3D Point Cloud Reduction using Mixed-integer Quadratic Programming

Hyun Soo Park*
Canegie Mellon University
hyunsoop@cs.cmu.edu

Yu Wang*
Canegie Mellon University
yuw@andrew.cmu.edu

Eriko Nurvitadhi
Intel Corporation
eriko.nurvitadhi@intel.com

James C. Hoe
Canegie Mellon University
jhoe@ece.cmu.edu

Yaser Sheikh
Canegie Mellon University
yaser@cs.cmu.edu

Mei Chen
Intel Corporation
mei.chen@intel.com

Abstract

Large scale 3D image localization requires computationally expensive matching between 2D feature points in the query image and a 3D point cloud. In this paper, we present a method to accelerate the matching process and to reduce the memory footprint by analyzing the view-statistics of points in a training corpus. Given a training image set that is representative of common views of a scene, our approach identifies a compact subset of the 3D point cloud for efficient localization, while achieving comparable localization performance to using the full 3D point cloud. We demonstrate that the problem can be precisely formulated as a mixed-integer quadratic program and present a point-wise descriptor calibration process to improve matching. We show that our algorithm outperforms the state-of-the-art greedy algorithm on standard datasets, on measures of both point-cloud compression and localization accuracy.

1. Introduction

Advances in structure from motion techniques have made it possible to construct 3D point cloud models at 'city-scale' with millions of feature points in a matter of hours [1]. To localize a query image we need to find correspondences between the 2D local features in the image and points in the 3D point cloud model. This correspondence search is equivalent to a nearest neighbor search when a descriptor (such as SIFT [20]) is associated with each 3D point. Although effective for laboratory datasets, the memory footprint and computational requirements of matching becomes prohibitive as larger and larger descriptor sets are considered.

Large-scale 3D point clouds offer significant opportunities for reduction. First, physical constraints, such as roads/walkways and the average height of typical photographers, produce considerable structure in the positions and orientations that are likely to occur in a space. These correlations are captured in the 3D point cloud and can be used to identify 'low-value' points that are unlikely to be used for future localization. Second, the spatial distribution of 3D point clouds reflects the texture statistics of the environment as well as the view statistics of the photographers. Since we are interested only in ensuring that future cameras can be accurately and efficiently localized, the point cloud can be culled to better reflect view statistics independently. Finally, during 3D reconstruction, certain points may simply not be accurately reconstructed and are therefore unlikely to help for future localizations. In this paper, we present a method to reduce the 3D point cloud explicitly based on the view statistics of a training corpus. It determines a compact subset of the full 3D point database that delivers comparable camera registration performance as using the full set. Given an image that is representative of view-sampling patterns in a given scene, our algorithm seeks a compact subset of the points such that at least b points in the subset are visible from a query image—a constraint inherited directly from typical perspective-n-point algorithms.

This problem is related to the maximum coverage problem, or the K-cover problem. The objective of the maximum coverage problem is to find K 3D points that maximize the sum of the number of 2D-3D correspondences for all training images; whereas for our problem, the cost of the subset, e.g., the number of 3D points in the subset, is minimized while keeping the number of correspondences in each image larger than a threshold (the required minimal number of correspondences to localize the image accurately). Karp [14] noted that the maximum coverage problem is NP-complete, so is our problem. Li et al. [18] and Irschara et al. [11] presented greedy techniques that approximately solve the problem. However, these greedy approaches are sub-optimal and it is difficult to characterize

*indicates equal contribiutions

978-0-7695-4990-3/13 $26.00 © 2013 IEEE
DOI 10.1109/CVPRW.2013.41

or analyze their precise behavior. Thus, even small design modifications of the objective function are difficult to incorporate into the algorithm, and may necessitate the design of an entirely new greedy procedure. Our approach formulates this problem as a mixed-integer linear/quadratic programming problem. This allows us to obtain an optimal 3D point subset (using branch-and-bound) to deliver better data compression rate and camera registration performance than greedy approaches. This also allows flexibility to users to design and modify the objective function based on the application.

Contributions. We present an algorithm for 3D point cloud reduction that outperforms state-of-the-art approaches on standard datasets. We formulate the problem as a mixed-integer program that is closely integrated with camera pose estimation algorithms (e.g., the number-of-inliers parameter in a RANSAC pose estimator). This formulation allows explicit characterization of occurrence and co-occurence constraints, facilitating future algorithm development. We also introduce a generative measure to find a correct match based on Mahalanobis distance, which allows us to obviate the need for finding the second nearest neighbor. This measure provides much tighter threshold directly learned from a database and results in higher success rate for RANSAC based matching.

2. Related Work

Image localization is one of the core problems in computer vision and robotics. There is a significant body of literature related to matching a set of images against a large non-coincidental image repository for localization. We discuss existing localization methods in this section.

