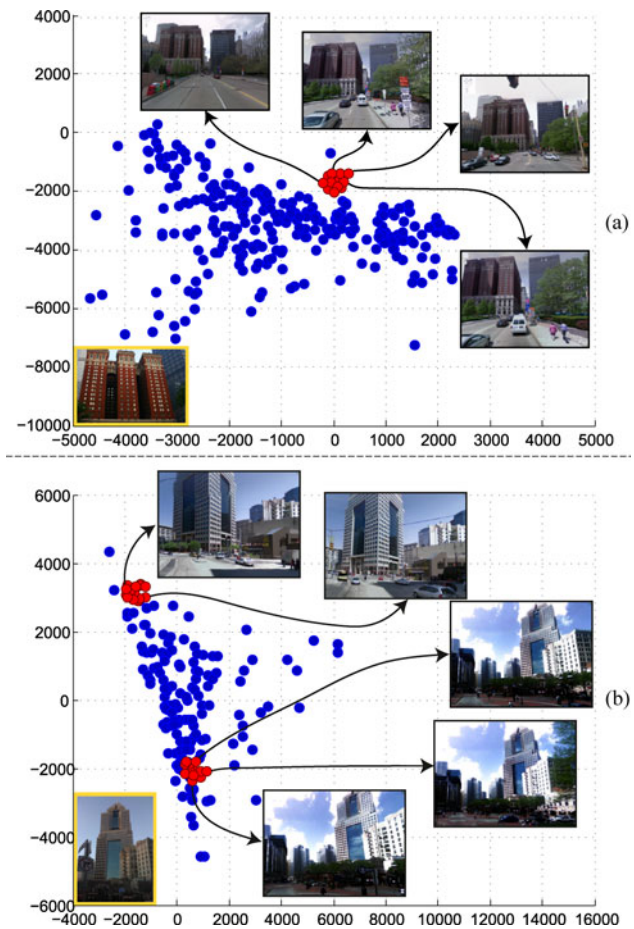

Fig. 4. (a) and (b) All the nodes in $\mathbf{V}$ shown in two-dimensional global feature space for two sample queries. Outlier and inlier NNs are illustrated in blue and red, respectively. The query images are shown with the yellow border. A subset of the matching reference images are shown linked to their corresponding nodes. (a) A case with one group of matching reference images. (b) A case with two groups of matching reference images with dissimilar global features.

3. The same applies to Fig. 6.

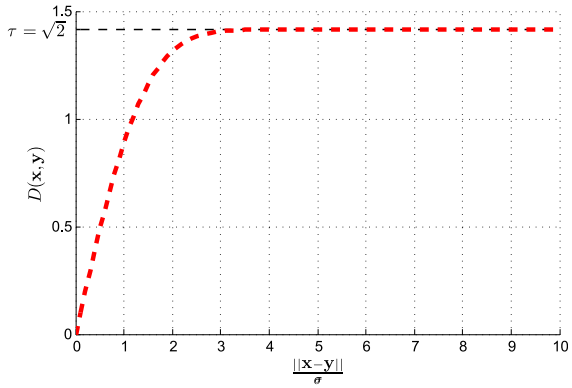

Fig. 5. The robust distance function D. It has the characteristic of damping the large values and boosting the short ones.

this issue by using a robust distance function for the global features.

2.2.2 Robustification of the Global Features' Distance Function

The existence of disjoint groups prevents the GMCP-based method from identifying all of the inliers. That is because a feasible solution which includes inlier nodes from several disjoint groups will have a high cost due to the considerable distance between these groups. We solve this problem by using the following robustified metric for computing the distance between the global features:

$$D(\mathbf{x},\mathbf{y}) = \sqrt{2 - 2e^{-\frac{\|\mathbf{x}-\mathbf{y}\|^2}{2\sigma^2}}}. \tag{6}$$

$\|\mathbf{x}-\mathbf{y}\|$ and $D(\mathbf{x},\mathbf{y})$ denote the original (e.g., Euclidean) and robustified distance between the two vectors $\mathbf{x}$ and $\mathbf{y}$, respectively.

The function in Equation (6) is plotted in Fig. 5. The characteristic of the distance function D is boosting the short distances and damping the large ones. The distances that are significantly larger than σ will be mapped to the constant value τ $(=\sqrt{2})$. In the context of our problem, it means the distances will contribute to the cost functions equally if they are significantly larger than a certain value which is determined by σ. This trait causes the intra-group distances to matter more than inter-group ones after robustification. This enables the optimization function of Equation (5) to find the tight groups of global features rather than getting bewildered by the excessive cost the outliers contribute.

Our robustification can be viewed as finding the distances between the global features in a space transformed using a Gaussian Radial Basis Function (G-RBF) kernel. This is because G-RBF kernel is defined as $k_G(\mathbf{x},\mathbf{y}) = e^{-\frac{\|\mathbf{x}-\mathbf{y}\|^2}{2\sigma^2}}$. Assuming the employed norm is ℓ_2, the distance between two vectors in a projected space transformed using an arbitrary kernel k equals $\sqrt{k(\mathbf{x},\mathbf{x}) + k(\mathbf{y},\mathbf{y}) - 2k(\mathbf{x},\mathbf{y})}$ [31]; plugging G-RBF kernel in this function yields Equation (6).

In the experiments section, we will empirically show that the distance function defined in Equation (6) yields the best results compared to ℓ_1 distance, Squared, Linear, and Huber loss functions. However, any function which has a form similar to the curve shown in Fig. 5 is expected to give similar robustification effect.

