Hierarchical Sparse Coding With Geometric Prior For Visual Geo-location

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

Abstract

We address the problem of estimating location information of an image using principles from automated representation learning. We pursue a hierarchical sparse coding approach that learns features useful in discriminating images across locations, by initializing it with a geometric prior corresponding to transformations between image appearance space and their corresponding location grouping space using the notion of parallel transport on manifolds. We then extend this approach to account for the availability of heterogeneous data modalities such as geo-tags and videos pertaining to different locations, and also study a relatively under-addressed problem of transferring knowledge available from certain locations to infer the grouping of data from novel locations. We evaluate our approach on several standard datasets such as im2gps, San Francisco and MediaEval2010, and obtain state-of-the-art results.

1. Introduction

Inferring which location co-ordinates an image corresponds to is a challenging problem in visual recognition, which has applications in areas such as surveillance, geo-mapping, landform modeling among others. It differs from other recognition problems pertaining to, say objects, scene, and faces, in that it is not enough to model visual appearance alone since two images that have similar appearance can correspond to different locations, and two images having dissimilar appearance can come from neighboring locations. While initial efforts investigating this problem began atleast two decades ago [24, 25], substantial progress was seen only in the last decade due to the availability of large quantity of data and the related progress in data-driven techniques [29, 14].

We now present a broad overview of methodologies that have been proposed in the literature for this problem. One of the prominent efforts is the im2gps method [13] that brought the location recognition problem to the mainstream vision community, by exploring different visual features coupled with big data to show that it is indeed plausible to estimate *where* an image was captured. There have been many other efforts focusing on learning efficient feature representations and location matching strategies using diverse set of techniques. For instance, [26] pursued a generative model based on epitomic image analysis to capture appearance and geometric structure of an environment while allowing for variations due to motion, occlusion and non-Lambertian effects. [20] combined 2D appearance constraints with 3D geometric relationships using iconic scene graphs. Faster retrieval methods were pursued by [28] using vocabulary trees and by [21] using prioritized feature matching. Ideas pertaining to structure from motion were used by [15] to generate synthetic views from 3D point clouds to assist image-based geo-location. Location search using hybrid image-keyword technique was explored by [35], while the contextual information conveyed by people and events on locations for personal photo collections was analyzed by [22].

More recently, several discriminative learning strategies that could handle larger data scale were proposed. [6] represented the structure of the training location database as a graph and proposed a method for selecting a set of subgraphs and learning a local discriminative distance function for each of them. [12] trained per-location classifiers using support vector machines (SVM) for each location, by handling the mismatch in the number of positive and negative class samples using non-parametric statistics. [4] pursued random forest-based codebook learning on the feature descriptors obtained using structure from motion computations on image collections. Manifold modeling of generative and discriminative appearance information was studied by [11], by transferring such information across locations using geodesic paths. Purely geometric correspondence-free geo-localization was performed by [2] using 3D point-ray features extracted from the digital elevation maps of urban environments, and [19] worked with human annotated line segments of ground view images to perform location matching of planar structures under projective uncertainty. There have also been studies based on repetitive

image structures [31], probabilistic crowd sourcing with visual odometry and free online road maps [5], and using overhead imagery and landcover survey data to geo-locate ground-level photographs [23]. Other efforts include ideas from random walks for GPS-tag refinement [37], and pairwise probability voting without RANSAC for fast location inference [17].

Contributions: Our approach addresses the aspect of learning good representations for location recognition, and we pursue this objective in both geometric and statistical flavors. (i) Firstly we understand how the appearance information of training images correlate with their location information. To this end we cluster the images based on their appearance features to form the *appearance space* and cluster the images based on their location information to obtain the *location space*. We then bridge the two spaces by considering the subspace spanned by each cluster and then employing the notion of parallel transport on Grassmann manifold to produce a collection of possible transformations that map information contained in subspaces from one space on to another. (ii) We use this geometric information as a prior for initializing a hierarchical sparse coding scheme, which automatically learns representations that compactly represent each image from the large pool of information conveyed by the geometric prior. We then train discriminative classifiers on the sparse codes to infer location information of the test image. An overview of our approach is given in Figure 1.

We believe an explicit modeling of transformations across location and appearance information is essential to address the challenging problem of location recognition. While many existing approaches consider this problem as just another classification problem, we empirically demonstrate that our geometric prior is beneficial to such techniques as well. Since such a prior increases the amount data to be processed and the dependency between them by several folds, our feature learning solution using hierarchical sparse coding brings in efficiency with regards to these challenges. Our objective in learning sparse codes also lends itself to integrate heterogeneous data modalities such as geotags and videos, in addition to images, as well as in transferring knowledge gained on recognizing certain locations to perform inference on never-seen-before locations. We present details of our approach in Section 2, followed by evaluations on three public datasets in Section 3. Section 4 concludes the paper.