Image localization techniques often incorporate other sensory data that show a strong correlation with images. Cozman and Krotkov [5] introduced localization of an image taken from unknown territory using temporal changes in sun altitudes. Jacobs et al. [12] incorporate weather data reported by satellite imagery to localize widely distributed cameras. They find matches between weather conditions on images over an year and the expected weather changes indicated by satellite imagery. As GPS becomes a viable solution for localization in many applications, GPS-tagged images can help to localize images that do not have the tags. Zhang and Kosecka [32] built a GPS-tagged image repository in urban environments and find correspondences between a query image and the database based on SIFT features [20]. Hays and Efros [10] leveraged GPS-tagged internet images to estimate a probability distribution over the earth and Kalogerakis et al. [13] extended the work to disambiguate locations of the images without distinct landmarks. They applied a travel prior in the form of temporal information in the images. Baatz et al. [3] estimated image location based on a 3D elevation model of mountain terrains and evaluated their method on the scale of a country (Switzerland).

Pure image based localization has also been studied extensively. Torralba et al. [30] used global context to recognize a scene category using a hidden Markov model framework. Se et al. [26] applied a RANSAC framework for global camera registration. Robertson and Cipolla [24] estimated image positions relative to a set of rectified views of facades registered onto a city map. Zamir and Shah [31] leveraged the Google Street View images that provide accurate geo-location. Cummins and Newman [6] showed a visual SLAM system that reliably estimates camera poses. Structure from motion have also been employed for large scale image localization. Snavely et al. [28] exploited structure from motion to browse a photo collection from the exact location where it was taken. They used hundreds of images to register in 3D. Agarwal et al. [1] presented a parallelizable system that can reconstruct hundreds of thousands of images (city scale) within two days. Frahm et al [8] showed larger scale reconstruction (millions of images) that can be executed on a single PC.

As the database becomes larger, the matching process between a query image and the database becomes computationally expensive. A number of algorithms have been introduced to accelerate the matching process. A vocabulary tree proposed by Nistér and Stewénius [22] has been widely adopted in image localization. Havlena et al. [9] indexed images using visual vocabularies to measure similarity between images and construct a graph that prioritizes image matching. Irschara et al. [11] synthesized views by exploiting the relationship between images and the point cloud and indexed synthesized view points based on coverage of projected 3D points. Tree building and search method based on N-best paths was proposed by Schindler et al. [25], while a vector quantization method was adopted by Baatz et al. [2]. Chen et al. [4] applied the visual vocabulary tree to localize images from various mobile devices. A graph representation of a image set can reduce the search space significantly. Simon et al. [27] presented a method to find a minimal set of images that can represent whole image sets via 3D reconstruction. Snavely et al. [29] employs skeletal graph models to identify a compact subset among a highly redundant image collection. Li et al. [17] represented a collection of images with an iconic image graph that enabled them to search on a tree structure. Li and Kosecka [16] showed a method to select discrimitve features that are optimized for location recognition. Ni et al. [21] adapted compact image epitomes for query image localization.

Our image localization approach exploits a 3D point cloud reconstructed by a collection of images. Our database representation includes 3D points and corresponding feature vectors. We find a compact subset of the 3D point cloud such that a query image can find at least b matches. This ap-

proach is related to the method by Li et al. [18]. They proposed a prioritization scheme to avoid having to match every point in the model against the query image. They show that using a reduced set of points of the highest priority is better than using all 3D points, as it permits registration of more images while reducing the time needed for registration. Since this method does not constrain the set of possible views, it outperforms the algorithm by Irschara et al. [11] in terms of the number of images that can be registered. However, while inverse matching from 3D points to image features can find correct matches quickly through search prioritization, it has difficulty on larger models. In their most recent work by Li et al. [19], the method of "inverse-matching" is used in conjunction with the traditional 2D-to-3D "forward matching" to create a bi-directional matching scheme for enhancing the similarities of correspondences. In this paper, we formulate the 3D point cloud reduction problem as a mixed integer quadratic programming problem. Our method produces the optimal solution given the objective function, whereas existing methods suffer from sub-optimality. Also our formulation allows us to design and modify the objective function depending on applications.

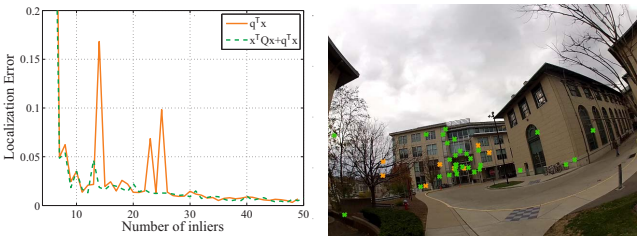
3. Method

A camera pose in 3D (translation and orientation) can be estimated by finding correspondences between 3D points from a database and 2D points from the image captured by the camera. A state-of-the-art perspective-*n*-point algorithm allows us to recover the pose parameters minimally from four 2D-3D point correspondences [15] given the camera intrinsic parameters. Finding the correspondences is computationally prohibitive for real-time image localization when the number of 3D points in the database is large [1, 8, 19]. In Section 3.1, we introduce an algorithm to intelligently reduce the number of 3D points in the database based on training images. Using this reduced database, we present a novel criterion for finding 2D-3D correspondences in Section 3.2.

