Minimal Scene Descriptions from Structure from Motion Models

Song Cao Noah Snavely
Cornell University

Abstract

How much data do we need to describe a location? We explore this question in the context of 3D scene reconstructions created from running structure from motion on large Internet photo collections, where reconstructions can contain many millions of 3D points. We consider several methods for computing much more compact representations of such reconstructions for the task of location recognition, with the goal of maintaining good performance with very small models. In particular, we introduce a new method for computing compact models that takes into account both image-point relationships and feature distinctiveness, and we show that this method produces small models that yield better recognition performance than previous model reduction techniques.

1. Introduction

In recent years, the increasing availability of online tourist photos has stimulated a line of work that utilizes structure-from-motion techniques to construct large-scale databases of images and 3D point clouds [9, 1], for a variety of applications, including location recognition [10, 2, 3, 7, 4]. These location recognition methods often directly match features (such as SIFT [5]) in a query image to descriptors associated with 3D points. These databases of 3D points, however, can be very large—ranging in size from a few million points in a single location, to hundreds of millions when multiple places are considered together [4]. For purposes of modeling and visualization, the denser the 3D points the better. However, for other applications, such as recognition, there are advantages in having fewer points, such as reduced memory and computation requirements. This brings up an interesting question: how much data do we need to describe a location? What is a minimal description of a place?

One way to make this question concrete is to define it as a visibility covering problem [3, 6]: every possible image that one could take of the location should see some minimal number of 3D points stored in the reconstruction. Such a covering constraint makes it likely that a new image of the scene will match a sufficient number of 3D points to enable pose estimation. Based on this idea, prior methods have used the visibility relationships between images and points in the database to compute reduced 3D point sets that cover the database images. However, another important factor is **distinctiveness**: in order to ensure accurate matching, one should select a subset of points that are distinct (rather than selecting points with very similar appearance). In this paper, we show that by computing a reduced scene description that takes into account both coverage and distinctiveness, one can compute very compact models that maintain good recognition performance.

We incorporate these considerations into a new point selection algorithm that predicts how well new images will be recognized using a probabilistic approach. We evaluate our algorithm on several standard location recognition benchmarks, and show that our computed scene representations consistently yield higher recognition performance compared to previous model reduction techniques.

2. Computing Minimal Point Sets

We begin by running structure from motion (SfM) to reconstruct one or more scenes that form a database for use in recognizing and posing new images [1]. The result of running SfM on an image set $\mathcal{I}$ of size m is a 3D point set $\mathcal{P}$ of size n, (typically $n \gg m$), as well as a *visibility matrix* M of size $m \times n$ defining the visibility relationships between images and points, where $M_{ij} = 1$ if point P_j is visible in image I_i in the reconstructed 3D model, and $M_{ij} = 0$ otherwise. Our goal is to compute a more compact database with a much smaller set of points $\mathcal{P}' \subset \mathcal{P}$, such that $\mathcal{P}'$ captures as much of the information in the full model as possible. In particular, we wish to be able to correctly register as many new query images to the subset $\mathcal{P}'$ as possible.

K-cover algorithm. The prior work of Li *et al*. [3] formulate this as a *K-cover (KC) problem* on the visibility graph $\mathcal{G}$: select a minimum subset of points such that each database image sees at least K points in the subset. Finding such a minimum set is a combinatorially hard problem, and so they use a greedy algorithm that starts with the empty set, and incrementally adds the next point P_j that maximizes the *gain* in coverage achieved by adding P_j to the current set $\mathcal{P}'$.

2.1. Appearance-aware point set selection

In order to increase distinctiveness and the probability of query features matching to the correct database point, we select points that are far away from each other in descriptor space. Like Li *et al*. [3], we use agreedy selection algorithm that adds points to $\mathcal{P}'$ sequentially, and so when computing the gain of a point P_j under consideration, we implement this strategy by down-weighting a point's gain according to its minimum distance to the current set of selected points $\mathcal{P}'$.

Thus, the point set selected by this method (which we call **KCD**, or "K-cover with distinctiveness") is "sparser" in descriptor space, which will tend to decrease the rate of false matches in the feature matching phase of the recognition pipeline. We use this appearance-aware selection method to seed our probabilistic point selection algorithm, which we describe next.