To summarize, we form the input to GMCP using the distance function D for finding the distances between global features. This is done by using the Equation (6) in the Equation (3):

$$w_D(v_m^i, v_n^j) = \sqrt{2 - 2e^{-\frac{\|\rho(v_m^i)-\rho(v_n^j)\|^2}{2\sigma^2}}}, \tag{7}$$

which provides the edge weights robustified by the metric D. The value of σ is set in a way that the distance between two inlier nodes in one group is unlikely to be significantly larger than σ. Additionally, a distance substantially larger than σ should be likely to involve either an outlier node or two groups of disjoint inliers. Therefore, σ should be set to the expected distance between the global features of two images of the same scene. It is a fixed value determined based on the type of the global feature and does not need to be tuned for each query.

Fig. 6 shows the impact of robustification for two sample cases. Right and left columns show examples with two and three disjoint groups of matching reference images. (a) shows all the nodes in $\mathbf{V}$ where inliers and outliers are shown in red and blue, respectively. Green nodes in (b) represent the selected nodes by GMCP, i.e, $\hat{\mathbf{V}}_s$, without robustification (using ℓ_2 distance) which indicates it failed to identify all the inliers from different groups. (c) illustrates the results using D, and it signifies that the robustification enables GMCP to include the inliers from all of the disjoint groups.

Generalized minimum spanning tree (GMST). One potential way of dealing with the problem of disjoint groups of matching images is leveraging a linkage mechanism in feature matching. Linkage based methods, such as single-linkage clustering [32], are commonly built on the following general definition: *the distance between two groups of entities is defined to be the distance between the two closest elements in the two groups.* Therefore, only one member of each group, and not all of them, are used for computing the similarity. The linkage-based dual of GMCP is GMST [33]. the only difference between their definitions is that the cost of the feasible solution $\mathbf{V}_s$ is defined as the cost of the Minimum Spanning Tree found on its nodes rather than the cost of the complete graph it forms: $C_{MST}(\mathbf{V}_s) = MST(\mathbf{V}_s)$. GMST can potentially deal with disjoint groups because of its linkage mechanism: consider the exemplified case in Fig. 4b. In order to link the two groups in GMST, a single edge between two nodes from the two groups would suffice. Therefore, all of the red nodes of both groups can be included in $\hat{\mathbf{V}}_s$ at the cost of a single long edge which does not add a considerable value to the overall cost of the solution. This is dissimilar to GMCP where all the nodes in the two different groups have to be connected pairwise and consequently would induce an excessive cost. Therefore, GMST is capable of dealing with the issue of disjoint inlier NNs without the need to a robust distance function. Fig. 6d shows the selected nodes by GMST which demonstrates that inliers from all groups are included.

However, we use GMST as a baseline in our experiments as we will show that GMCP with the robust distance

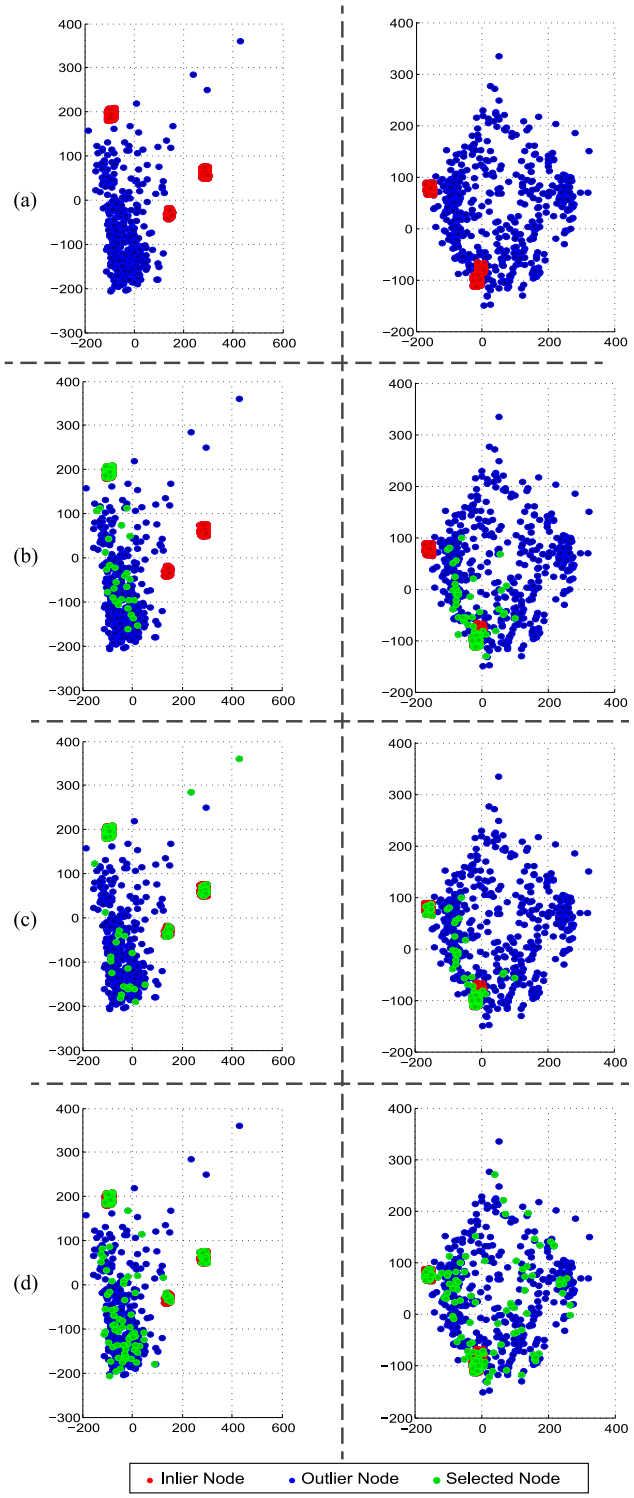

Fig. 6. The results of robustification. Right and left columns show sample cases with two and three disjoint inlier groups. In (a), the outlier and inlier nodes of $\mathbf{V}$ are shown in blue and red, respectively. In (c) and (b), the green nodes show the ones selected by GMCP with and without robustification, respectively. Note that there are some outliers included in $\hat{\mathbf{V}}_s$; which typically correspond to the query features without any inlier NN. (d) shows the selected nodes by GMST. Color histogram was employed as the global feature.

function outperforms GMST due to an issue known as *chaining phenomenon* in the linkage-based clustering literature [32], [34]. This phenomenon makes linkage-based methods essentially prone to outliers and noise. In the context of our problem, Chaining Phenomenon occurs when an outlier, which is distant to the majority of inlier nodes, is included in the selected subgraph merely due to being in the proximity of a single inlier node. That way, the outlier will be incorrectly included in the solution since the linkage mechanism of GMST considers only the *shortest* distance between a selected node and the remaining ones, and not all of the distances. Reference [34] provides an in-depth explanation of the chaining phenomenon.