3.1. Database Reduction

Given 3D points in the database, we seek a compact subset of the points such that at least b number of points in the subset are visible from a query image. This problem is closely related to the maximum coverage problem, or K-cover problem. In this section, we start from the well known maximum coverage problem and derive our objective function in the form of a mixed-integer quadratic programming problem.

Let $\mathcal{S} = \{\mathbf{p}_i\}_{i=1,\cdots,N}$ be a set of all 3D points, $\mathbf{p}$, where N is the number of points and let $\mathcal{S}_{\text{in}} \subset \mathcal{S}$ be a subset of points where $\mathbf{p}_i$ is discarded if $\mathbf{p}_i \notin \mathcal{S}_{\text{in}}$. We want to find a $\mathcal{S}_{\text{in}}$ that satisfies a user-defined criterion.



(a) Accuracy (b) Feature distribution

Figure 1. A quadratic term in Equation (4) allows us to design the binary relation between two points. We penalize co-occurring feature points for accurate camera pose localization. (a) The quadratic term in Equation (4) contributes to produce a more stable/accurate localization. (b) Green points are points that are selected by solving Equation (4) while orange points are selected by solving Equation (3). By penalizing the co-occurance, the quadratic term encourages widely distributed correspondence configuration. Variance of point distribution is 303.65 and 174.24 for green and orange points, respectively.

The maximum coverage problem is to find K points that maximize the number of correspondences, i.e.,

$$\begin{aligned} \underset{\mathbf{x}}{\text{maximize}} \;\; & \mathbf{1}^\mathsf{T}\mathbf{A}\mathbf{x} \\ \text{subject to} \;\; & \mathbf{1}^\mathsf{T}\mathbf{x} \leq K \\ & \mathbf{x} \in \{0,1\}^N \end{aligned} \quad , \tag{1}$$

where $\mathbf{x}$ is a binary vector whose i^{th} element is one if the $\mathbf{p}_i$ is kept or zero otherwise and $\mathbf{1}$ is a vector whose element is one, i.e., if $\mathbf{p}_i \in \mathcal{S}_{\text{in}}$, then $\mathbf{x}_i = 1$. $\mathbf{A}$ is an F by N visibility matrix where F is the number of images, i.e., if the j^{th} point is visible from the i^{th} image, $\mathbf{A}_{ij} = 1$, otherwise $\mathbf{A}_{ij} = 0$. Note that $\mathbf{A}\mathbf{x}$ is a vector whose i^{th} element is the number of correspondences for the i^{th} image. Equation (1) does not explicitly encode the fact that there must be at least b number of correspondences to estimate the image pose parameters in 3D. By incorporating such a constraint, the problem can be formulated as:

$$\begin{aligned} \underset{\mathbf{x}}{\text{minimize}} \;\; & \mathbf{1}^\mathsf{T}\mathbf{x} \\ \text{subject to} \;\; & \mathbf{A}\mathbf{x} \geq b\mathbf{1} \\ & \mathbf{x} \in \{0,1\}^N \end{aligned} \quad , \tag{2}$$

where b is the minimum number of 2D-3D correspondences to be kept for each image. Equation (2) directly minimizes the number of elements in $\mathcal{S}_{\text{in}}$ while maintaining the number of correspondences larger than b. Therefore, the smallest subset of 3D points can be obtained.

When prior knowledge about a 3D point is available, a different weight on each point can produce a solution (Equation (2) has the same weight on all points.). For instance, a point that is frequently matched to the training images is more favorable because it is more likely to be

matched to a query image. To weigh on each point individually, Equation (2) can be modified as:

$$\begin{aligned} \underset{\mathbf{x}}{\text{minimize}} \;\; & \mathbf{q}^\mathsf{T}\mathbf{x} \\ \text{subject to} \;\; & \begin{matrix} \mathbf{A}\mathbf{x} \geq b\mathbf{1} \\ \mathbf{x} \in \{0,1\}^N \end{matrix} \;\; , \end{aligned} \tag{3}$$

where $\mathbf{q}$ is a weight vector. Binary relation between two correspondences can also be considered. For example, spatially widely distributed 3D point configuration is favorable because it can enhance the accuracy of image localization; the uncertainty of pose estimation becomes higher when the baseline between 3D points becomes smaller. By encouraging high 2D/3D Euclidean distance between two correspondences or penalizing co-occurance in the same image, a desirable solution that produces highly accurate image localization can be achieved. This relation can be encoded in the form of a binary quadratic programming problem as follows:

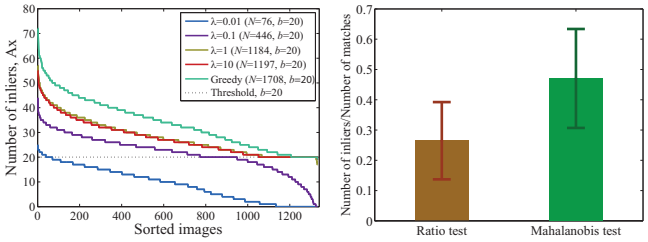
$$\begin{aligned} \underset{\mathbf{x}}{\text{minimize}} \;\; & \frac{1}{2}\mathbf{x}^\mathsf{T}\mathbf{Q}\mathbf{x} + \mathbf{q}^\mathsf{T}\mathbf{x} \\ \text{subject to} \;\; & \begin{matrix} \mathbf{A}\mathbf{x} \geq b\mathbf{1} \\ \mathbf{x} \in \{0,1\}^N \end{matrix} \;\; , \end{aligned} \tag{4}$$

where $\mathbf{Q}$ is a symmetric matrix. $\mathbf{Q}_{ij}$ accounts for a weight on the binary relation between the i^{th} and j^{th} points. Figure 1(a) shows that the quadratic term can contribute towards accurate and robust image localization. This is also observed in the image itself; the solution from quadratic programming produces more widely distributed correspondences.

The inequality constraint $\mathbf{A}\mathbf{x} \geq b\mathbf{1}$ in Equation (2), (3), and (4) is a hard constraint, i.e., if the total number of correspondences of the i^{th} image is less than b, there is no feasible solution. For some training images, the number of correspondences do not need to be greater than b because the images are not informative. The hard constraint can be relaxed by introducing a slack variable as follows:

$$\begin{aligned} \underset{\mathbf{x},\boldsymbol{\xi}}{\text{minimize}} \;\; & \frac{1}{2}\mathbf{x}^\mathsf{T}\mathbf{Q}\mathbf{x} + \mathbf{q}^\mathsf{T}\mathbf{x} + \lambda\mathbf{1}^\mathsf{T}\boldsymbol{\xi} \\ \text{subject to} \;\; & \begin{matrix} \mathbf{A}\mathbf{x} \geq b\mathbf{1} - \boldsymbol{\xi} \\ \mathbf{x} \in \{0,1\}^N \\ \boldsymbol{\xi} \in \{0,\mathbb{Z}_+\}^F \end{matrix} \;\; , \end{aligned} \tag{5}$$

where $\boldsymbol{\xi}$ is a semi-positive integer vector that allows small violation of the inequality constraint and λ determines the hardness of the inequality constraint. When λ goes to infinite, the constraint becomes hard, i.e., the same as Equation (4), and when λ approaches zero, the constraint becomes soft. This slack variable also prevents the solution from overfitting to the training images. As shown in Figure 2(a), λ controls the shape of graph. $\lambda = 0.1$ well generalizes the shape of the testing graph shown in Figure 4(a).



(a) Slack contribution (b) Distance measure comparison

Figure 2. (a) Slack variable in Equation (5) allows small violation of the inequality constraint. This prevents the solution to overfit the training data. When $\lambda = 0.1$, it predicts the underlying distribution of test images well. Compare the shape of the testing set graph shown in Figure 4(c) and 4(a). (b) Mahalanobis distance measure rejects more false positive matches, i.e., fewer noisy matches. As a result, the RANSAC success rate becomes higher.

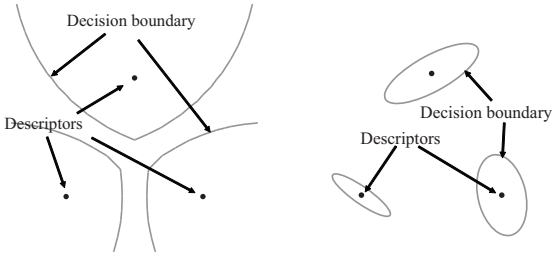
$\mathbf{Q}$, $\mathbf{q}$, and λ are user designable parameters based on applications.

3.2. Matching Process

The number of correspondences per image decreases as the size of database becomes smaller. The database size reduction accelerates the matching process whereas finding correct 2D-3D matches becomes more challenging. The number of inliers is significantly lower than the number of outliers($< 5\%$). We intend the number of correspondences to be similar to b when solving Equation (5), where b is a small number compared to the number of all feature points in a query image. Lower probability of choosing the inlier requires many iterations for RANSAC [7]. To have a 95% success rate, the required number of iterations is more than 4.8×10^5 for 5% of inliers using the four point method [15].

A previous method for finding correct matches relies on a discriminative measure, i.e., the ratio test [20] to reject false positive matches. If the ratio between distances of the nearest neighbor and the second nearest neighbor is smaller than a threshold (0.7 is popularly used), the 2D and 3D points are considered to be a correct match. This ratio test ensures that the match is distinctive. However, the ratio criterion produces many false positives for our case where correct matches are extremely sparse among all detected feature points in the query image. To reduce the number of false positive matches, Li et al. [18] used a bidirectional matching method with different ratio tests, and Li et al. [19] applied a co-occurance prior distribution for false positive rejection. Unlike previous methods, we utilize a generative measure, Mahalanobis distance, by learning a distribution of descriptors directly from the training images. This approach can provide a more strict criterion for false positive detection.

