Learning Cross-domain Information Transfer for Location Recognition and Clustering

Raghuraman Gopalan
Video and Multimedia Technologies Research Dept., AT&T Labs-Research, Middletown, NJ 07748 USA
raghuram@research.att.com

Abstract

Estimating geographic location from images is a challenging problem that is receiving recent attention. In contrast to many existing methods that primarily model discriminative information corresponding to different locations, we propose joint learning of information that images across locations share and vary upon. Starting with generative and discriminative subspaces pertaining to domains, *which are obtained by a hierarchical grouping of images from adjacent locations, we present a top-down approach that first models cross-domain information transfer by utilizing the geometry of these subspaces, and then encodes the model results onto individual images to infer their location. We report competitive results for location recognition and clustering on two public datasets, im2GPS and San Francisco, and empirically validate the utility of various design choices involved in the approach.*

(a) (b)

Figure 1. While images in (a) correspond to familiar locations that either have distinct visual features or have good exposure amongst the general public, the location of images in (b) is hard to infer. One intuitive way to address such cases is to analyze how those images are *relatively* similar to *and* different from other known locations so that a meaningful location estimate can be obtained. We pursue such a goal in this work using tools pertaining to subspace geometry. *All figures are best viewed in color.*

1. Introduction

Image-based identification of locations is an important high-level vision problem that augments the potential of pervasive computing. It compliments techniques that heavily rely on GPS information (eg. [15]), which could either be noisy or missing depending on the location of interest, and application areas such as surveillance. While preliminary work on this problem started atleast two decades ago [20, 25], only in the recent years have we seen substantial progress [26, 31, 23] partly due to large availability of data and the emergence of mobile vision applications.

Most existing methods for location recognition follow the paradigm of discriminative modeling for feature selection and classification. For instance, [11] used several low-level features that could distinguish images across locations and used a nearest-neighbor classifier to estimate query locations from a large data set. [21] proposed epitomic feature analysis that captures appearance and geometric structure of environments while allowing for variations due to motion and occlusion related effects. Utility of 3D information corresponding to locations was investigated by [7, 12, 1]. In addition to robust feature descriptors, there have been several studies on efficient schemes for classification and retrieval of location queries. [18] presented an adaptive, prioritized feature matching technique that learns reliable features with certain view independency for better localization. Scalable vocabulary tree coding algorithms were presented by [22, 13], while [16] modeled landmark image collections using iconic scene graphs. Image features that are confusing from a place recognition perspective was studied by [14], and [5] addressed obtaining discriminative features that are geographically informative, while occurring frequently at the same time. There have also been efforts that provide landmark search engines for web-scale image collections [32] and for mobile vision applications [3]. Besides modeling location specific information, some studies have examined the utility of complimentary information provided by other data modalities. While [19] recognized locations from consumer photos by jointly modeling contextual information conveyed by people and events in those data collections, the advantage of using user-provided tags was illustrated by [17, 27].

Discriminative approaches, however, do not entirely address an important problem in location recognition that is illustrated in Figure 1. While images in Figure 1(a) correspond to famous locations that are visually very distinct or popular enough among the public to be recognized easily, the location of images in Figure 1(b) is hard to be inferred since neither are these images popular locations, nor do they have unique distinguishing features. One feasible way to obtain an *approximate* location estimate of these images is to *jointly* analyze the properties they have in common to *and* vary from other well known locations. An example would be the image of a downtown in Figure 1(b) where the presence of skyscrapers suggests that it should correspond to a urban locality and *not* semi-urban or rural, and the presence of large water bodies further helps narrowing down amongst the potential urban location possibilities (eg. it could be *somewhere* in New York City, but *not* Phoenix). Such an analysis should also account for the fact that the visual and location information of images do not always correlate for instance, one could have images that look very much alike but correspond to vastly different geographic areas.

We address this problem by pursuing a top-down approach where given a set of training images representative of different locations, we first group the images into different *domains* based on location adjacency. We then derive generative and discriminative subspaces of same dimensions from these domains, and motivated by [9], we model cross-domain transfer of similar (*resp.* distinct) information by pursuing a Grassmann manifold interpretation of the space spanned by these generative (*resp.* discriminative) subspaces. We finally embed the effect of this transformation onto images from training and query, and perform location inference in both recognition and clustering settings. The motive behind this is to account for possible lack of correlation between location and visual information of images, whereby the creation of domains offer *location*-based support and the subsequent operations account for modeling *visual* (dis-)similarities in a top-down fashion (from domains to individual images).

2. Proposed Approach

Let us start with the problem setting. Assume that we have n training images spread across m different locations; $\mathcal{X} = \{(x_i, y_i)\}_{i=1}^{n}$, where $x_i \in \mathbb{R}^d$ denotes the $d-$dimensional feature descriptor corresponding to the i^{th} training image, and $y_i \in \{1, 2, ..., m\}$ denotes its location (i.e. latitude and longitude co-ordinates). Now given a query image x_t, the goal of this work is to estimate its location $y_t = f_2(f_1(\mathcal{X}))$ where f_1 models information transfer across domains (that are created by grouping x_i based on their location y_i) and f_2 denotes the subsequent classification or clustering mechanism. More details are provided in the following sub-sections.