2.3 Location Estimation Using the Matched Feature Points

The benefits of using the GMCP-based method for feature matching is twofold: First, it matches the query features to the top few matching reference images; this trait causes the query features which are typically assigned to incorrect reference images using only the first NN to be matched to the top few reference images. Second, the algorithm favors to assign the majority of query features to the strongest match among the top few discovered reference images; this is because the distance between the global features of the NNs belonging to the same image is zero, while the distance between the global features of two different matching reference images is non-zero, even though small. These two characteristics cause the strongest matching image found by GMCP to be more accurate compared to the one found by the baselines. Therefore, we estimate the location of the query using a winner-take-all scenario in which the reference image found by GMCP to have the highest number of matched feature points is selected as the strongest match, and its location is identified as an approximation of the location of the query image. If the reference data set is a dense sampling of the covered area, which is typically the case for our data set, the location of the strongest match is expected to be within a few tens of meters of the ground truth which is generally acceptable for a city scale localization.

In addition, the matched feature points by GMCP could be used for performing further reasoning about the camera location of the query. For instance, the top few reference images with a number of matched features beyond a certain threshold can be identified. Then, the camera matrix of the query image, which includes its geo-location, can be computed using the feature correspondences found by GMCP utilizing epipolar geometry-based techniques. In the experiments section, we will show that the simple yet effective winner-take-all scheme yields satisfactory results for geo-localization at a city scale. However, the epipolar geometry-based methods using the feature correspondences found by GMCP are useful for finer localization.

3 Solving GMCP

GMCP is an $\mathcal{NP}$-hard problem [16]. A few approaches to solving GMCP such as branch-and-cut and multi-greedy heuristics [27], [35] have been explored to date; however, the majority of them are formulated according to particular problems and are suited for small inputs [16]. Our graphs typically include $L \times k = 300$ to $1,500$ nodes which require an efficient and approximate solver as the problem is

Fig. 7. Left: Forty sample street view images belonging to eight place marks of the reference data set. Right: Sixteen sample user uploaded images from the test set.

$\mathcal{NP}$-hard. Therefore, we employ Local-neighborhood Search to solve the optimization problem of Equation (5) as it has been shown to work efficiently for large combinatorial problems such as Tabu-search for GMST [36], [37].

Local-neighborhood search methods are based on examining the neighbors of the current solution in hope of discovering a better one. Two solutions are neighbors of size ε if they are identical except in ε elements. Choosing a small neighborhood size makes the optimization process prone to getting stuck at suboptimal regions. On the other hand, choosing a large neighborhood significantly enlarges the number of neighbors in each iteration, resulting in an increase in the complexity. In order to deal with this issues, we use a different approach in which we change the neighborhood size from 1 to δ repeatedly in each iteration.

Algorithm 1 Local Neighborhood Search GMCP Solver

Initialize the best solution, $\hat{\mathbf{V}}_s$, with a random solution.
while termination conditions not satisfied **do**
  $\mathbf{N}_{size-1} \leftarrow$ size-1 neighbors of $\hat{\mathbf{V}}_s$.
  $\mathbf{\Gamma}_{size-1} \leftarrow \delta$ neighbors in $\mathbf{N}_{size-1}$ with the lowest costs.
  $\mathbf{\Lambda} \leftarrow$ the elements changed in $\mathbf{\Gamma}_{size-1}$.
  $\mathbf{N}_{size-\delta} \leftarrow$ size-1 to size-δ neighbors of $\hat{\mathbf{V}}_s$; only the elements in $\mathbf{\Lambda}$ are allowed to change.
  $\hat{\mathbf{N}} \leftarrow$ the solution with the lowest cost in $(\mathbf{N}_{size-1} \cup \mathbf{N}_{size-\delta})$.
  if (cost of $\hat{\mathbf{V}}_s$) $\geq$ (cost of $\hat{\mathbf{N}}$) **then**
    $\hat{\mathbf{V}}_s \leftarrow \hat{\mathbf{N}}$
  else
    return $\hat{\mathbf{V}}_s$ as the found solution.
  end if
end while
return $\hat{\mathbf{V}}_s$ as the found solution.

The details of our solver are provided in Algorithm 1. First, the solver is initialized with a random solution. We fix the neighborhood size to 1 and identify the δ best solutions. Next, we fix the neighborhood size to δ while we allow only the elements replaced in the top δ neighbors of size-1 to change and find the resulting neighbors. If any of the size-1 to size-δ neighbors induce a cost lower than the best known solution, the best solution is updated. This procedure continues iteratively until the minimum is found or a termination condition (maximum time or maximum number of iterations) is met. Our algorithm allows up to δ elements to change in one iteration, yet we do not need to examine all of the feasible neighbors of size δ. This can accelerate the process up to δ times where $L \gg k^{\delta-1}$.