2. Proposed Approach

Let $z = \{(x_i, y_i)\}_{i=1}^N$ denote the training data, where $x_i \in \mathbb{R}^n$ is the n-dimensional feature vector pertaining to the i^{th} training image and y_i is its location co-ordinates. Given a test image feature vector $\bar{x}$, the goal of this work is to estimate its location $\bar{y}$. We proceed by, (i) learning the geometric prior $\mathcal{T}$ from the training data (Sec 2.1), (ii) using it as an initialization in learning hierarchical sparse codes (Sec 2.2), and (iii) constructing an SVM classifier on the sparse codes to obtain a similarity matrix W (Sec 2.2.1). We then compute the sparse code impacted by the prior for the test data $\bar{x}$, obtain its similarity vector from SVM, and compute its nearest neighbor in W and assign $\bar{y}$ to the location co-ordinates of that neighbor.

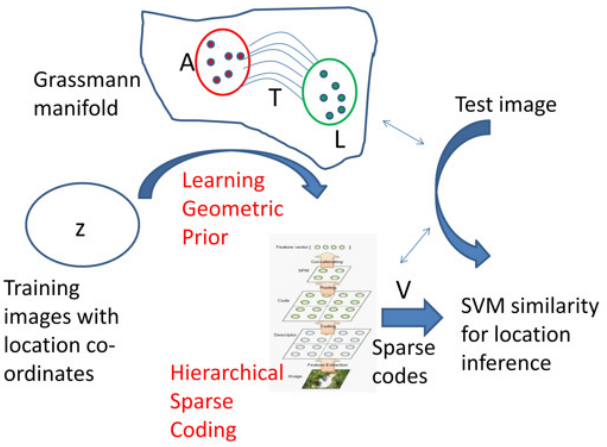


Figure 1. Learning representations for location recognition by explicitly characterizing relationship between image appearance and their location grouping. Given training data z with location labels, we cluster the data to form appearance space $\mathcal{A}$ and location space $\mathcal{L}$, and derive geometric transformations between them using parallel transport $\mathcal{T}$ on the Grassmann manifold. Shown on the manifold are points corresponding to subspaces obtained from each cluster. We then learn hierarchical sparse codes by initializing it with the geometric prior and perform location estimation of test data using SVM similarities learnt on sparse codes of the training data.

2.1. Geometric Prior

We first understand how the appearance information and location information contained in z correlate. We group z according to their appearance features $\{x_i\}_{i=1}^N$ into k clusters using the normalized cuts algorithm [30], to form the appearance space $\mathcal{A}$. Similarly we group z using their location $\{y_i\}_{i=1}^N$ into k clusters, by dividing the locations uniformly in rectangular patches under a planar earth surface assumption, to obtain the location space $\mathcal{L}$. We then obtain subspaces for each cluster, by performing principal component analysis on the features x_i of images contained in those clusters. We now have k subspaces $\mathcal{S}^A = \{\mathcal{S}_j^A\}_{j=1}^k$ for $\mathcal{A}$ and $\mathcal{S}^L = \{\mathcal{S}_j^L\}_{j=1}^k$ for $\mathcal{L}$, respectively. Let d be the dimension of each subspace, with $d < n$. Since the subspaces are points on the Grassmann manifold $\mathcal{G}_{n,d}$ [1], which is the space of all d-dimensional subspaces in $\mathbb{R}^n$, we now perform computations on this manifold to obtain the geometric prior.

Since there could be multiple ways in which $\mathcal{A}$ and $\mathcal{L}$ can be related, we utilize the notion of parallel transport to obtain a collection of transformations $\mathcal{T}$. We now briefly discuss the concept of parallel transport, and refer the reader to [1] for more details. Parallel transport $\mathcal{T}_S$ of a point $\mathcal{S}$ on $\mathcal{G}_{n,d}$ consists of moving the *tangent space* representation of $\mathcal{S}$, Δ_S, along the *geodesic* $\gamma(t)$ in the direction such that it parallel to itself with respect to the geodesic. The tangent space is a locally Euclidean representation around a point on the manifold, which is obtained using inverse exponential maps, and the geodesic is the shortest path between a pair of points on the manifold. With this qualitative background, we now present our method for obtaining the geometric prior $\mathcal{T}$ in Algorithm 1.