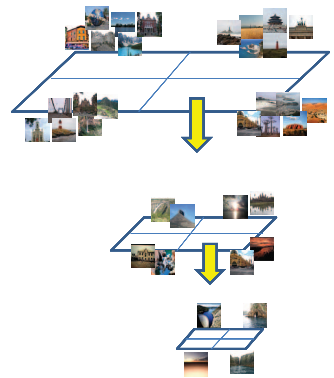


Figure 2. Assigning images from $\mathcal{X}$ into domains $\mathcal{D}$ in a three-level hierarchical manner, and organizing the domains into groups $\mathcal{G}$ for further analysis. Each group contains four domains within which generative and discriminative subspaces are analyzed for cross-domain information transfer. Such an analysis on groups in all three levels convey top-down information on how visually similar and distinct information looks like among image collections that trend progressively from global to local.

2.1. Modeling Cross-domain Information Transfer

Creating Domains: Assuming that $\mathcal{X}$ correspond to images from all over the earth, we flatten the earth and create domains $\mathcal{D}$ in a three-level hierarchical fashion. The first level domains $\mathcal{D}_1$ to $\mathcal{D}_4$ correspond to images from four quadrants (with each quadrant covering 90 degrees in latitude and 180 degrees in longitude) of the flattened earth, and let group $\mathcal{G}_1$ represent the collection all these four domains. The second level domains $\mathcal{D}_5$ to $\mathcal{D}_{20}$ are obtained by splitting each first level domain into four quadrants, and we thus obtain four groups $\mathcal{G}_i = \{\mathcal{D}_i\}_{4^*(i-1)+1}^{4^*i}, i = 2 \textit{ to } 5$. Similarly we obtain the third level domains $\mathcal{D}_{21}$ to $\mathcal{D}_{84}$ by splitting each of the second level domains into four quadrants, with which we constitute 16 groups. Figure 2 provides an illustration. So we have a total of $c = 84$ domains $\mathcal{D} = \{\mathcal{D}_i\}_{i=1}^{84}$ that are split into 21 groups $\mathcal{G} = \{\mathcal{G}_i\}_{i=1}^{21}$ containing four domains each, which represents image collections pertaining to location neighborhoods that trend progressively from global to local. We now model how visually similar and different information transform across domains within each group $\mathcal{G}_i$.

Subspace Representation of a Domain: We resort to linear subspace representation of data contained within each domain for two reasons: (i) subspaces have widely been used to model data characteristics for many computer vision applications [29, 2], and (ii) there exist a set of analytical tools that can be used to interpolate information across subspaces [6]. Towards this end we obtain generative and discriminative subspaces (that represent holistic and distinct information respectively), of the same inherent dimension $N(< d)$, corresponding to these domains by performing principal component analysis (PCA) [29] and partial least squares (PLS) [30] respectively. We perform PCA on each domain $\mathcal{D}_i$ to obtain a $d \times N$ orthonormal matrix whose column space denotes the generative subspace S_{i1}. We obtain discriminative subspace pertaining to each $\mathcal{D}_i$ by considering a one-vs.-remaining setting (i.e $\mathcal{D}_i$ vs. other three domains in the group that $\mathcal{D}_i$ belongs to) and performing a two-class PLS to obtain a $d \times N$ orthonormal matrix whose column space correspond to the discriminative subspace S_{i2}. While one could use other methods for linear generative and discriminative dimensionality reduction, we chose PCA since it is one of the widely used methods to model generative properties, and PLS since it provides flexibility in choosing the subspace dimensions unlike other discriminative methods such as the linear discriminant analysis [2]. Let $\mathcal{S} = \{S_{i1}\}_{i=1}^{84} \cup \{S_{j2}\}_{j=1}^{84}$ refer to the collection of generative and discriminative subspaces obtained from $\mathcal{G}$.

The problem of modeling cross-domain information now translates into: (i) analyzing the space of these N-dimensional subspaces in $\mathbb{R}^d$ to study the transfer of visually generic (*resp.* distinct) information across generative (*resp.* discriminative) subspaces within each group $\mathcal{G}_i$, and (ii) embedding this information transfer onto each individual training data x_i to obtain a new representation $f_1(x_i)$ that is cognizant of the cross-domain variations.

Grassmann Manifold: Before starting our analysis, we note that the space of subspaces is non-Euclidean, and it can be characterized by the Grassmann manifold [6]. The Grassmannian $\mathbb{G}_{d,N}$ is an analytical manifold which is the space of all N-dimensional subspaces in $\mathbb{R}^d$ containing the origin. Each subspace in the collection $\mathcal{S}$ is a 'point' on this manifold. Analyzing the geometric and statistical properties of this manifold has been addressed by works such as [6, 4]. We now utilize some of these results to model f_1.