In order to investigate the optimality of our solver, we generated $1,000$ random GMCP instances with L (ranging from 4 to 15) clusters and k (ranging from 3 to 8) nodes in each cluster. We found the optimal GMCP solution of these instances using exhaustive search and compared them against the solution found by the proposed solver. In 79 percent of the instances, our approximate solver converged to the optimal answer. Additionally, in the majority of the rest of the cases, the solution found by our solver was less than 30 percent different from the optimal answer.

Employing the proposed optimization method with $\delta = 3$, we can solve a typical GMCP instance of our feature matching problem in less than one second on average, using non-optimized MATLAB code on an octo-core 2.4 GHz machine.

4 Experimental Results

4.1 Evaluation Data Set

We evaluated the proposed algorithm using a reference data set of over 102,000 Google Street View images. The data set covers downtown and the neighboring areas of Pittsburgh, PA; Orlando, FL and partially Manhattan, NY. Fig. 7(left) shows forty sample reference images belonging to eight place marks. The place marks are approximately 12m apart, and the data set covers about 204km of urban streets. The 360 degrees view of each place mark is broken down into four side views and one top view image. The test set consists of 644 unconstrained user uploaded images downloaded from Flickr, Panoramio and Picasa which are GPS-tagged by users. We manually verified their location and made the necessary adjustments as the user specified GPS-tags are often inaccurate. In our experiments, each query image is matched against the entire reference data set and not the ground truth city only. Sixteen sample queries are shown in Fig. 7(right). The quality of our reference set, which will be made available to the public for research purposes,[4] surpasses that of the currently available street view data sets [2], [38] in terms of the image resolution.

4. For downloading the dataset please visit: http://crcv.ucf.edu/projects/GMCP_Geolocalization/.

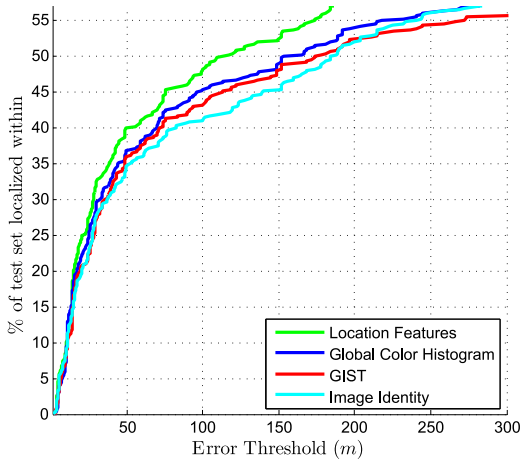

Fig. 8. Comparison of the overall geo-localization results using different global features.

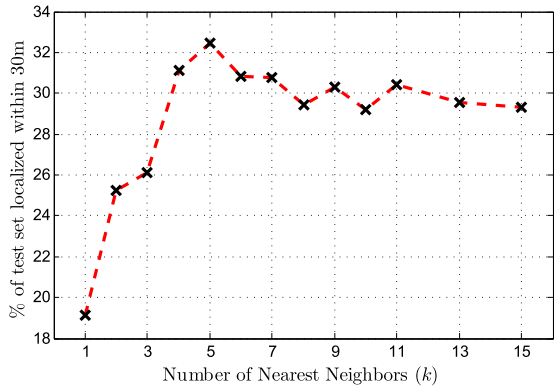

Fig. 9. Geo-localization accuracy with respect to k.

4.2 Analysis of the Proposed Method

In this section, we provide two experiments to quantitatively analyze various aspects of the proposed method and demonstrate its robustness.

Comparison of different global features. Any arbitrary type of global features can be used in the proposed framework. Fig. 8 compares the geo-localization results obtained by four different global features while setting $k = 5$ and employing the winner-take-all voting scheme for location estimation as explained in Section 2.3. We normalized the local and global feature vectors prior to forming the input to GMCP in order to make sure they produce comparable distances and do not dominate each other. The horizontal and vertical axes shows the error threshold in meters and the percentage of the test set localized within a particular error threshold, respectively. Since the scope of this work is precise localization in a city scale, we focus on error values less than 300 meters in our plots as a higher error typically implies failure.

The blue and red curves show the results of using a 960-dimensional GIST [39] and a 60-dimensional RGB Color Histogram as the global feature, respectively. Each image in our reference data set is associated with a GPS-tag denoted by a two dimensional vector of latitude and longitude (ϕ, λ). Even though the location is not based on visual information, it can serve as the global feature as it is a holistic tag for the image. The green curve was computed using the (ϕ, λ) location vector as the global feature after conversion to Cartesian coordinate values. The superior performance of the location features is mainly because they provide complimentary information to the visual content of the image as they are non-visual descriptors. We used the location as the global feature in the rest of our experiments.

The cyan curve depicts the results of using the image identity as the global feature; that is, the edge weight between two NNs is zero if they come from the same reference image and 1, otherwise. The fact that the other global features perform better than the image identity, in particular location and color histogram by up to 7.4 and 4.8 percent, respectively, signifies that the improvement made by our algorithm is not due to simply encouraging the NN matches to be selected from one image or a small set of images. In other words, GMCP is indeed leveraging the relationship between the global features of the NNs to identify the inlier nodes. However, Fig. 8 shows that different global features have different performances in encoding the relationships among NNs, and choosing the appropriate type of global feature is essential.

The value of σ in Equations (6) and (7) was determined empirically using a small validation set of 10 random queries for which there are multiple disjoint groups of inlier NNs. This resulted in the values of 32, 1,024 and 256 for color histogram, GIST and location features, respectively. Typically, the bandwidth depends on the type of the feature, the number of dimensions and the range of values.