A 3D point associates with at least two similar descrip-



(a) Discriminative measure (b) Generative measure

Figure 3. We present a novel method to reject false positive matches by learning a distribution of descriptors from the training images. (a) Previous methods have used a discriminative measure, the ratio test. If the ratio between distances of the nearest and the second nearest neighbor is less than 0.7, it is considered to be a match. (b) Instead, we learn the covariance of descriptors. This measure results in a more strict criterion, which enables us to reject false positive matches. We show the decision boundary of matching.

tors because two images are minimally required to be triangulated to estimate the 3D point. In many cases, a point corresponds to more than two descriptors from the training images. From many descriptors associated with a single 3D point, we learn the covariance of the descriptors, $\mathbf{C_y} = \mathbf{Y}\mathbf{Y}^\mathsf{T}$, where $\mathbf{Y} = \begin{bmatrix} \mathbf{y}_1 - \bar{\mathbf{y}} & \cdots & \mathbf{y}_m - \bar{\mathbf{y}} \end{bmatrix}$, $\mathbf{y}_i$ is a descriptor vector, $\bar{\mathbf{y}}$ is the mean descriptor vector, and m is the number of the descriptors associated with the 3D point. The covariance enables us to define a Mahalanobis distance between 3D and 2D descriptors to measure the similarity, i.e.,

$$d(\bar{\mathbf{y}}, \mathbf{z}) = \sqrt{(\mathbf{z} - \bar{\mathbf{y}})^\mathsf{T} \mathbf{C}_\mathbf{y}^{-1} (\mathbf{z} - \bar{\mathbf{y}})}, \tag{6}$$

where $\mathbf{z}$ is a query descriptor. The main benefit of this distance is that we can estimate the probability of the match given the query descriptor. Also the distance is a normalized measure, therefore a single threshold can be applied to all points to reject false positives. This generative measure is a much tighter criterion than the discriminative measure in previous works while generalizing well across all matches as long as the query image is drawn from the distribution of the training images. Figure 2(b) shows the effectiveness of our distance measure. The ratio between the number of inliers and matches is higher than the ratio test, which results in higher success rate for RANSAC given the same number of iterations.

In practice, we look for bidirectionally consistent matches: we match from a query image to the 3D database and vice versa, and keep only the consistent matches. From these consistent matches, we apply Mahalanobis distance test based on Equation (6), i.e., if $d(\bar{\mathbf{y}}, \mathbf{z}) > t$, we reject the match. t is a threshold. For the covariance matrix, we use $\widetilde{\mathbf{C}}_\mathbf{y} = \mathbf{C}_\mathbf{y} + \mathbf{I}$ to avoid ill-conditioned inverse operation. Instead of representing a distribution of descriptors with a D by D matrix where D is the feature dimension, only diagonal elements in $\widetilde{\mathbf{C}}_\mathbf{y}$ is used (off-diagonal terms are extremely sparse) for compute and memory efficiency. We use the variance of each element of the descriptors by assuming that there is no correlation across elements in a descriptor.