2.1.1 Analyzing Information Flow Between Subspaces

We first learn how information transforms across different domains within a group. For this we consider a pair of generative (or discriminative) subspaces in that group, although the following analysis can be extended beyond a pair of subspaces. One geometrically meaningful 'path' to 'connect' such pair of 'points' on the manifold, say[1] S_1 and S_2, is the geodesic between them, which are constant velocity curves on the manifold. By viewing $\mathbb{G}_{d,N}$ as a quotient space of $SO(d)$, the geodesic path in $\mathbb{G}_{d,N}$ starting from S_1 is given by a one-parameter exponential flow [6]: $\Psi(t') = Q\exp(t'B)J$, where exp refers to the matrix exponential, and $Q \in SO(d)$ such that $Q^T S_1 = J$ and $J = \begin{bmatrix} I_N \\ 0_{d-N,N} \end{bmatrix}$. I_N is a $N \times N$ identity matrix, and B is a skew-symmetric, block-diagonal matrix of the form $B = \begin{pmatrix} 0 & A^T \\ -A & 0 \end{pmatrix}$, $A \in \mathbb{R}^{(d-N)\times N}$, where the superscript T denotes matrix transpose, and the sub-matrix A specifies the direction and the speed of geodesic flow. Now to obtain the geodesic flow between S_1 and S_2, we compute the direction matrix A such that the geodesic along that direction, while starting from S_1, reaches S_2 in unit time. A is generally computed using inverse exponential mapping (Algorithm 1).

Given two points S_1 and S_2 on the Grassmannian $\mathbb{G}_{d,N}$.

- Compute the $d \times d$ orthogonal completion Q of S_1.
- Compute the thin CS decomposition of $Q^T S_2$ given by $Q^T S_2 = \begin{pmatrix} X_C \\ Y_C \end{pmatrix} = \begin{pmatrix} V_1 & 0 \\ 0 & \tilde{V}_2 \end{pmatrix} \begin{pmatrix} \Gamma(1) \\ -\Sigma(1) \end{pmatrix} V^T$
- Compute $\{\theta_i\}$ which are given by the arccos and arcsine of the diagonal elements of Γ and Σ respectively, i.e. $\gamma_i = \cos(\theta_i)$, $\sigma_i = \sin(\theta_i)$. Form the diagonal matrix Θ with θ's as diagonal elements.
- Compute $A = \tilde{V}_2 \Theta V_1^T$.

Algorithm 1: Numerical computation of the velocity matrix: The inverse exponential map [8].

Using the information contained in A, we can 'sample' points along the geodesic to understand how information transforms between different domains. This is performed using the exponential map (Algorithm 2), by using the expression for $\Psi(t')$ to obtain intermediate points (subspaces) between S_1 and S_2 by varying the value of t' between 0 and 1. Let N' denote the number of subspaces obtained from a geodesic, which includes S_1, S_2 and all intermediate subspaces sampled between them. This process, when repeated between all pairs of generative (*resp.* discriminative) subspaces, provides a wealth of information on how visually generic (*resp.* distinct) properties transform across different domains. Since we analyze 252 geodesics (footnote 1), we have $c_1 = 252 {*} N'$ subspaces conveying cross-domain information transfer.

2.1.2 Embedded Data Representation

We then embed this information onto the training data by projecting each x_i on all c_1 subspaces to result in a matrix M_i' of size $N \times c_1$. The column vectors of this matrix represent a set of instances that describe x_i relative to cross-domain variations. One way to collectively describe such

[1] We use symbols S_1 and S_2 for sake of clarity. S_1 and S_2 could either correspond to a pair of generative subspaces S_{i1} and S_{j1} within a group, or a pair of discriminative subspaces S_{i2} and S_{j2} within a group. In our analysis we have 6 pairs of generative and 6 pairs of discriminative subspaces within each group (since a group has four domains), thereby making $12{*}21$ subspace pairs in all. While one could consider a pair made of a generative subspace S_{i1} and a discriminative subspace S_{j2}, we did not pursue that since the information contained in such a pair is different.

- Given a point on the Grassmann manifold S_1 and a tangent vector $B = \begin{pmatrix} 0 & A^T \\ -A & 0 \end{pmatrix}$.
- Compute the $d \times d$ orthogonal completion Q of S_1.
- Compute the compact SVD of the direction matrix $A = \tilde{V}_2 \Theta V_1$.
- Compute the diagonal matrices $\Gamma(t')$ and $\Sigma(t')$ such that $\gamma_i(t') = \cos(t'\theta_i)$ and $\sigma_i(t') = \sin(t'\theta_i)$, where θ's are the diagonal elements of Θ.
- Compute $\Psi(t') = Q \begin{pmatrix} V_1 \Gamma(t') \\ -\tilde{V}_2 \Sigma(t') \end{pmatrix}$, for various values of $t' \in [0, 1]$.

Algorithm 2: Algorithm for computing the exponential map, and sampling along the geodesic [8].

a set is to consider the subspace it spans[2]. We hence perform PCA on M_i' to obtain an orthonormal matrix M_i of size $N \times N_1$, with $N_1 < N$, whose column space signifies the new embedded data representation $f_1(x_i)$. $f_1(x_i)$ is a point on the Grassmannian $\mathbb{G}_{N,N_1}$. By repeating the above process for the entire training set $\mathcal{X}$, we obtain n points on $\mathbb{G}_{N,N_1}$ having location information y_i associated with them.

2.2. Performing Location Inference

We now train a classifier f_2 by performing statistics over the point cloud $f_1(x_i)$'s on $\mathbb{G}_{N,N_1}$, to recognize location y_t of the query x_t. Of the many possible techniques [4], we pursued the method of [10] since its utility for visual recognition has been demonstrated before. This method essentially performs kernel linear discriminant analysis on the points on $\mathbb{G}_{N,N_1}$ using the projection kernel $k_P(M_i, M_j) = \|M_i^T M_j\|_F^2 = trace[(M_i M_i^T)(M_j M_j^T)]$, which is a Mercer kernel that implicitly computes the inner product between $M_i's$ in the space obtained using the embedding; $\omega_P : \mathbb{G}_{N,N_1} \rightarrow \mathbb{R}^{N \times N}, span(M_i) \rightarrow M_i M_i^T$. To make the paper self-contained, we present the details of this method in Algorithm 3.