Algorithm 1: Computing the geometric prior between appearance space and location space

(1) Compute the mean of subspaces in $\mathcal{A}$ and $\mathcal{L}$, denoted by μ_A and μ_L respectively, using the Karcher mean algorithm [18].
(2) Define tangent plane at μ_A and obtain the geodesic $\gamma(t)$ between the tangent plane representations of the means, Δ_{μ_A} and Δ_{μ_L} respectively.
(3) Obtain parallel transport $\mathcal{T}_{\mu_A}$ of μ_A by moving Δ_{μ_A} in a direction such that it is parallel with respect to the geodesic $\gamma(t)$.
(4) Using $\mathcal{T}_{\mu_A}$ as the reference, parallel transport all subspaces $\mathcal{S}^A$ in $\mathcal{A}$ to obtain their corresponding $\{\mathcal{T}_{S_j^A}\}_{j=1}^k$.

Let $\mathcal{T}$, the union of $\mathcal{T}_{\mu_A}$ and $\{\mathcal{T}_{S_j^A}\}_{j=1}^k$, denote our geometric prior. Once we have this set of transformations, we map the information contained in them onto each training data in z by the following process. We sample points on $\mathcal{T}$ using exponential maps to get a new collection of m subspaces $\mathcal{S}^* = \{S_j^*\}_{j=1}^m$. We then project each x_i onto these subspaces to obtain a set of m d-dimensional vectors, $X_i^* = [x_{i,1}^*, ..., x_{i,m}^*] \in \mathbb{R}^{d \times m}$. The collection $\{X_i^*\}_{i=1}^N$ thus contains information on how the appearance and location properties of images contained in z correlate, and we arrange them in columns to form a $d \times N'$ matrix P, where $N' = m^* N$.

2.2. Hierarchical Sparse Coding

We now perform statistics on $Z^* = \{(X_i^*, y_i)\}_{i=1}^N$ to discriminate between locations. Automated feature learning techniques [3] are widely popular in recent times due to their ability to handle large quantities of data. Among many such methodologies, the notion of sparse representation is particularly useful for our problem since the geometric prior $\mathcal{T}$ increases the amount of data Z^* to be processed by several times, and in doing so the dependency between the data has also increased since they have been affected by common set of transformations corresponding to the prior. We thus apply techniques from hierarchical sparse coding [16, 36] to learn codes that can be used to discriminate between locations.

Given a $d \times N'$ matrix P of training samples, whose columns correspond to feature vectors contained in X_i^*'s, the task of learning a dictionary $D = [d_1, .., d_K] \in \mathbb{R}^{d \times K}$ together with sparse codes $Q = [q_1, .., q_{N'}] \in \mathbb{R}^{K \times N'}$ is typically given by the following optimization problem,

$$\begin{aligned} D^\star, Q^\star = \arg\min_{D,Q} ||P - DQ||_F^2 \\ s.t\ ||q_i||_0 \leq \lambda, \forall i = 1, .., N' \\ ||d_j||_2 = 1, \forall j = 1, .., K \end{aligned} \tag{1}$$

where $||.||_F$ is the Frobenius norm, λ is a positive constant, and the constraint $||q_i||_0 \leq \lambda$ promotes sparsity in the coefficient vectors. The second constraint $||d_j||_2 = 1$ keeps the columns of the dictionary D from becoming arbitrarily large that may result in very small sparse codes. Among different solvers from the literature, we used the widely popular K-SVD technique [9] to solve (1), which operates by alternatively computing D and Q. Having learnt the dictionary D, given a feature vector x^* from the column of P, we obtain its sparse code q by minimizing the following objective,

$$||x^* - Dq||_2^2 \ \ s.t.\ \ ||x^*||_0 \leq \lambda \tag{2}$$

q can be obtained by applying orthogonal matching pursuit [27], which is a greedy algorithm that iteratively selects an element of the sparse code to be made non-zero to minimize the residual reconstruction error. This explains the basic sparse coding technique, and we extend it in a hierarchical fashion using the method of [34] by using two layers with max pooling and keeping all other design choices unchanged. While the input to the first layer is the features obtained from the geometric prior, the input to the next stage is the codes output by the first layer obtained from (2).

2.2.1 Location recognition

Let the features obtained from the above hierarchical sparse coding approach for all training data in P be represented by the matrix $V = [v_{1,1}, .., v_{1,m}, v_{2,1}, ..., v_{N,m}] \in \mathbb{R}^{K \times N'}$. Each column inherits corresponding location labels from Z^*. We now train a multi-class SVM classifier on this data [8]. Since the number of location labels can be large, for the sake of simplicity, we group the location co-ordinates into k classes in the manner described for creating the location space $\mathcal{L}$. We then assign the class with highest SVM probability for each $v_{i,j}$, and concatenate the class labels for $\{v_{i,j}\}_{j=1}^m, \forall i = 1, .., N$ to obtain the final similarity matrix $W = [w_1, ..., w_N] \in \mathbb{R}^{m \times N}$. The reasoning behind this is

that the images belonging to a particular location will have closer similarity across location groupings when subjected to the transformations produced by the geometric prior. Then given a test image, we extract its feature $\bar{x}$, subject it to the geometric prior transformations $\mathcal{T}$ to obtain $\bar{X}^*$, using which we obtain its sparse code $[\bar{v}_1, \bar{v}_2, .., \bar{v}_m] \in \mathbb{R}^{K \times m}$ and the m-dimensional SVM similarity vector $\bar{w}$. We then compute 1-nearest neighbor (1-nn) between $\bar{w}$ and W and assign the its location co-ordinates $\bar{y}$ to that of its nearest neighbor.