Number of considered nearest neighbors (k). The appropriate value for k in the GMCP-based method depends on the amount of repetition and similarity in the features of the reference data set. An insufficiently large k would lead to considering too few NNs in the matching process and consequently a small chance of discovering the correct one. On the other hand, an excessively large value would result in too many noisy NNs in the input graph and higher complexity of the optimization task. In order to show the impact of the number of considered NNs, we performed an experiment by running the GMCP-based method with different values of k ranging from 2 to 15; Fig. 9 provides the percentage of the test set localized within the arbitrary distance threshold of 30 meters. For all values of k (when >1), the overall accuracy is observed to be significantly higher than the baseline, i.e., using the first NN only. However, when k becomes too large (>5 for our data set), more features which do not have any inlier among their NNs survive the pruning, and consequently the accuracy slightly decreases. This observation is consistent with the characteristics of our data set as a building is typically visible in the view of 3 to 5 street view place marks. Therefore, there is a higher chance to find the correct NN among the top five NNs. We set $k = 5$ in the rest of our experiments.

4.3 Comparison of the Geo-Localization Results

Fig. 10 shows the results of evaluating the proposed algorithm along with the baselines in terms of *overall geo-localization results*. The black and light green curves illustrate the performance of GMST and GMCP based feature matching

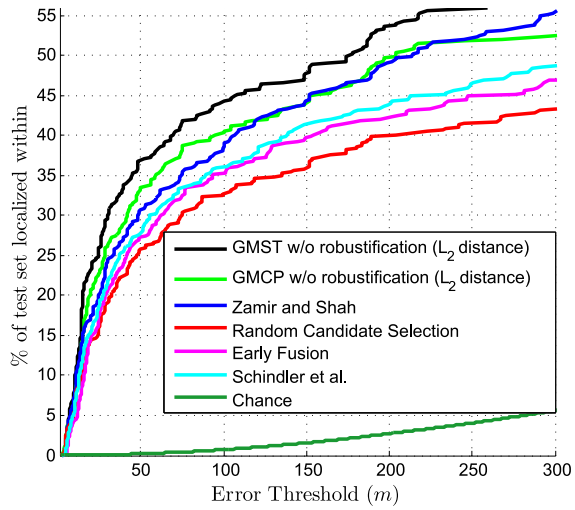

Fig. 10. Overall geo-localization results using GMST and GMCP (without robustification) along with the baselines. Horizontal and vertical axes show the distance threshold and the percentage of the test set localized within the distance threshold, respectively.

methods, respectively when no robustification is utilized (employing ℓ_2 distance instead), along with the winner-take-all voting scheme for location estimation. The dark green curve illustrates the performance of chance where the query images were randomly matched to the reference images; the curve is generated by calculating the expected value of the percentage of test set localized within a particular error threshold. The poor performance of chance in Fig. 10 is due to the fact that the covered area is several square kilometers wide. Therefore, it would be unlikely to randomly localize an image within a few hundred meters of its actual location. However, even though many of the query images are localized within few tens of meters of their ground truth using our method, yet some large error values of over 150 meters are observed in the curves of Fig. 10. There are two main reasons behind such large distances: First, many of the user uploaded query images were taken at locations where the closest street view place mark is over 150 meters away, e.g., parks and play grounds. For those cases, a large error will be reported in the localization curves even if the algorithm finds the best matching street view image. Second, many of the tall and large buildings in urban area have facades which look identical from different viewpoints. We observed that the algorithm often matches a query image of such facades to the correct building yet not necessarily the correct facade. Those cases typically lead to large error values which are due to the confusion caused by the symmetry of buildings. In the next section, we provide the quantitative results which show the majority of images localized within a few hundred meters of ground truth indeed have an overlap in scene with the matched image.

The red curve in Fig. 10 represents a baseline in which one of the top k NNs for each feature point is randomly selected as the match. The cyan curve shows the localization results obtained by Schindler et al.'s method [3] which is based on image matching employing the bag of visual words model. The blue curve depicts the results of our previous work [2] which is based on using the first NN only. By increasing the size of the data set, the pruning methods which are based on the first NN only, e.g., [26] and [2], tend to over-prune the query features. This results in too few query features participating in the voting scheme and consequently less reliable geo-localization which is one of the reasons behind the relatively low accuracy achieved by [2].

Early fusion. One way of combining local and global features, typically termed early fusion, is normalizing the features vectors, concatenating them and treating the new vector as one feature. The purple curve in Fig. 10 depicts the localization results when feature matching is performed using this method. Using early fusion for feature matching has a number of inherent disadvantages which explain its poor performance. For instance, treating concatenated features as one vector requires the two features to be matched in the same space and using one type of distance, which is undesirable in many cases.

4.4 Results of Robustifying the Distance Function

The results illustrated in Fig. 10 were obtained *without* using the robust distance function, which is why the improvement made by the GMCP-based method is not more than 3 percent over the baselines. The solid and dashed green curves in Fig. 11a compare the performance of GMCP with and without the robust distance function D (using ℓ_2 distance instead), which shows the robustification improves the overall localization results by up to 7 percent as it deals with the issue of disjoint inlier groups.

In Section 2.2.2 we argued that GMST outperforms GMCP-without-robustification due to its capability of dealing with the disjoint inlier groups. This is consistent with the experimental results of Fig. 11a. However, as shown in Fig. 11a, using the robust distance function does not make a sensible difference in the results of GMST since the discussion of Section 2.2.2 is not applicable to it. Additionally, Fig. 11a shows that the robustified GMCP performs significantly better than GMST. This is due to chaining phenomena which makes GMST more prone to noise and outliers as discussed in Section 2.2.2. Therefore, we conclude that employing GMCP with the robust distance function D yields the best results in the proposed framework.

In addition, in order to provide further insight into the quality of our results, we manually verified if the found street view images actually match their corresponding query images: We found 94.4 percent of the query images which were localized within 300 meters of the ground truth by the robustified GMCP (i.e., the solid green curve in Fig. 11) had an overlap in scene with the found reference image while only 5.6 percent were matched incorrectly and did not show anything in common.