However since the number of locations m is generally much higher than the amount of data available at each location, we discriminate between the domains instead. But instead of using all 84 domains, we used only $c' = 64$ domains from third level since they provide the finest location grouping of images x_i among all the three levels of the hierarchy (Sec 2.1). We then learn the discriminative space f_2 by solving for (1) using M_i's and their associated domain labels, using which the reduced $(c' - 1)$ dimensional representation F_{train} for training data is obtained. The query location y_t is then inferred by first computing the matrix M_t from x_t using the procedure described in Sec 2.1.2, obtaining its reduced dimensional representation F_{test} from (1),

[2] We empirically evaluate some alternatives to model the information contained in matrix M_i in Sec 3.

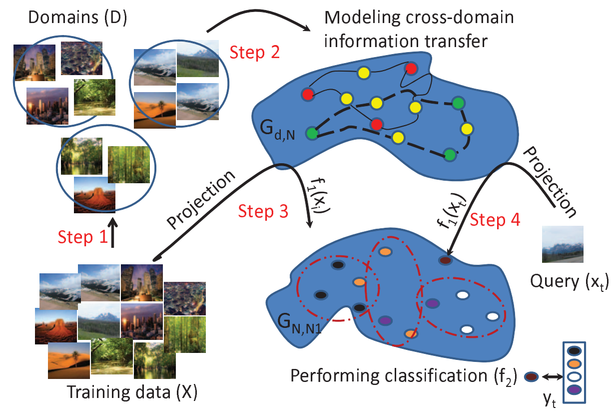


Figure 3. Overview of our approach. **Step 1:** Grouping $n = 11$ training data x_i from four unique locations y_i ($m = 4$; a *specific* mountain, wetland, city, desert) into three domains $\mathcal{D}$ ($c = 3$). These domains may contain visually dissimilar images as we do only a coarse grouping. Assume these three domains are combined into a single group $\mathcal{G}$. **Step 2:** Obtaining generative (red) and discriminative (green) subspaces from these domains, and sampling points (yellow) along the geodesic between them (solid and dashed lines, resp.) to learn cross-domain information transfer. **Step 3:** Projecting each training data x_i onto these subspaces to obtain an embedded representation $f_1(x_i)$ - colored ovals (based on y_i): black-city, orange-wetland, white-mountain, purple-desert. **Step 4:** Learning a discriminative space f_2 using Algo 3 (red ellipses) on $f_1(x_i)$ grouped by their domains ($c' = c$ here), to infer location y_t of $f_1(x_t)$ (brown oval) derived from query x_t.

and finally selecting the location y_i of the nearest neighbor from F_{train}. Figure 3 presents a visualization of the proposed approach.

2.2.1 Clustering

Besides location 'recognition', there could be cases where the data is not labeled. In such cases we can perform 'clustering' on the Grassmannian to determine the grouping of data, and one possibility is to perform k-means [28]. From the set of points $\mathcal{P} = (f_1(x_1), f_1(x_2), ..., f_1(x_n))$ on $\mathbb{G}_{N,N_1}$, we seek to estimate k clusters $\mathcal{C} = (C_1, C_2, ..., C_k)$ with cluster centers $(\mu_1, \mu_2, ..., \mu_k)$ so that the sum of geodesic-distance squares, $\sum_{i=1}^{k} \sum_{f_1(x_j) \in C_i} \bar{d}^2(f_1(x_j), \mu_i)$ is minimized. Here $\bar{d}^2(f_1(x_j), \mu_i) = |\exp_{\mu_i}^{-1}(f_1(x_j))|^2$, where $\exp_{\mu_i}^{-1}$ is the inverse exponential map computed from tangent plane centered at μ_i (Algorithm 1). As is the case with standard Euclidean k-means, we can solve this problem using an EM-based approach. We initialize the algorithm with a random selection of k points as the cluster centers. In the E-step, we assign each of the points of the dataset $\mathcal{P}$ to the nearest cluster center. Then in the M-step, we recompute the cluster centers using the Karcher mean algorithm described in the supplementary material.

From the training data x_i's grouped into c' domains, and query images x_{ti}'s, compute their respective embedded cross-domain data representation M_i's and M_{ti}'s (a collection of orthonormal matrices).