2.3. Heterogeneous Geo-location

We now address situations where heterogeneous data modalities such as, image, video and geo-tags, are available for data from all locations. We extract relevant features from each of the M modalities, and obtain their sparse code matrix $\{V^i\}_{i=1}^M$ using the process described before. Since the V^i's can have different dimensions, we map them on to a common dimensional latent space using the heterogeneous manifold alignment approach of [32]. This approach considers each modality as a manifold, and constructs mapping functions $(f_1, f_2, .., f_M)$ by minimizing the following cost function

$$\mathcal{F}(f_1, f_2, .., f_M) = (A + C)/B \tag{3}$$

$$A = 0.5 \sum_{i=1}^{M} \sum_{j=1}^{M} \sum_{a=1}^{N'} \sum_{b=1}^{N'} ||f_i^T v_a^i - f_j^T v_b^j||^2 W_s^{i,j}(a,b)$$

$$B = 0.5 \sum_{i=1}^{M} \sum_{j=1}^{M} \sum_{a=1}^{N'} \sum_{b=1}^{N'} ||f_i^T v_a^i - f_j^T v_b^j||^2 W_d^{i,j}(a,b)$$

$$C = 0.5 \sum_{i=1}^{M} \sum_{a=1}^{N'} \sum_{b=1}^{N'} ||f_i^T v_a^i - f_i^T v_b^i||^2 W_c^{i}(a,b)$$

where the indices i and j denote the data modalities, and v_a^i denote the sparse code corresponding to a^{th} column of the matrix V^i. The cost function A encourages data from same class across modalities to be projected closer to each other in the latent space, where $W_s^{i,j}(a,b) = 1$ if v_a^i and v_b^j are from same class, and $W_s^{i,j}(a,b) = 0$ otherwise. The cost function B encourages data from different classes across modalities to be separated in the latent space, with $W_d^{i,j}(a,b) = 1$ if v_a^i and v_b^j are from different classes, and $W_d^{i,j}(a,b) = 0$ otherwise. The cost function C preserves topology of the data within each modality, where $W_c^i(a,b)$ denotes similarity between two data samples, v_a^i and v_b^i, within a same i^{th} modality and we define $W_c^i(a,b) = e^{-||v_a^i - v_b^i||^2}$. Hence the cost function in (3) keeps similar data samples closer by minimizing A and C, and keeps dissimilar samples farther by maximizing B, and to produce the mapping functions $\{f_i\}_{i=1}^M$.

We thus obtain an integrated sparse code U across different modalities mapped onto the latent space given by, $U = [f_1(V^1)\ f_2(V^2)\ ..\ f_M(V^M)]$ using which we train the SVM classifier as before to obtain similarity vectors and perform location inference on multi-modal test data.

3. Experiments

We tested our method on three public datasets namely, im2gps [13] containing earth scale images, San Francisco [7] containing images from a smaller scale corresponding to the city of San Francisco, and MediaEval2010 [10] containing Flickr videos of varied locations along with textual meta data. These set of experiments serve as a good test bed to evaluate our algorithm on different types of features as well as on varying range of locations.