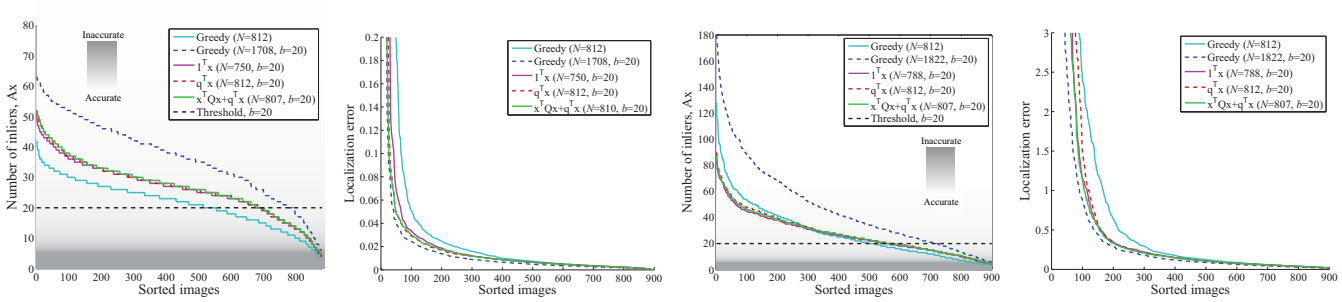
4. Result

We test our algorithm on four real datasets: two from our own data collection and two from standard benchmark datasets provided by Li et al. [18]. All quantitative evaluation is compared to the baseline greedy algorithm presented by Li et al. [18]. For our data collection, we reconstruct 3D points from videos that exhaustively scan the space of interest. We use a standard structure from motion algorithm to reconstruct the 3D scene. We ask people to wear head-mounted cameras and move freely in the space. We use one of the videos as the test set and the rest of the videos as the training set. Two sequences, outdoor and indoor, are used for our evaluation. For the standard datasets, we used Dubrovnik and Rome data reconstructed by internet photos. We use known camera intrinsic parameters available from the meta data and estimate image pose in 3D using the efficient perspective-*n*-point algorithm [15]. While the minimal number of correspondences for image localization is four given intrinsic parameters, accurate and robust estimation typically requires more correspondences. Only when the number of inliers from RANSAC is higher than 10, the image is considered to be registered. The element of the unary weight, $\mathbf{q}$, is set to $\mathbf{q}_i = q_{\max} - q_i$, where q_i is the number of times the i^{th} point is visible and $q_{\max} = \max\{q_i\}_{i=1,\cdots,N}$. For the binary weight, $\mathbf{Q}$, we penalize the co-occurance of two points by linearly increasing the $\mathbf{Q}_{ij}$ whenever the i^{th} and j^{th} points are observed from the same image. To solve the mixed-integer quadratic programming problem, we use the commercial optimization software Gurobi[1].

4.1. Our datasets: Outdoor and Indoor

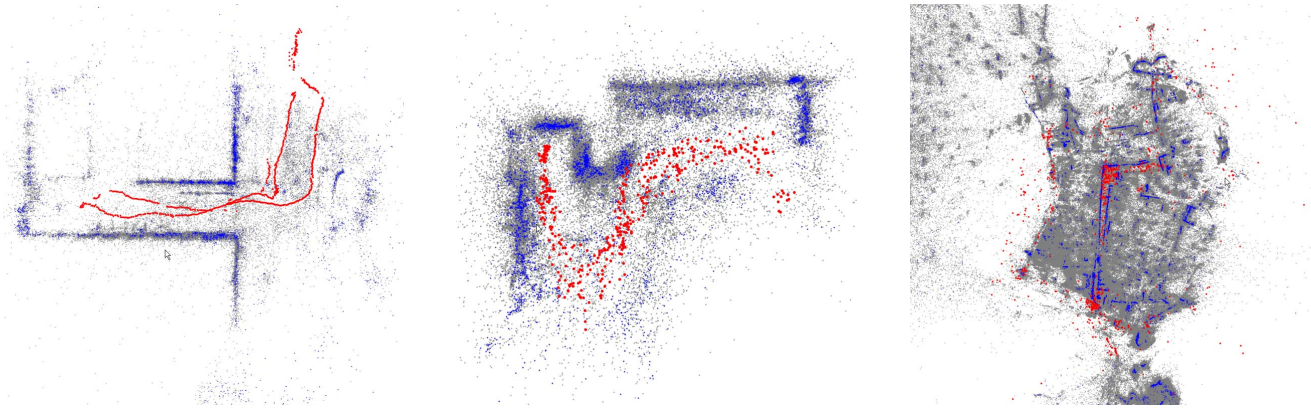
We set the threshold, b, for inequality constraint in Equation (4) to 20 to ensure that the number of correspondences is greater than 10 (registration threshold). Figure 4(a) shows the number of inliers after RANSAC (images are sorted in descending order based on the number of inliers). Our algorithm outperforms the greedy algorithm, particularly around 10 to 20 inliers. This is an important range for image localization because the accuracy of the localization degrades significantly within the range (the grey gradient encodes the accuracy of localization in Figure 4(a).). Figure 4(b) shows that the solution obtained by our method is more accurate

[1] http://www.gurobi.com



(a) Indoor sequence: Inlier (b) Indoor sequence: Error (c) Ourdoor sequence: Inlier (d) Ourdoor sequence: Error

Figure 4. (a) and (c): our method outperforms a greedy solution which has comparable size, particularly around 10 to 20 of the number of inliers where camera localization becomes inaccurate. The grey gradient at the bottom of the graph shows the accuracy changes as the number of inliers increase. To get the same threshold, b, with our method, the greedy solution is more than twice larger than ours. (b) and (d): image localization accuracy is measured. Our method consistently performs better than comparable size of the greedy solution. Note that the quadratic term produces the highest accuracy among our solutions.



(a) Outdoor sequence (b) Indoor sequence (c) Dubrovnik

Figure 5. Our method significantly reduces the number of points. Grey points are original 3D points, blue points are the reduced subset, and red points are reconstructed image locations.

than the greedy method. The same observation also applies to the outdoor sequence as shown in Figure 4(c) and Figure 4(d). In both sequences, our method achieves over 90% image registration rate. Figure 5(a) and 5(b) show the qualitative results of our method.