Training:

- Compute the matrix $[K_{train}]_{ij} = k_P(M_i, M_j)$ for all M_i, M_j in the training set, where k_P is the projection kernel defined earlier.
- Solve $\max_\gamma L(\gamma)$ by eigen-decomposition (1), with $K^\star = K_{train}$.
- Compute (c'-1)-dimensional coefficients, $F_{train} = \gamma^T K_{train}$

Testing:

- Compute the matrix $[K_{test}]_{ij} = k_P(M_i, M_{tj})$ for all M_i in training, and M_{tj} in testing.
- Compute (c'-1)-dimensional coefficients, $F_{test} = \gamma^T K_{test}$ by solving for (1) with $K^\star = K_{test}$.
- Perform one-nearest neighbor classification from the Euclidean distance between F_{train} and F_{test}, and **associate location y_t of a query in F_{test} to the location y_i of its nearest neighbor in F_{train}.**

The Rayleigh quotient $L(\gamma)$ is given by,

$$L(\gamma) = \max_\gamma \frac{\gamma^T K^\star(\bar{V} - 1_{B'} 1_{B'}^T / B')K^\star \gamma}{\gamma^T (K^\star (I_{B'} - \bar{V}) K^\star + \sigma^2 I_{B'})\gamma} \quad (1)$$

where $K^\star$ is the Gram matrix (K_{train} or K_{test}), $1_{B'}$ is a uniform vector $[1...1]^T$ of length B' corresponding to the number of gallery images, $\bar{V}$ is the block-diagonal matrix whose z^{th} block ($z = 1\ to\ c'$) is the uniform matrix $1_{B'_z} 1_{B'_z}^T / B'_z$, B'_z is the number of training images in z^{th} class, and $\sigma^2 I_{B'}$ is a regularizer to make computations stable ($\sigma = 0.3$ in our experiments).

Algorithm 3: Grassmann Kernel Discriminant Analysis [10].

3. Experiments

We evaluate the method on two datasets, im2GPS [11], and San Francisco [3], for location recognition and clustering and present an analysis of relative merits of some design choices involved in our approach.

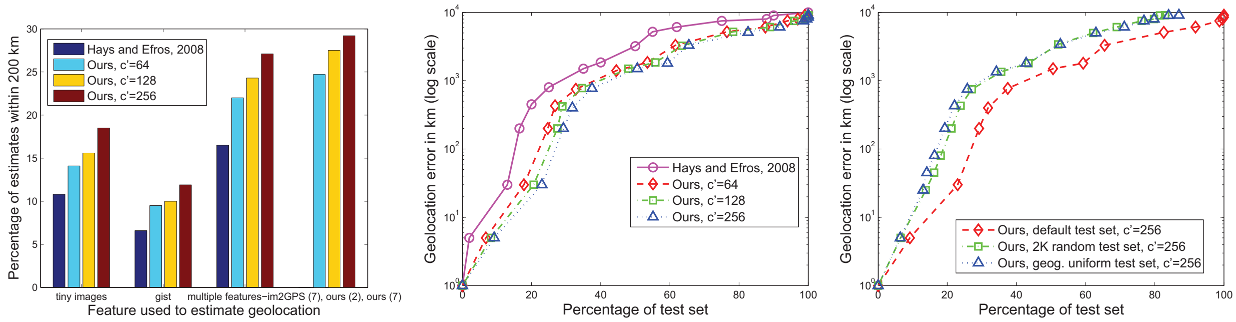
Value of parameters: We chose the values of N, N_1 and N' by performing 5-fold cross-validation on the training data (from each dataset) by varying subspace dimensions N and N_1 to reflect $85 - 95\%$ of PCA variance (in steps of 2%), and the number of samples N' along a geodesic ranging from 3 to 5 (in steps of 1).

3.1. im2GPS Dataset

We first experimented with the im2GPS dataset [11]. The training set contains images obtained from Flickr collections, while the test set contains 237 images representing different locations. We first used two of the seven features proposed in [11], tiny images and the gist descriptor with color, and then experimented with all seven features. In each case the selected features were concatenated into a long vector, which denotes our x_i. The reason behind this choice is to see how well our method performs with varying number of features, and for the trial with two features we chose tiny images and gist since they had lesser variance across different classification strategies studied in [11] and at the same time had reasonably good performance. We created domains $\mathcal{D}$ using the procedure outlined in Section 2.1, then modeled cross-domain information transfer using the geometry of subspaces derived from the domains (Sec 2.1.1), embedded those results into each training data x_i to obtain $f_1(x_i)$ (Sec 2.1.2) and learned the classifier space f_2 (Sec 2.2) using $c' = 64$ domains with which the query location was inferred. With the training done offline, it takes about 10 seconds on a 2 GHz machine to process a query. Some visualizations of nearest neighbors corresponding to query images is given in Figure 4, and the performance curves are reported in Figure 5. We then repeated the above process but with the classifier f_2 trained on even finer domains, by first splitting each of the 64 domains vertically into two ($c' = 128$) and then horizontally into two ($c' = 256$), to study the sensitivity of the classifier to the number of domains. Please note that this impacts only the classification stage (Sec 2.2) and not any of the earlier stages.

Observations: It can be seen that our method performs better overall, even by using only two features (out of the original seven), which shows the utility of the joint generative and discriminative information captured by our model. Using all seven features results in an improved performance. Another observation is that the recognition improves with finer grouping of domains, which is intuitive since such domains are more representative of finer locations. In Figure 5(c) we report the location recognition performance on two other test sets, 2K random and geographically uniform, that are provided as a part of the im2GPS dataset. These two test sets are relatively more challenging than the earlier (default) test set because, (i) the random 2K test set contains several instances that are not common landmarks, and (ii) the images in the geographically uniform set may not contain equally dense neighborhood around them that are distinct.