Fig. 12(left) shows the geo-localization results obtained using various distance functions, and the corresponding functions are illustrated in the right plot. The linear loss (Euclidean) represents the case where ℓ_2 norm was used as the distance (i.e., no robustification). The purple and the dark blue curves illustrate the results of using Squared ($\|\mathbf{x}-\mathbf{y}\|^2$) and Huber-loss [40] functions with tuned σ, respectively. As apparent in Fig. 12(left), robustification using the distance function D yields the best results which justifies our choice. In particular, the fact that Huber-loss performs worse than the distance function D signifies that

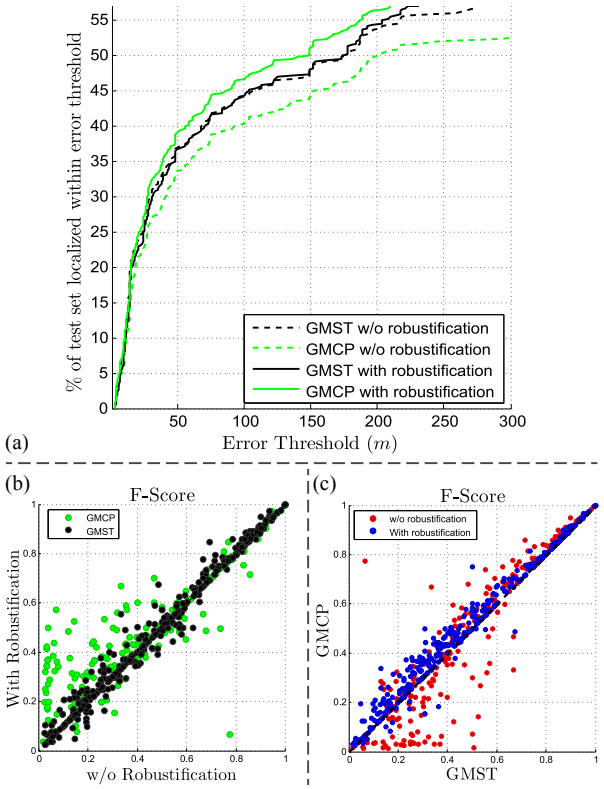

Fig. 11. The impact of using the robust distance function. (a) depicts overall geo-location results. Note the significant positive effect on GMCP results, and negligible impact on GMST. (b) and (c) show the scatter plots of F-score values. (b) illustrates the effect of robustification; green and black points show that for GMCP and GMST, respectively. (c) compares the performance of GMCP to GMST; that is shown for the two settings of with and without robustification.

mapping the large distances to a maximal value, as compared to only damping them with an non-zero slope (which is what Huber-loss does by definition) is essential. Additionally, Squared loss boosts the large distances, and consequently has the worst results, which empirically shows our argument on the necessity of addressing the large distances is valid.

4.5 Feature Matching Evaluation

The evaluation of the proposed method and the baselines in terms of the *performance of feature matching* is shown using scatter plots of precision, recall and F-score values (Figs. 11 and 13). For each query, the set of reference images that have an overlap with it are known. Therefore, we can examine how many of the query features are matched to one of these correct image matches. The precision of feature matching is defined as the number of correctly matched query features divided by the total number of query features after pruning. Recall is defined as the number of correctly matched query features divided by the total number of query features which have a NN belonging to one of the matching reference images among their top k NNs. F-score is a measure which combines both precision and recall and is defined as their harmonic mean.

The green points (where each point represents one query image) in the F-score scatter plot of Fig. 11b compare the performance of GMCP before and after robustification. The diagonal dashed line shows the neutral

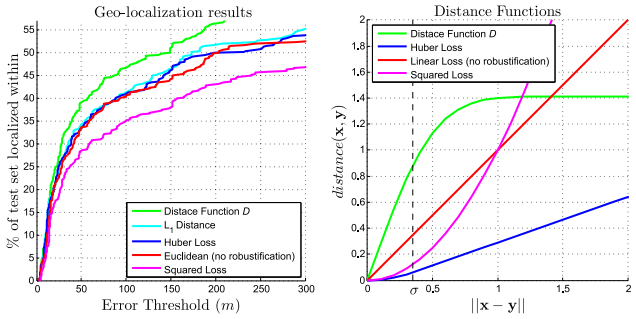

Fig. 12. Left: Geo-localization results using various distance functions. Right: Illustration of the functions.

border where the performances of both settings is the same. Therefore, a green node above the dashed line represents a query for which the performance of GMCP is improved after robustification. Similarly, the impact of robustification on GMST is shown using the black points. As apparent in Fig. 11b, the performance of GMST does not considerably change with and without robustification, while GMCP's performance significantly improves by employing the robust distance function D.

The green nodes in Fig. 11b which are positioned close to the neutral line represent the query images which do not match to more than one group of matching reference images, and consequently the robustification does not make a sensible difference on their corresponding performance. On the other hand, the nodes which are located above the neutral line represent the queries which have disjoint groups of matching reference images.