2.2. Probabilistic K-cover algorithm

Using our KCD algorithm, we can select an initial covering set of points. This allows us to bootstrap a second, probabilistic point selection method (called **KCP**). Rather than treating the visibility matrix as binary, our probabilistic approach treats this matrix as a set of noisy observations of visibility, and selects a small number of additional points to add to $\mathcal{P}'$ such that the number of images that satisfy $\Pr(v_{i,\mathcal{P}'} \geq K) \geq p_{\min}$ is as large as possible. That is, unlike the K-cover algorithm, which seeks to combinatorially "cover" the images at least K times, we set a minimum probability value $p_{\min}$ and our goal is to achieve $\Pr(v_{i,\mathcal{P}'} \geq K) \geq p_{\min}$ for each image I_i. Like the K-cover algorithm, we use a greedy approach, but choosing the point P_{j^*} that maximizes *expected gain*.

3. Experiments

We evaluate on the Dubrovnik dataset of Li *et al*. [3], the Aachen dataset of Sattler *et al*. [8], and the much larger Landmarks dataset [4]. We evaluate three approaches to computing minimal scene descriptions: the K-cover algorithm (KC) [3], our initial point set selection algorithm only (KCD), and our full approach (KCP). All methods output a list of points to keep in the original 3D point cloud database. We use each subset of points to construct a reduced database, and use the algorithm of [4] to register the query images for each dataset. We record the percentage of successfully registered images and use it as a measure of how well the point set represents the original database in Table 3.

In all datasets, KCD improves the performance compared to the K-cover algorithm for nearly all values of K. The improvement is especially significant when K is low (and hence the number of selected points is small). Our full approach (KCP) consistently outperforms the K-cover algorithm, and further improves on the gains achieved by our KCD algorithm.

Dubrovnik Dataset [3]				
# query images: 800, registered by full set: 99.50%				
K	12 (9)	20 (12)	30 (20)	50 (35)
# points	5,788	10,349	17,241	31,752
% points	0.31%	0.55%	0.91%	1.68%
KC	58.00%	77.06%	86.00%	91.81%
KCD	62.88%	78.88%	**87.38%**	92.50%
KCP	**64.25%**	**79.13%**	87.25%	**93.38%**
Aachen Dataset [8]				
# query images: 369, registered by full set: 88.08%				
K	30 (20)	50 (32)	80 (52)	100 (65)
# points	13,299	23,675	40,377	52,161
% points	0.67%	1.20%	2.04%	2.63%
KC	50.95%	62.06%	66.40%	71.27%
KCD	54.20%	63.14%	69.38%	72.36%
KCP	**56.37%**	**64.23%**	**70.19%**	**73.98%**
Landmarks Dataset [4]				
# query images: 10,000, registered by full set: 94.33%				
K	6 (4)	9 (6)	12 (9)	20 (12)
# points	140,306	222,161	311,035	571,864
% points	0.37%	0.58%	0.81%	1.50%
KC	44.84%	59.86%	69.56%	81.06%
KCD	45.45%	61.26%	70.59%	81.04%
KCP	**45.90%**	**61.50%**	**71.87%**	**81.45%**

Table 1. **Registration performance on various datasets.** For comparison, we also show the performance of [4] using the full set of input points.

References

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

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

[3] Y. Li, N. Snavely, and D. Huttenlocher. Location recognition using prioritized feature matching. In *ECCV*, 2010. 1, 2

[4] Y. Li, N. Snavely, D. Huttenlocher, and P. Fua. Worldwide pose estimation using 3d point clouds. In *ECCV*, 2012. 1, 2

[5] D. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 2004. 1

[6] H. S. Park, Y. Wang, E. Nurvitadhi, J. C. Hoe, Y. Sheikh, and M. Chen. 3d point cloud reduction using mixed-integer quadratic programming. In *CVPR Workshops*, 2013. 1

[7] T. Sattler, B. Leibe, and L. Kobbelt. Fast image-based localization using direct 2D-to-3D matching. In *ICCV*, 2011. 1

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

[9] N. Snavely, S. Seitz, and R. Szeliski. Photo tourism: exploring photo collections in 3d. In *SIGGRAPH*, 2006. 1

[10] W. Zhang and J. Kosecka. Image based localization in urban environments. In *Int. Symp. on 3DPVT*, 2006. 1