Utility of hierarchical formation of domains, and creating groups from them: We now study two alternate strategies to create and analyze domains $\mathcal{D}$ as opposed to the scheme discussed in Sec 2.1. In the first setting we do not pursue a hierarchical scheme and use just the domains from the third level along with their grouping. So we have 64 domains $\{\mathcal{D}_i\}_{i=21}^{84}$ that are consolidated into 16 groups $\{\mathcal{G}_i\}_{i=5}^{21}$ (from Sec 2.1). We then create generative and discriminative subspaces to analyze geodesics between as described earlier. We have 12*16 subspace pairs in this case, and let us call this setup Case-A1. We then consider another setup, Case-A2, where we remove the



(a) (b) (c)

Figure 5. Results on the im2GPS data. (a) Analyzing the performance of features using our method and that of [11]. It can be seen that our method learns more information by using only 2 features in comparison with 7 used by [11]. Finer concentration of domains (using larger c' for classification) improves performance. (b) Similar trends are observed in the location retrieval of query images on the default test set. Graphs using all seven features are shown here.(c) Results on 2K and geographically uniform test sets are inferior to that on the default test set (reproduced from (b)). Results with 64 and 128 domains on these two test sets are given in the supplementary material.

Figure 4. Sample retrieval results from our approach on the im2GPS dataset [11]. Each row shows five nearest matches for a query image in the first column. It can be seen that the famous locations (top two rows) have retrievals that are both visually and geographically similar, while the retrievals for rows 4 and 5 are visually similar but geographically varying. Row 3 presents a case where both the visual and geographical similarities are divergent.

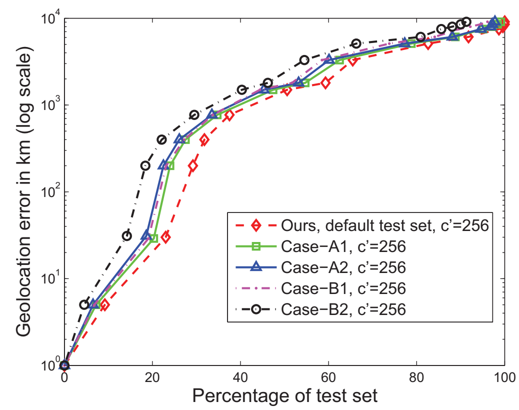


Figure 6. Analyzing relative merits of some design choices involved in our approach. Case-A1 and A2 deal with the domain and group creation aspect, whereas Case-B1 and B2 deal with obtaining the embedded representation $f_1(x_i)$ using Euclidean tools instead of geometry-driven ones. The curve corresponding to the legend 'Ours-default test set' is reproduced from Figure 5.

group information from Case-A1 and consider generative and discriminative subspace pairs among all 64 domains. Discriminative subspaces in the case are obtained by a two-class PLS in a one-vs-remaining(63 domains) setting. We have 4032 subspace pairs in this case (${}^{64}C_2$ each for generative pairs and discriminative pairs). We present the results for these two cases, with $c' = 256$, in Figure 6. It can be see that Case-A1 is better than Case-A2 while both cases are inferior to the hierarchical domain formation scheme (Sec 2.1). This suggests that a top-down mechanism of obtaining domains is better, and for analyzing subspaces across domains it is important to have some supervision (in terms of groups $\mathcal{G}_i$) in modeling visual properties across locations. Results with $c' = 64$ and 128, which follow similar trends as that of $c' = 256$, are given in the supplementary material.

Utility of considering column space of M_i to perform location recognition: We then address the utility of obtaining the embedded cross-domain representation $f_1(x_i)$ by considering the column span of matrix M_i. We consider two alternate possibilities. Case-B1: Performing PLS based dimensionality reduction, which has shown to be effective for recognition tasks [24], by concatenating the columns of M_i into a long vector and learning a discriminative space using the domain labels c' of M_i. We then project matrices from training M_i and query M_{ti} onto this space and perform 1-nearest neighbor classification using the PLS projection co-efficients. We also consider Case-B2 where we replicate the steps of Case-B1 using matrices M'_i instead of M_i (i.e. not performing PCA to model the projections of x_i on the c_1 subspaces). We report the comparison in Figure 6 where Case-B1 is better than Case-B2 that suggests that doing a PCA is a good way to encompass information contained in the matrix M_i', and Algorithm 3 is better than Case-B1 which suggests that utilizing the *geometry* spanned by the column space of matrices M_i has advantages over an Euclidean treatment.

3.1.1 Clustering

We then performed a clustering experiment to account for cases where the data x_i may not have location information y_i. We used the im2GPS training set $\mathcal{X}$ (without y_i) for this purpose. We first created 64 random groupings of the data into domains $\mathcal{D}$. We learnt generative and discriminative subspaces from these domains along the lines of Case-A2 as we do not have location information to form groups $\mathcal{G}$ (Note that while in Case-A2 the domains were created using location information but the subsequent groups were not formed *deliberately*, here in clustering we do not actually have location information to construct the domains, and therefore the groups). We then modeled cross-domain information by projecting each data $x_i \in \mathcal{X}$ onto the geodesic between these subspaces to obtain $f_1(x_i)$, and performed k-means clustering (Sec 2.2.1) by setting $k = 64$ equaling the number of domains. We computed the geolocation error for each x_i by picking out four closest neighbors of $f_1(x_i)$ from its cluster (using $\bar{d}^2$), computing the error between the 'ground truth' location y_i with the ground truth location of its four neighbors, and taking the average of those four values. The experiment was repeated with 128 and 256 clusters (without changing the number of domains), and the performance curves are reported in Figure 7(b) along with sample clustering results in Figure 7(a). While the clustering accuracy is not very high, we are still able to infer approximate locations without any labeled data, for a problem where visually similar images can come from vastly different locations.