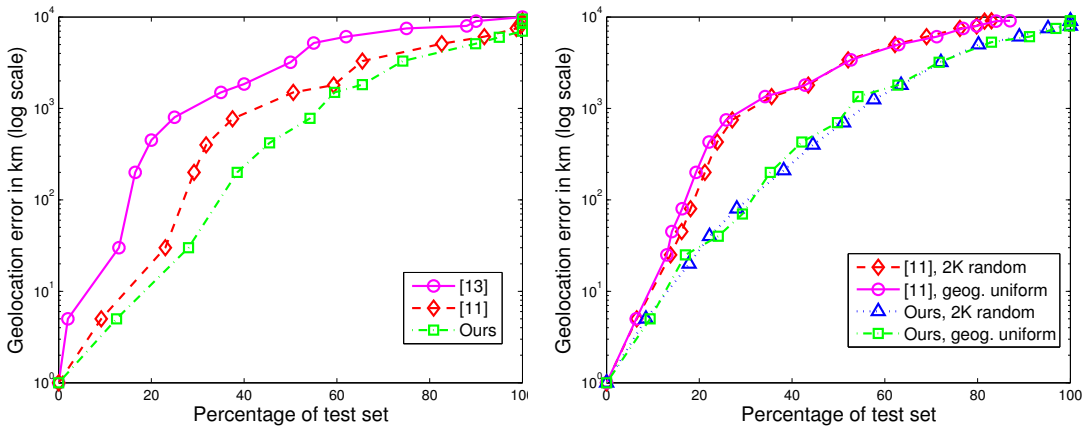
Parameter settings and design choices: For all the experiments, we used the following parameters. The number of clusters k for obtaining the appearance space $\mathcal{A}$ and location space $\mathcal{L}$, as well as the number of classes in training the SVM, was set as 64. Subspace dimension d was set as 100, as it contained upto 99% of energy of the data we are modeling. We sampled five uniformly spaced points on each transformation obtained from parallel transport contained in our geometric prior $\mathcal{T}$. For multi-class SVM, linear kernel was used in a one-class vs. remaining classes configuration. Parameters related to hierarchical sparse coding and heterogeneous latent space mapping were retained as those prescribed in the original techniques. Also note that our choices of algorithms such as normalized cuts, principal component analysis, and SVM can be substituted with other alternatives from the literature.

3.1. im2gps dataset

This dataset [13] contains over six million geo-tagged photos of the world collected from Flickr. While this is a tremendous amount of data, it can not be considered an exhaustive visual sampling of earth, since it averages only about 0.0435 pictures per square kilometer of the earth's land area. From each image the following features were extracted as per the protocol namely, tiny image, color histograms, texton histograms, line features, gist descriptor with color, and geometric context. We then concatenate these features into a single vector to obtain our training collection z. The standard test set contains 237 images of which 5% were recognizable as popular tourist sites, while the remaining are not. The results are given in Figure 2(a). We also experimented with the 2K random test set, that contains much less tourist sites, and geographically uniform test set that avoids denser representation of certain locations. We present our location recognition results for these two test sets in Figure 2(b).

3.2. San Francisco Dataset

We next experimented with the San Francisco Landmark Dataset [7], which contains a database of 1.7 million train-



(a) (b)

Figure 2. Geo-location accuracy on the im2gps dataset using (a) the standard test set and (b) the 2K random, geographically uniform test sets. *Best viewed in color.*

ing images of buildings in San Francisco with ground truth labels, and calibration data, as well as a difficult query set of 803 cell phone images taken with a variety of different camera phones. The training data was originally acquired by vehicle-mounted cameras with wide-angle lenses capturing spherical panoramic images. For all visible buildings in each panorama, a set of overlapping perspective images is generated. z was obtained from training images using a bag-of-words histogram codebook of length 800 representing by performing Euclidean k-means/vector quantization on the SIFT features. We present the results in Figure 3 and 4, which reports recall as a function of the number of correct matches within the top few closest neighbors, and the full precision-recall curves respectively. Results on this dataset by using the GPS information of images in selecting the neighbors resulted in an average improvement of 12% over the numbers reported in the above figures. With the GPS option used, the similarity vectors corresponding to training data only from the five closest location clusters (out of k) to that of the ground truth test data location are used.

3.3. MediaEval dataset

We then followed the protocol of [10] in experimenting with the MediaEval2010 dataset. This heterogeneous dataset contains 5091 Flickr videos accompanied by geo-tags for various locations. For the text modality, we extracted the co-occurrence feature descriptor of words using probabilistic latent semantic analysis to form z. Whereas for the visual modality, we extracted the following features from each video frame namely, color and edge directivity descriptor, scalable color, edge histogram, fuzzy color and texture histogram, and color layout, and concatenated them for all frames to form z. We then pursued the heterogeneous geo-location approach detailed in Section 2.3 to infer the location of heterogeneous test data comprising of 5125