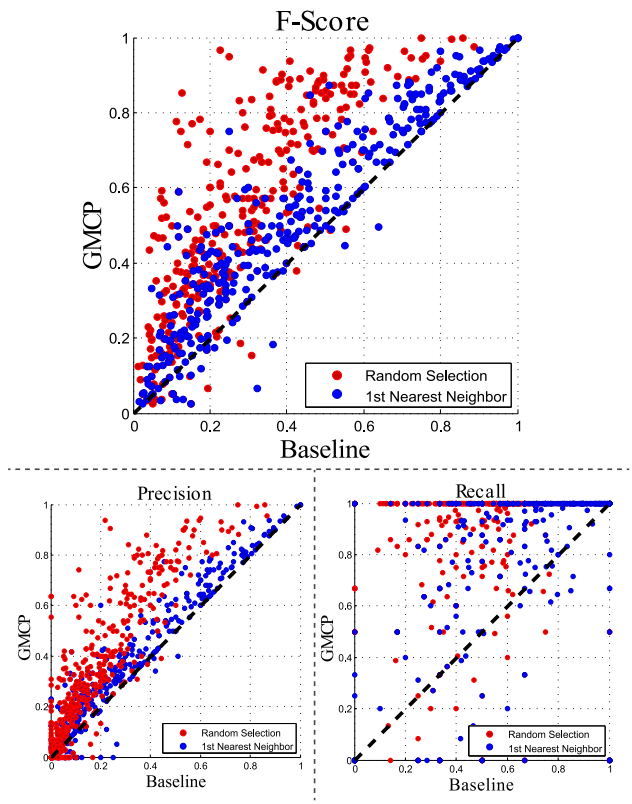

Fig. 13. Scatter plots of F-score, precision and recall values. Vertical and horizontal axes show the values gained by the GMCP-based method and the baseline, respectively. Each node represents one query image.

The scatter plot of Fig. 11c compares the performance of GMCP versus GMST on individual query images: that is shown for the two settings of *with* and *without* robustification in blue and red, respectively. This plot signifies that GMCP is superior to GMST when the robust distance function is utilized.

Fig. 13 shows the scatter plots of precision, recall and F-score values which compare the performance of GMCP (with the robust distance function) vs. the baselines. Blue and red baselines represent using the 1st NN and randomly selecting one of the NNs, respectively. As apparent in this figure, the observed improvement in the results of feature matching is attributed to considering more than one NN, and the performance of the GMCP-based method in robustly discovering the correct feature correspondences.

5 Conclusion

In this paper, a novel framework for geo-localizing images in urban areas was proposed. We developed a multiple-NN feature matching method based on Generalized Minimum Clique Problem. The proposed method is capable of incorporating both global and local features simultaneously. We showed that using a robustified function for finding the distances between the global features is essential when the query image matches multiple reference images with dissimilar global features. Additionally, different types of local features can be used for nominating the NNs. Therefore, our method can be adopted to utilize multiple types of local features in order to maximize the amount of leveraged information. We evaluated the proposed algorithm on a new reference data set of Google Street View images which will be made available to the public for research purposes.

References

[1] J. Hays and A. Efros, "IM2GPS: Estimating Geographic Information from a Single Image," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2008.

[2] A.R. Zamir and M. Shah, "Accurate Image Localization Based on Google Maps Street View," *Proc. 11th European Conf. Computer Vision*, 2010.

[3] G. Schindler, M. Brown, and R. Szeliski, "City-Scale Location Recognition," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2007.

[4] J. Knopp, J. Sivic, and T. Pajdla, "Avoding Confusing Features in Place Recognition," *Proc. European Conf. Computer Vision*, 2010.

[5] J. Sivic and A. Zisserman, "Video Google: A Text Retrieval Approach to Object Matching in Videos," *Proc. Ninth IEEE Int'l Conf. Computer Vision*, 2003.

[6] T. Sattler, B. Leibe, and L. Kobbelt, "Fast Image-Based Localization Using Direct 2D-to-3D Matching," *Proc. 13th IEEE Int'l Conf. Computer Vision*, 2011.

[7] Y. Li, N. Snavely, and D.P. Huttenlocher, "Location Recognition Using Prioritized Feature Matching," *Proc. 11th European Conf. Computer Vision*, 2010.

[8] T. Sattler, B. Leibe, and L. Kobbelt, "Improving Image-Based Localization by Active Correspondence Search," *Proc. 12th European Conf. Computer Vision*, 2012.

[9] E.N. Mortensen, H. Deng, and L. Shapiro, "A Sift Descriptor with Global Context," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2005.

[10] K. Mikolajczyk, A. Zisserman, and C. Schmid, "Shape Recognition with Edge-Based Features," *Proc. British Machine Vision Conf.*, 2003.

[11] B. Cao, C. Ma, and Z. Liu, "Affine-Invariant Sift Descriptor with Global Context," *Proc. Third Int'l Congress on Image and Signal Processing*, 2010.

[12] Q. Hao, R. Cai, Z. Li, L. Zhang, Y. Pang, and F. Wu, "3D Visual Phrases for Landmark Recognition," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2012.

[13] Y. Zhang, Z. Jia, and T. Chen, "Image Retrieval with Geometry-Preserving Visual Phrases," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2011.

[14] A. Torii, J. Sivic, T. Pajdla, and M. Okutomi, "Visual Place Recognition with Repetitive Structures," *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, 2013.

[15] H. Jegou, M. Douze, and C. Schmid, "On the Burstiness of Visual Elements," *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, 2009.

[16] C. Feremans, M. Labbe, and G. Laporte, "Generalized Network Design Problems," *European J. Operational Research*, vol. 148, no. 1, pp. 1-13, 2003.

[17] Y. Avrithis, G. Tolias, and Y. Kalantidis, "Feature Map Hashing: Sub-Linear Indexing of Appearance and Global Geometry," *Proc. Int'l Conf. Multimedia*, 2010.

[18] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Object Retrieval with Large Vocabularies and Fast Spatial Matching," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2007.

[19] A. Hakeem, R. Vezzani, M. Shah, and R. Cucchaira, "Estimating Geospatial Trajectory of a Moving Camera," *Proc. 18th Int'l Conf. Pattern Recognition*, 2006.

[20] T. Sattler, B. Leibe, and L. Kobbelt, "SCRAMSAC: Improving Ransac's Efficiency with a Spatial Consistency Filter," *Proc. IEEE 12th Int'l Conf. Computer Vision*, 2009.

[21] P.H.S. Torr and A. Zisserman, "MLESAC: A New Robust Estimator with Application to Estimating Image Geometry," *Computer Vision and Image Understanding*, vol. 78, no. 1, pp. 138-156, Apr. 2000.