3.2. San Francisco Dataset

We next experimented with the San Francisco dataset [3] that was generated by aligning panoramic images to a 3D model of the city. There are two sets of images, perspective central images (PCI) and perspective frontal images (PFI), which were subjected to histogram equalization before extracting upright SIFT feature keypoints. We then obtained a bag-of-words histogram codebook of length 800 representing x_i, for each of these two images sets separately, by performing (standard Euclidean) k-means/vector quantization on the SIFT features. We then created domains $\mathcal{D}$ and the corresponding groups $\mathcal{G}$ by partitioning the rectangular grid covering the city. All other parameters we retained from the im2GPS dataset in order to study the experimental results in a level field.

We then learnt f_1 and f_2 from the procedure described before to infer the locations of the test set containing 803 query images. The results are given in Figure 8. When the GPS option is used, we infer query location by computing nearest neighbors (Algorithm 3) from the training data pertaining to the domain of the query (obtained from its ground truth) and to the four domains adjacent to it. It can be seen that the use of GPS information does improve recognition, and the non-GPS results in general are better when compared to the im2GPS dataset, specifically in the very low error tolerance region. One reason for this, besides the obvious difference in the data, could be the finer spatial concentration of data $\mathcal{X}$ (a city vs. entire world). The results for cases A1, A2, B1 and B2 largely follow the pattern observed in the other dataset, and we present those results in the supplementary material.

4. Conclusion

We proposed a top-down approach to jointly model generative and discriminative information portrayed by the data and demonstrated its utility for the challenging problem of location recognition and clustering, where the visual and location properties of images may not always correlate. The competitive results obtained on two public datasets, along with an empirical analysis on the utility of certain design choices, seems to suggest the importance of modeling tools that are cognizant of the underlying geometric space of the data they operate on.

References

[1] G. Baatz, K. Köser, D. Chen, R. Grzeszczuk, and M. Pollefeys. Leveraging 3d city models for rotation invariant place-of-interest recognition. *IJCV*, 96(3):1–20, 2011. 1

[2] P. Belhumeur, J. Hespanha, and D. Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE TPAMI*, 19(7):711–720, 1997. 2, 3

[3] D. Chen, G. Baatz, K. Koser, 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. In *CVPR*, pages 737–744, 2011. 1, 5, 7, 8

[4] Y. Chikuse. *Statistics on special manifolds*, volume 174 of *Lecture Notes in Statistics*. Springer Verlag, 2003. 3, 4

[5] C. Doersch, S. Singh, A. Gupta, J. Sivic, and A. A. Efros. What makes paris look like paris? *ACM Transactions on Graphics (SIGGRAPH)*, 31(4), 2012. 1

[6] A. Edelman, T. Arias, and S. Smith. The geometry of algorithms with orthogonality constraints. *SIAM Journal of Matrix Analysis and Application*, 20:303–353, April 1999. 2, 3

[7] F. Fraundorfer, C. Wu, J. Frahm, and M. Pollefeys. Visual word based location recognition in 3d models using distance augmented weighting. In *3DPVT*, 2008. 1

[8] K. Gallivan, A. Srivastava, X. Liu, and P. Van Dooren. Efficient algorithms for inferences on grassmann manifolds. In *Workshop on Statistical Signal Processing*, pages 315–318, Feb 2003. 3, 4

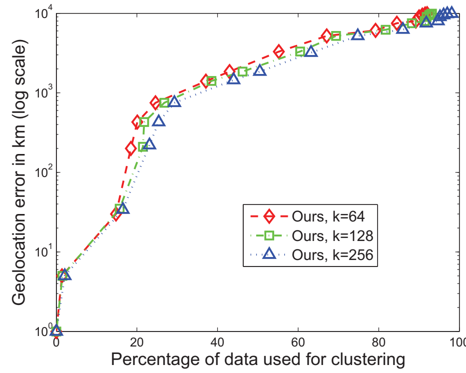
[9] R. Gopalan, R. Li, and R. Chellappa. Domain adaptation for object recognition: An unsupervised approach. In *ICCV*, pages 999–1006. IEEE, 2011. 2

[10] J. Hamm and D. Lee. Grassmann discriminant analysis: a unifying view on subspace-based learning. In *ICML*, 2008. 4, 5

[11] J. Hays and A. Efros. Im2gps: estimating geographic information from a single image. In *CVPR*, pages 1–8, 2008. 1, 5, 6

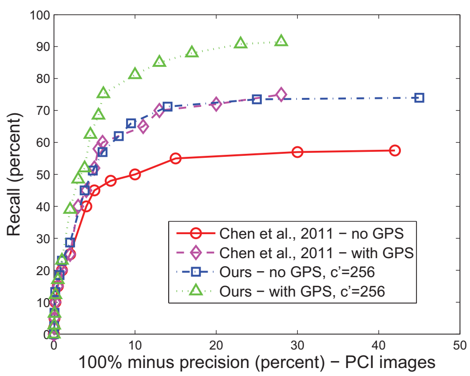
[12] A. Irschara, C. Zach, J. Frahm, and H. Bischof. From structure-from-motion point clouds to fast location recognition. In *CVPR*, 2599–2606 2009. 1