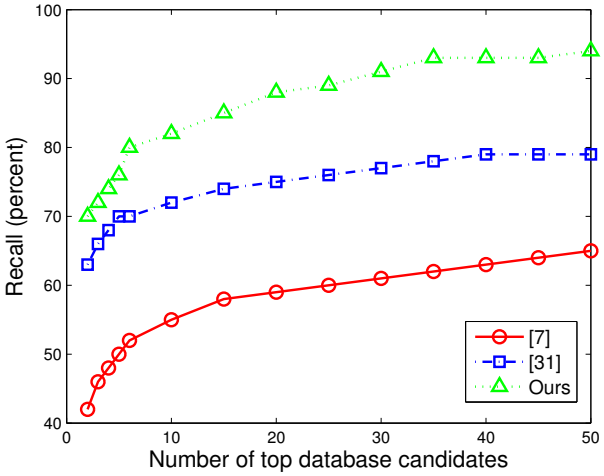
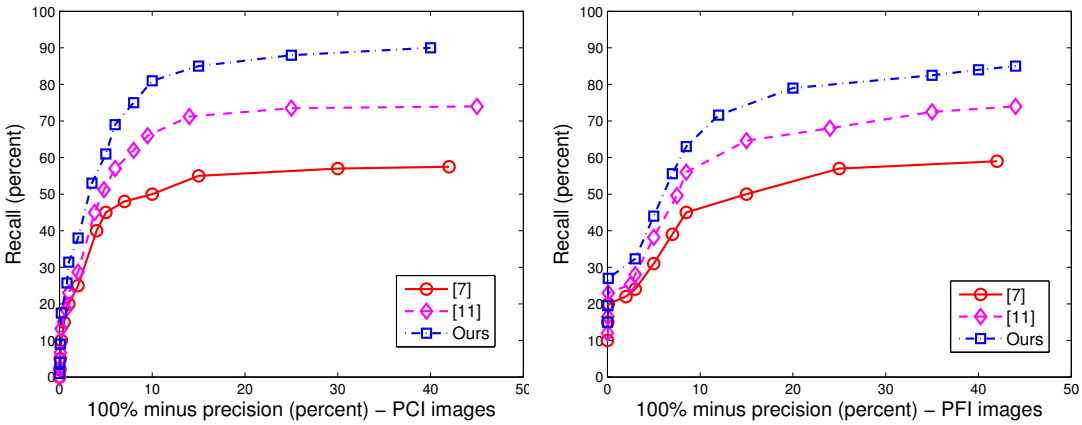


Figure 3. Geo-location recall rates on the San Francisco dataset as a function of the number of retrievals within which closest matching training neighbors are present. *Best viewed in color.*

Flickr videos. Results of our approach are given in Figure 5. We also tested our approach using video and geo-tags separately, and the results were on average 10% and 17% lesser than using the two modalities together.

3.4. Discussion

It can be seen that our approach consistently outperforms existing techniques on diverse datasets with different location patterns and data modalities. To test the sensitivity of our approach to parameter variations, we additionally experimented with k values 4 and 16, d accounting for energy between 95% and 99.5%, and uniformly sampling 3 and 7 points on the geometric prior. The results for different combinations of these parameters, on all the three datasets, resulted in a performance reduction of around 4.5% on average, which in many cases was still better than existing ap-



(a) (b)

Figure 4. Precision recall curves for geo-location on the San Francisco dataset corresponding to (a) perspective central images (PCI) and (b) perspective frontal images (PFI). *Best viewed in color.*

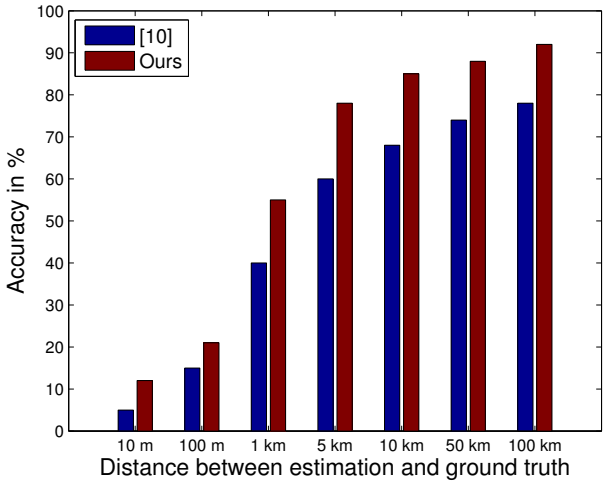


Figure 5. Heterogeneous geo-location accuracy on the MediaEval dataset as a function of distance between estimated location coordinates with that of the ground truth. *Best viewed in color.*

proaches. This sheds light on the robustness of our approach to parameter choices. For all the above experiments, we also performed 1-nn on the hierarchical sparse codes V directly instead of training an SVM. The performance dropped by around 15% on average which highlights the utility of label information, even if coarse, to perform inference. Computationally, we could perform location inference of a test image in about 3 seconds on a single 2GHz machine. Some examples of correct and incorrect location estimates for test data are shown in Figure 6.

3.4.1 Utility of geometric prior

We then analyzed the impact of the geometric prior. Instead of projecting training data x_i on $\mathcal{T}$ to obtain X_i^*, we gave x_i as the input to hierarchical sparse coding. This resulted in an average reduction in performance of 14% across all three datasets. To see if $\mathcal{T}$ could help other existing techniques as well, we implemented the method of [12] that learns per-location classifiers with x_i as input, and then with X_i^* as input. We saw an average increase in performance of 6.3%. These experiments strongly convey the utility of learning transformation priors between appearance space and location space. We then studied if parallel transport is indeed the best way to obtain such a prior. For this, we used the geodesic $\gamma(t)$ between the means of $\mathcal{A}$ and $\mathcal{L}$, μ_A and μ_L as a single transformation instead of obtaining several transformations using parallel transport. The results decreased by 8.9% on average across all datasets, thereby reinforcing the benefits of our approach for obtaining the prior (Sec 2.1).