[22] H. Wang, D. Mirota, and G.D. Hager, "A Generalized Kernel Consensus-Based Robust Estimator," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 32, no. 1, pp. 178-184, Jan. 2010.

[23] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Lost in Quantization: Improving Particular Object Retrieval in Large Scale Image Databases," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2008.

[24] J.L. Bentley, "Multidimensional Binary Search Trees in Database Applications," *IEEE Trans. Software Eng.*, vol. 5, no. 4, pp. 333-340, July 1979.

[25] M. Muja and D.G. Lowe, "Fast Approximate Nearest Neighbors with Automatic Algorithm," *Proc. Int'l Conf. Computer Vision Theory and Applications*, 2009.

[26] D.G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *Int'l J. Computer Vision*, vol. 60, pp. 91-110, 2004.

[27] A. Koster, S.V. Hoesel, and A.W.J. Kolen, "The Partial Constraint Satisfaction Problem: Facets and Lifting Theorems," *Operations Research Letters*, vol. 23, pp. 89-97, 1998.

[28] A.R. Zamir, A. Dehghan, and M. Shah, "GMCP-Tracker: Global Multi-Object Tracking Using Generalized Minimum Clique Graphs," *Proc. 12th European Conf. Computer Vision*, 2012.

[29] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, 1988.

[30] Y. Weiss, "Correctness of Local Probability Propagation in Graphical Models with Loop," *Neural Computation*, vol. 12, pp. 1-41, 2000.

[31] G. Wu, E.Y. Chang, and N. Panda, "Formulating Distance Functions via the Kernel Trick," *Proc. 11th ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining*, 2005.

[32] J.C. Gower and G.J.S. Ross, "Minimum Spanning Trees and Single Linkage Cluster Analysis," *J. Royal Statistical Soc.*, vol. 18, pp. 54-64, 1969.

[33] Y.S. Myung, C.H. Lee, and D.W. Tcha, "On the Generalized Minimum Spanning Tree Problem," *Networks*, vol. 26, pp. 231-241, 1995.

[34] B.S. Everitt, S. Landau, and M. Leese, *Cluster Analysis*, fourth ed., John Wiley & Sons, 2009.

[35] E. Althaus, O. Kohlbacher, H.P. Lenhof, and P. Müller, "A Combinatorial Approach to Protein Docking with Flexible Side Chains," *J. Computational Biology*, vol. 9, pp. 597-612, 2002.

[36] D. Ghosh, "Solving Medium to Large Sized Euclidean Generalized Minimum Spanning Tree Problems," WP No. 2003-08-02, Indian Inst. of Management, 2003.

[37] Z. Wang, C.H. Che, and A. Lim, "Tabu Search for Generalized Minimum Spanning Tree Problem," *Proc. Ninth Pacific Rim Int'l Conf. Artificial Intelligence*, 2006.
[38] D.M. Chen, G. Baatz, K. Koser, S.S. Tsai, R. Vedantham, T. Pylvanainen, K. Roimela, X. Chen, J. Bach, M. Pollefeys, B. Girod, and R. Grzeszczuk, "City-Scale Landmark Identification on Mobile Devices," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2011.
[39] A. Torralba, "Contextual Priming for Object Detection," *Int'l J. Computer Vision*, vol. 53, no. 2, pp. 169-191, 2003.
[40] P.J. Huber, "Robust Estimation of a Location Parameter," *The Annals of Math. Statistics*, vol. 3, no. 1, pp. 73-101, 1964.

Amir Roshan Zamir received the MS degree in computer engineering from the University of Central Florida (UCF). He has been with UCF's Center for Research in Computer Vision (CRCV) since 2009, where he is currently working toward the PhD degree in the electrical engineering. He has published several papers in conferences and journals such as CVPR, ECCV, ACM Multimedia, and *IEEE Transactions on Pattern Analysis and Machine Intelligence*. He was a program committee member of the CVPR '13 Workshop on Visual Analysis and Geo-Localization of Large-Scale Imagery, and the program chair of ICCV '13 Workshop on Action Recognition with a Large Number of Classes (THUMOS). His research interests include image and video geo-localization, location based services, data association, action recognition, mathematical optimization, and graph theory. He received UCF's Research Excellence Award and National Geospatial-Intelligence Agency NARP-SW Best Research Poster Award.

Mubarak Shah, the Trustee chair professor of computer science, is the founding director of the Center for Research in Computer Vision at the University of Central Florida (UCF). He is an editor of an international book series on video computing, editor-in-chief of *Machine Vision and Applications* journal, and an associate editor of *ACM Computing Surveys* journal. He was the program cochair of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) in 2008, an associate editor of the *IEEE Transactions on Pattern Analysis and Machine Intelligence*, and a guest editor of the special issue of *the International Journal of Computer Vision on Video Computing*. His research interests include video surveillance, visual tracking, human activity recognition, visual analysis of crowded scenes, video registration, UAV video analysis, and so on. He is an ACM distinguished speaker. He was an IEEE distinguished visitor speaker for 1997-2000 and received the IEEE Outstanding Engineering Educator Award in 1997. In 2006, he was awarded a Pegasus Professor Award, the highest award at UCF. He received the Harris Corporation's Engineering Achievement Award in 1999, TOKTEN awards from UNDP in 1995, 1997, and 2000, Teaching Incentive Program Award in 1995 and 2003, Research Incentive Award in 2003 and 2009, Millionaire's Club Awards in 2005 and 2006, University Distinguished Researcher Award in 2007, Honorable mention for the ICCV 2005 Where Am I? Challenge Problem, and was nominated for the Best Paper Award at the ACM Multimedia Conference in 2005. He is a fellow of the IEEE, AAAS, IAPR, and SPIE.