(a)

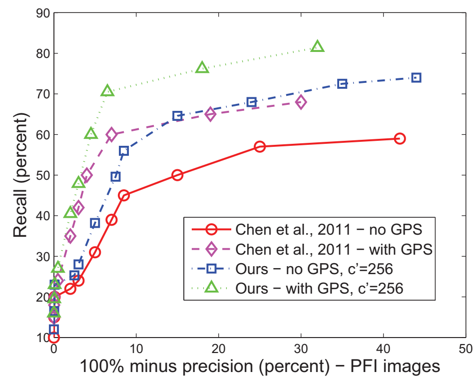


(b)

Figure 7. Clustering on im2GPS data. (a) Sample clustering results. In each row, the first column picks an image and displays its four nearest clustered neighbors. (b) For every image, we compute the difference of its 'ground truth' location with the ground truth location of its four nearest neighbors, and consider the average of these location errors. The k-means clustering and the corresponding random grouping of data into domains $\mathcal{D}$ was repeated 10 times and the average location errors are plotted. While the performance slightly improves with larger clusters, it is not as significant as in the recognition setting, which reiterates the advantage of having labels (or supervision).



(a)



(b)

Figure 8. Precision-recall curves on the San Francisco data [3] with PCI (a) and PFI (b) images. It can be seen that the GPS information offers a good performance improvement. Results using 64 and 128 domains are given in the supplementary material.

[13] R. Ji, L. Duan, J. Chen, H. Yao, J. Yuan, Y. Rui, and W. Gao. Location discriminative vocabulary coding for mobile landmark search. *IJCV*, 96(3):1–25, 2011. 1

[14] J. Knopp, J. Sivic, and T. Pajdla. Avoiding confusing features in place recognition. In *ECCV*, pages 748–761, 2010. 1

[15] A. Kumar, J. Tardif, R. Anati, and K. Daniilidis. Experiments on visual loop closing using vocabulary trees. In *CVPR*, pages 1–8, 2008. 1

[16] X. Li, C. Wu, C. Zach, S. Lazebnik, and J. Frahm. Modeling and recognition of landmark image collections using iconic scene graphs. In *ECCV*, pages 427–440, 2008. 1

[17] Y. Li, D. Crandall, and D. Huttenlocher. Landmark classification in large-scale image collections. In *CVPR*, pages 1957–1964, 2009. 1

[18] Y. Li, N. Snavely, and D. Huttenlocher. Location recognition using prioritized feature matching. *ECCV*, pages 791–804, 2010. 1

[19] D. Lin, A. Kapoor, G. Hua, and S. Baker. Joint people, event, and location recognition in personal photo collections using cross-domain context. In *ECCV*, pages 243–256, 2010. 1

[20] H. Nasr and B. Bhanu. Landmark recognition for autonomous mobile robots. In *ICRA*, pages 1218–1223, 1988. 1

[21] K. Ni, A. Kannan, A. Criminisi, and J. Winn. Epitomic location recognition. *IEEE TPAMI*, 31(12):2158–2167, 2009. 1

[22] G. Schindler, M. Brown, and R. Szeliski. City-scale location recognition. In *CVPR*, pages 1–7, 2007. 1

[23] G. Schroth, R. Huitl, D. Chen, M. Abu-Alqumsan, A. Al-Nuaimi, and E. Steinbach. Mobile visual location recognition. *IEEE Signal Processing Magazine*, 28:77–89, April 2011. 1

[24] W. Schwartz, H. Guo, and L. Davis. A robust and scalable approach to face identification. *ECCV 2010*, pages 476–489, 2010. 6

[25] Y. Takeuchi and M. Hebert. Evaluation of image-based landmark recognition techniques. In *Robotics Institute CMU-RI-TR-98-20*, 1998. 1

[26] A. Torralba, K. Murphy, W. Freeman, and M. Rubin. Context-based vision system for place and object recognition. In *CVPR*, 2003. 1

[27] S. Tsai, H. Chen, D. Chen, R. Vedantham, R. Grzeszczuk, and B. Girod. Mobile visual search using image and text features. In *Asilomar*, pages 845–849, 2011. 1

[28] P. Turaga, A. Veeraraghavan, A. Srivastava, and R. Chellappa. Statistical computations on special manifolds for image and video-based recognition. *IEEE TPAMI*, 33(11):2273–2286, 2011. 4

[29] M. Turk and A. Pentland. Face recognition using eigenfaces. In *CVPR*, pages 586–591, 1991. 2, 3

[30] H. Wold. *Partial Least Squares*, volume 6. Encyclopedia of Statistical Sciences, 1985. 3

[31] K. Yap, T. Chen, Z. Li, and K. Wu. A comparative study of mobile-based landmark recognition techniques. *IEEE Intelligent Systems*, 25(1):48–57, 2010. 1

[32] Y. Zheng, M. Zhao, Y. Song, H. Adam, U. Buddemeier, A. Bissacco, F. Brucher, T. Chua, and H. Neven. Tour the world: building a web-scale landmark recognition engine. In *CVPR*, 2009. 1