3.4.2 Utility of hierarchical sparse coding

We then studied if hierarchical feature learning is essential, by working with a single layer sparse coding approach. The results reduced by 9.5% on average across all datasets. We then studied if sparse coding is indeed essential, by training SVM directly on Z^* instead of V. For this case the results reduced by 21% on average. These studies show the utility of hierarchical feature learning using sparse coding. While there are discriminative methods to learn sparse codes, we learn them in a generative manner in Sec 2.2 so that we could use them for further processing with heterogeneous modalities (Sec 2.3) and for inferring novel locations discussed next.

3.4.3 Transferring knowledge to novel locations

We now study how our model trained on specific locations can be used to perform inference on new locations that were not a part of training. For this we separately considered z from the im2gps and San Francisco datasets, and randomly

chose half the number of clusters (32) for training purpose and the remaining half for testing. We repeated this process ten times separately for both the datasets. For each trial, we learnt the SVM classifier on the training data sparse codes. We then obtained the similarity vectors for the test data (Sec 2.2.1), and grouped them into 32 clusters using normalized cuts. Ideally, we would like the grouping results to match with the ground truth clusters. We obtained average clustering accuracy of 68%, using the method of [33], which signifies that the model learnt on different locations is generalizable enough to new locations. While we stayed at the cluster level to compute the accuracy, we calculated the location co-ordinate level accuracy by using the following process. For each similarity vector obtained for novel location test data, we computed its four closest neighbors from the test set based on similarity vectors and computed the average difference between the ground truth location of the test data with that of its neighbors. Using this distance, our results were about 55% accurate within a 100km error tolerance zone.

We then performed experiments by training on one dataset and testing it on another, and using the data features prescribed by the training dataset. We did two trials by using im2gps dataset as training and San Francisco dataset as testing, and vice-versa. Results for the first case was 73% cluster-level accuracy and 58% location level accuracy, while for the second case it was 51% and 38% respectively. This could be because im2gps dataset has data samples pertaining to San Francisco city and hence we do have some knowledge about the novel test set, albeit not dense. Whereas for the case with San Francisco dataset for training, we have far less knowledge to extrapolate for im2gps locations outside of San Francisco city. This also explains why the results for this case is less than the results obtained for within a dataset split between training and testing that was discussed in the previous paragraph.

4. Conclusion

We have addressed the challenging problem of location recognition by learning transformational priors capturing relations between image appearance and their location, and encoding them using hierarchical feature learning mechanism based on sparse coding. Besides obtaining state-of-the-art results in three public datasets and justifying the reasons behind our design choices, we also demonstrated the utility of our approach in handling heterogeneous data modalities and in transferring knowledge gained on known locations to unknown locations, which are important practical problems. Through this work we stress that while deep statistical feature learning methods are very efficient in handling big data problems, it is equally important to feed them with strong geometric priors pertaining to the actual application that one is trying to address.

References

[1] P.-A. Absil, R. Mahony, and R. Sepulchre. Riemannian geometry of grassmann manifolds with a view on algorithmic computation. *Acta Applicandae Mathematica*, 80(2):199–220, 2004.

[2] M. Bansal and K. Daniilidis. Geometric urban geo-localization. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3978–3985, 2014.

[3] Y. Bengio. Learning deep architectures for ai. *Foundations and trends® in Machine Learning*, 2(1):1–127, 2009.

[4] A. Bergamo, S. N. Sinha, and L. Torresani. Leveraging structure from motion to learn discriminative codebooks for scalable landmark classification. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 763–770, 2013.

[5] M. A. Brubaker, A. Geiger, and R. Urtasun. Lost! leveraging the crowd for probabilistic visual self-localization. In *CVPR*, pages 3057–3064, 2013.

[6] S. Cao and N. Snavely. Graph-based discriminative learning for location recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 700–707, 2013.

[7] D. M. Chen, G. Baatz, K. Koser, S. S. Tsai, R. Vedantham, T. Pylvanainen, K. Roimela, X. Chen, J. Bach, M. Pollefeys, et al. City-scale landmark identification on mobile devices. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 737–744, 2011.

[8] K. Crammer and Y. Singer. On the algorithmic implementation of multiclass kernel-based vector machines. *The Journal of Machine Learning Research*, 2:265–292, 2002.

[9] M. Elad. *Sparse and redundant representations*. Springer, 2010.

[10] G. Friedland, J. Choi, and A. Janin. Video2gps: a demo of multimodal location estimation on flickr videos. In *ACM International Conference on Multimedia*, pages 833–834, 2011.