4.2. Standard Datasets: Dubrovnik and Rome

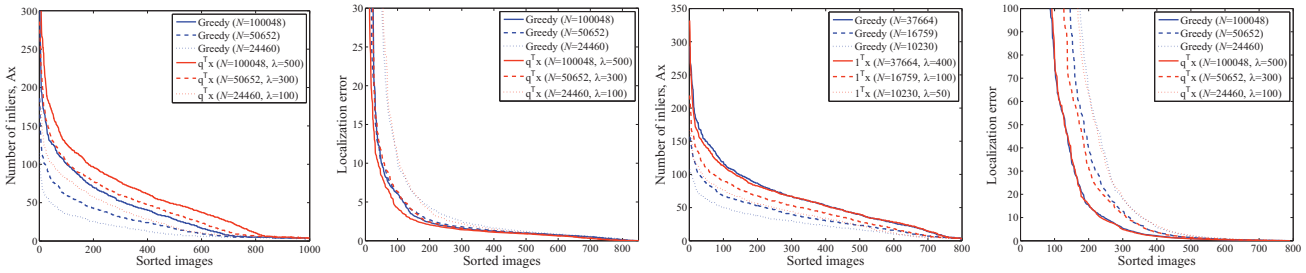
We apply our algorithm to two standard datasets provided by Li et al. [18]. While these datasets contain many images (6000 for Dubrovnik and 16000 for Rome), the number of image samples per area is lower than our datasets, i.e., covering area per image is fairly large. To account for sparse training images, we raise b to 80. Due to memory constraint, we only employ binary linear programming, i.e., Equation (2) and (3). Figure 6(a) shows our method can register about 80% of query images and consistently outperforms the greedy solution. Our localization error is also lower than the greedy algorithm as shown in Figure 6(b). Our experiments on Dubrovnik data produce similar observations as shown in Figure 6(c) and 6(d). Detailed comparison is listed in Table 1 and 2. Figure 5(c) shows camera registration (red points) using the reduced set of 3D point cloud (blue points).

Table 1. Registration performance for Rome data (885 query images)

	N	$\mathbf{1}^\mathsf{T}\mathbf{x}$	Greedy		N	$\mathbf{q}^\mathsf{T}\mathbf{x}$	Greedy
λ=1	94708	786	662	λ=500	100048	806	676
λ=0.5	57263	730	540	λ=300	50652	722	603
λ=0.1	20157	584	416	λ=100	24460	620	466

Table 2. Registration performance for Dubrovnik data (780 query images)

	N	$\mathbf{1}^\mathsf{T}\mathbf{x}$	Greedy		N	$\mathbf{q}^\mathsf{T}\mathbf{x}$	Greedy
λ=0.5	46820	711	716	λ=400	37664	711	704
λ=0.25	17436	602	604	λ=100	16759	649	637
λ=0.125	10502	560	536	λ=50	10230	598	576



(a) Rome: Inlier (b) Rome: Error (c) Dubrovnik: Inlier (d) Dubrovnik: Error

Figure 6. (a) and (c): we reduce the 3D points by varying slack parameter, λ. As λ decreases, the reduction rate significantly increases. The difference between the greedy method and ours is more emphasized. For all experiments, our method consistently register more images than the greedy method. (b) and (d): our method is more accurate.

5. Discussion

In this paper, we present an algorithm to find a compact subset of 3D points based on view-statistics for efficient image localization. This compact subset of 3D points speeds up the matching process and results in a comparable image localization rate. We formulate the problem as a mixed-integer quadratic programming problem. This formulation allows a user to design the objective function depending on applications unlike existing greedy methods. We demonstrate that our solution outperforms a related greedy method on standard datasets. We also introduce a method to calibrate descriptors associated with a 3D point by learning a distribution of the descriptors directly from the training images. This generative measure provides higher success rate on a RANSAC based matching.

Our formulation of the database reduction problem allows a user to design the objective function based on the application. This enables users to manipulate the desired output easily and to incorporate domain knowledge. One interesting future direction might be to learn $\mathbf{Q}$ and $\mathbf{q}$ for different target camera motion and different types of databases. For example, camera motion in a vehicle is different from that of a first-person camera. The reduced point set should reflect and benefit from the prior knowledge about camera motion. Also if the database contains a subset of GPS-tagged images, weights on points corresponding to the tagged images should be different. The optimal weight can be learned from different datasets.

We show that the data reduction problem is inherently a mixed integer quadratic programming problem. It is a non-convex optimization that requires high capacity of memory and heavy computation where the global optimum cannot be guaranteed. For our outdoor dataset (30,000 points), it took 30 minutes on an Intel i7 CPU@2.5GHz with 16GB memory to solve the mixed integer quadratic programming problem using a branch and bound method. Such memory requirement and computation are apparent limitations of our method. However, efficient optimization algorithms have solved large scale problems via spectral or semidefinite relaxation [23]. Rapid advances in computing power and memory capacity will also enable us to handle the problem at large scale.

