VOCABULARY TREE INCREMENTAL INDEXING FOR SCALABLE LOCATION RECOGNITION

Rongrong Ji[1)], *Xing Xie*[2)], *Hongxun Yao*[1)], *Yongjian Wu*[3)], *Wei-Ying Ma*[2)]
[1)] Harbin Institute of Technology, No.92, West Dazhi Street, Harbin, 150001, P. R. China
[2)] Microsoft Research Asia, Sigma Building, No.49, Zhichun Road, Beijing, 100080, P. R. China
[3)] Wuhan University, Luojia Hill, Wuhan, 430072, P. R. China

ABSTRACT

This work aims at developing a scalable vision-based location recognition system where the backend database can be updated incrementally. Our proposed framework enables incremental indexing of vocabulary tree model, which efficiently includes new data into model refinement without re-generating entire model from overall dataset. An adaption trigger criterion is presented to lessen system computational cost, which is achieved by density-based relative entropy estimation between original dataset and newly coming data. Experiments on *Seattle* urban scene datasets with over *20K* street-side images show the effectiveness of our work.

Index Terms—scene retrieval, location recognition, vocabulary tree, incremental indexing, data similarity

1. INTRODUCTION

Coming with the popularization of mobile devices, the success in patch-based image retrieval [5] facilitates vision-based location recognition in recent years [1-4, 8]. Generally speaking, given a scene dataset in which each scene contains multiple images and is associated with a GPS location: In server-end computer, images are offline indexed beforehand using patch-based scene recognition model [1]. In user-end client, user takes a photo of his surroundings online via camera-embedded PDA or mobile phone. This photo is transmitted to the server, in which patch-based recognition process is carried out to identify its belonging scene. Consequently, its corresponding GPS location, together with other location related information (such as shopping information and restaurant comments), are returned to the user as query results.

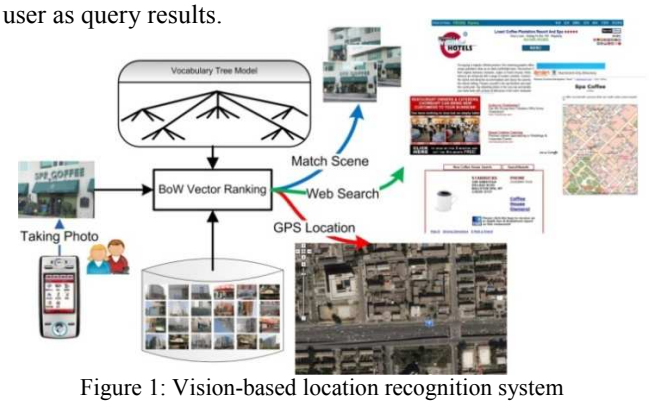

Figure 1: Vision-based location recognition system

To ensure sufficient coverage of urban city scene, gigantic scene image corpora are demanded. It requires the scene recognition model to be effectively constructed and maintained in large-scale scenario. In such case, vocabulary tree (VT) is popular in literature [1, 8] to address large-scale requirement. Images in the database are scanned offline to extract salient regions (such as DoG [7]) and then descriptors (such as SIFT [7]). These descriptors are quantized by hierarchical k-means clustering to generate the vocabulary tree, which produces "visual words" (quantized clusters with SIFT features) to represent each image as a *Bag-of-Word* (*BoW*) vector. In retrieval, image similarity is evaluated by the *Cosine* distance between their *BoW* vectors.

One important extension to improve system flexibility is to enable the recognition model to be maintained in a scalable and incremental scenario. In such case, scene images from different sources such as web search results and user query examples can be incrementally uploaded to extend scene dataset in server.

This paper presents a unified solution to adapt vision-based location recognition system to dataset variation. We investigate *How* (adaption implementation algorithm) and *When* (adaption trigger criterion) to update recognition model in such incremental scenario, based on which a unified solution for recognition model adaption is proposed, including both system level architecture and theoretical adaption algorithm, as well as trigger criterion.

The rest of this paper is organized as follows: Section 2 presents our model adaption algorithm. Section 3 presents our adaption criterion. Experiments are presented in Section 4. Finally, this paper concludes in Section 5.

2.VOCABULARY TREE INCREMENTAL INDEXING

2.1 Incremental Indexing Algorithm

Facing an incremental dataset, in the case that the recognition system receives new image batchs, a nature thought is to re-train the recognition model all over again. Although good performance can be expected, the computational cost would be extremely unbearable once the dataset volume is gigantic. Especially, to maintain our system for online service, such cost would be unbearable.

A replacement of overall model re-training is that, we only use vocabulary tree to generate the *BoW* vectors for new images, without updating or re-training the VT-based recognition model. However, once the distribution of new image patch is diverse comparing to original dataset, the performance would be continuing to decrease with the

coming of new image batches.