[11] R. Gopalan. Learning cross-domain information transfer for location recognition and clustering. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 731–738, 2013.

[12] P. Gronat, G. Obozinski, J. Sivic, and T. Pajdla. Learning and calibrating per-location classifiers for visual place recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 907–914, 2013.

[13] J. Hays and A. A. Efros. Im2gps: estimating geographic information from a single image. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008.

[14] J. Hays and A. A. Efros. Large-scale image geolocalization. In *Multimodal Location Estimation of Videos and Images*, pages 41–62. Springer, 2015.

[15] A. Irschara, C. Zach, J.-M. Frahm, and H. Bischof. From structure-from-motion point clouds to fast location recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2599–2606, 2009.

[16] R. Jenatton, J. Mairal, G. Obozinski, and F. Bach. Proximal methods for hierarchical sparse coding. *The Journal of Machine Learning Research*, 12:2297–2334, 2011.

(a) (b)

Figure 6. Examples of test data with (a) correct and (b) incorrect location estimates from our approach. While the first two rows in (a) are from common tourist places, the next two rows aren't. Images in (b) are noticeably harder.

[17] E. D. Johns and G.-Z. Yang. Pairwise probabilistic voting: Fast place recognition without ransac. In *European Conference on Computer Vision*, pages 504–519. 2014.

[18] H. Karcher. Riemannian center of mass and mollifier smoothing. *Communications on pure and applied mathematics*, 30(5):509–541, 1977.

[19] A. Li, V. I. Morariu, and L. S. Davis. Planar structure matching under projective uncertainty for geolocation. In *European Conference on Computer Vision*, pages 265–280. Springer, 2014.

[20] X. Li, C. Wu, C. Zach, S. Lazebnik, and J.-M. Frahm. Modeling and recognition of landmark image collections using iconic scene graphs. In *European Conference on Computer Vision*, pages 427–440. Springer, 2008.

[21] Y. Li, N. Snavely, and D. P. Huttenlocher. Location recognition using prioritized feature matching. In *European Conference on Computer Vision*, pages 791–804. 2010.

[22] D. Lin, A. Kapoor, G. Hua, and S. Baker. Joint people, event, and location recognition in personal photo collections using cross-domain context. In *European Conference on Computer Vision*, pages 243–256. Springer, 2010.

[23] T.-Y. Lin, S. Belongie, and J. Hays. Cross-view image geolocalization. In *CVPR*, pages 891–898, 2013.

[24] H. Nasr and B. Bhanu. Landmark recognition for autonomous mobile robots. In *International Conference on Robotics and Automation*, pages 1218–1223, 1988.

[25] U. Nehmzow, T. Smithers, and J. Hallam. *Location recognition in a mobile robot using self-organising feature maps*. Springer - Information Processing in Autonomous Mobile Robots, 1991.

[26] K. Ni, A. Kannan, A. Criminisi, and J. Winn. Epitomic location recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(12):2158–2167, 2009.

[27] Y. C. Pati, R. Rezaiifar, and P. Krishnaprasad. Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition. In *Asilomar Conference on Signals, Systems and Computers*, pages 40–44. IEEE, 1993.

[28] G. Schindler, M. Brown, and R. Szeliski. City-scale location recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–7, 2007.

[29] G. Schroth, R. Huitl, D. Chen, M. Abu-Alqumsan, A. Al-Nuaimi, and E. Steinbach. Mobile visual location recognition. *IEEE Signal Processing Magazine*, 28(4):77–89, 2011.

[30] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.

[31] A. Torii, J. Sivic, T. Pajdla, and M. Okutomi. Visual place recognition with repetitive structures. In *CVPR*, pages 883–890, 2013.

[32] C. Wang and S. Mahadevan. Heterogeneous domain adaptation using manifold alignment. In *International Joint Conference on Artificial Intelligence*, page 1541, 2011.

[33] L. Xu, J. Neufeld, B. Larson, and D. Schuurmans. Maximum margin clustering. In *Advances in Neural Information Processing Systems*, pages 1537–1544, 2004.

[34] J. Yang and M.-H. Yang. Learning hierarchical image representation with sparsity, saliency and locality. In *British Machine Vision Conference*, pages 1–11, 2011.

[35] T. Yeh, K. Tollmar, and T. Darrell. Searching the web with mobile images for location recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages II–76, 2004.

[36] K. Yu, Y. Lin, and J. Lafferty. Learning image representations from the pixel level via hierarchical sparse coding. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1713–1720, 2011.

[37] A. R. Zamir, S. Ardeshir, and M. Shah. Gps-tag refinement using random walks with an adaptive damping factor. In *CVPR*, pages 4280–4287, 2014.