References

[1] S. Agarwal, N. Snavely, I. Simon, S. Seitz, and R. Szeliski. Building rome in a day. In *ICCV*, 2009. 1, 2, 3

[2] G. Baatz, K. Köser, D. Chen, R. Grzeszczuk, and M. Pollefeys. Handling urban location recognition as a 2d homothetic problem. In *ECCV*, 2010. 2

[3] G. Baatz, O. Saurer, K. Köser, and M. Pollefeys. Large scale visual geo-localization of images in mountainous terrain. In *ECCV*, 2012. 2

[4] D. M. Chen, G. Baatz, K. Köser, S. S. Tsai, R. Vedantham, T. Pylvänäinen, K. Roimela, X. Chen, J. Bach, M. Pollefeys, B. Girod, and R. Grzeszczuk. Handling urban location recognition as a 2d homothetic problem. In *CVPR*, 2011. 2

[5] F. Cozman and E. Krotkov. Robot localization using a computer vision sextant. In *ICRA*, 1995. 2

[6] M. Cummins and P. Newman. FAB-MAP: Probabilistic localization and mapping in the space of appearance. *IJRR*, 2008. 2

[7] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 1981. 4

[8] J.-M. Frahm, P. Georgel, D. Gallup, T. Johnson, R. Raguram, C. Wu, Y.-H. Jen, E. Dunn, B. Clipp, S. Lazebnik, and M. Pollefeys. Building rome on a cloudless day. In *ECCV*, 2010. 2, 3

[9] M. Havlena, A. Torii, and T. Pajdla. Efficient structure from motion by graph optimization. In *ECCV*, 2010. 2

[10] J. Hays and A. A. Efros. Im2gps: Estimating geographic information from a single image. In *CVRP*, 2008. 2

[11] A. Irschara, C. Zach, J.-M. Frahm, and H. Bischof. From structure-from-motion point clouds to fast location recognition. In *CVPR*, 2009. 1, 2, 3

[12] N. Jacobs, S. Satkin, N. Roman, R. Speyer, and R. Pless. Geolocating static cameras. In *ICCV*, 2009. 2

[13] E. Kalogerakis, O. Vesselova, J. Hays, A. A. Efros, and A. Hertzmann. Image sequence geolocation with human travel priors. In *ICCV*, 2009. 2

[14] R. M. Karp. Reducibility among combinatorial problems. *Complexity of Computer Computations*, 1972. 1

[15] V. Lepetit, F. Moreno-Noguer, and P. Fua. EPnP: An accurate O(n) solution to the PnP problem. *IJCV*, 2009. 3, 4, 5

[16] F. Li and J. Kosecka. Probabilistic location recognition using reduced feature set. In *ICRA*, 2006. 2

[17] X. Li, C. Wu, C. Zach, S. Lazebnik, and J.-M. Frahm. Modeling and recognition of landmark image collection using iconic scene graphs. In *ECCV*, 2008. 2

[18] Y. Li, N. Snavely, and D. Huttenlocher. Location recognition using priotized feature matching. In *ECCV*, 2010. 1, 3, 4, 5, 6

[19] Y. Li, N. Snavely, D. Huttenlocher, and P. Fua. Worldwide pose estimation using 3d point clouds. In *ECCV*, 2012. 3, 4

[20] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 2004. 1, 2, 4

[21] K. Ni, A. Kannan, A. Criminisi, and J. Winn. Epitomic location recognition. *TPAMI*, 2009. 2

[22] D. Nistér and H. Stewénius. Scalable recognition with a vocabulary tree. In *CVPR*, 2006. 2

[23] C. Olsson, A. P. Eriksson, and F. Kahl. Solving large scale binary quadratic problems: Spectral methods vs. semidefinite programming. In *CVPR*, 2007. 7

[24] D. Robertson and R. Cipolla. An image-based system for urban navigation. In *BMVC*, 2004. 2

[25] G. Schindler, M. Brown, and R. Szeliski. City-scale location recognition. In *ICCV*, 2007. 2

[26] S. Se, D. Lowe, and J. Little. Global localization using distinctive visual features. In *IROS*, 2002. 2

[27] I. Simon, N. Snavely, and S. M. Seitz. Scene summarization for online image collections. In *ICCV*, 2007. 2

[28] N. Snavely, S. M. Seitz, and R. Szeliski. Photo tourism: Exploring photo collections in 3d. *TOG(SIGGRAPH)*, 2006. 2

[29] N. Snavely, S. M. Seitz, and R. Szeliski. Skeletal graphs for efficient structure from motion. In *CVPR*, 2008. 2

[30] A. Torralba, K. P. Murphy, W. T. Freeman, and M. A. Rubin. Context-based vision system for place and object recognition. In *ICCV*, 2003. 2

[31] A. R. Zamir and M. Shah. Accurate image localization based on google maps street view. In *ECCV*, 2010. 2

[32] W. Zhang and J. Kosecka. Image based localization in urban environments. In *3DPVT*, 2006. 2