Different from above methods to deal with the incremental dataset, we address this problem by enabling our model to be scalable to data variation. We utilize s*calable* in this paper to indicate that the recognition model should be adaptive to data increase & removal in an incremental dataset. To achieve scalability in vision-based location recognition, adaption algorithm is demanded to upgrade the patch-based scene recognition model in such incremental scenario. A novel vocabulary tree incremental indexing algorithm is presented to fit vocabulary tree based recognition model to renewed data distribution.

Model adaption methods have been proposed in pattern recognition literature [10, 11] for refining the model trained from, usually, high-dimensional data. For instance, Cadez[10] utilized Kolmogorov-Smirnov statistics to evaluate distribution similarity and based on which update parameters of mixture model. Campbell [11] adopted linear programming to detect data novelty in incremental dataset. However, in patch-based scene / object recognition, little works has been conducted to address this issue. In vocabulary tree based scene recognition, model efficiency depends on the fitness of tree structure to current data distribution [6]. Once the original dataset is renewed by adding new data batch, the overall data distribution diversity would largely affects model efficiency and recognition performance. To maintain an effective recognition model over renewed dataset, vocabulary tree adaption is necessary.

Firstly, SIFT features of a new data batch are re-indexed using the original tree, based on which the new TF-IDF term weightings of each word is calculated. The frequency of each word reveals its rationality of existence and necessity of further expanding. Subsequently, words in vocabulary tree that contain overabundant or over-limited features are adapted to fit the renewed distribution.

Three operations are defined to iteratively refine the model structure to fit renewed data distribution:

1). *Leaf Split*: If number of features contained in a leaf node is higher than maximum threshold L_{max}, cluster features of this node to m leaves in its sub-level (m is the same branching factor as in vocabulary tree construction).

2). *Leaf Delete*: If the feature frequency of a newly-generated leaf is lower than a pre-defined minimum threshold L_{min}, its features are reassigned to the nearest leaves within the sibling nodes of this deleted leaf.

3). *Parent Withdraw*: If 1).the feature frequency of a newly-generated leaf is lower than minimum threshold, and 2).this leaf is the only child of its parent, we withdraw this leaf and degrade its parent as a new leaf.

Table 1: Vocabulary tree incremental indexing

Input: SIFT feature set of new data batches.
For each feature in new data batches{
  Re-index this feature vector over the VT hierarchy
  Increase the feature frequency of the nodes in VT hierarchy that this feature goes through in indexing}
Go through all leaf nodes of VT{
  If its feature frequency is lower than L_{min} or higher than L_{max}, **push** this node into the *Operation Array*.}
While the *Operation Array* is not empty{
  Get the first element
  If its feature frequency is higher than L_{max}**{**
    Leaf Split, **push** new leaves into *Operation Array***}**
  If its feature frequency is lower than L_{min}**{**
    If there are siblings of this node**{**
      Leaf Delete, **push** renewed leaves into *Operation Array***}**
    Else{
      Parent Withdraw**}}}**
  Delete this first element**}**
Output: Refined vocabulary tree after adaption.

2.2 System-Level Implementation

In system-level implementation, we should figure out a preliminary question about the genesis of new data.

Generally speaking, our system collects incremental scene images as well as their GPS locations from three following sources:

1). Scene images uploaded by system administrators, which are carefully selected and treated as *Fully Trusted*.

2). Query images sent by users to the server-end computer, which are considered as *Under Evaluated*.

3). Images periodically crawled from Flick®. We use both scene name and city name as crawling criterion. Such data source is also viewed as *Under Evaluated*.

For those *Under Evaluated* scene images, pre-processing is conducted to further filter irrelevance: Treating each newly coming image as query example, scene recognition process is simulated in server, in which the *Cosine* distance between this query and the best matched image is compared with a maximum diversity threshold T_{max}. If it is larger than T_{max}, we discard this image; otherwise we add this image into the *Fully Trusted* image set, which is treated as the new data batch to update the database.

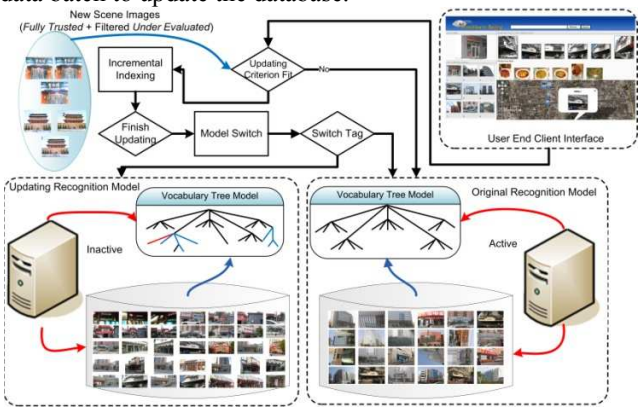

Figure 2: Recognition model updating

To provide consistent service along with the process of incremental indexing, our system maintains two central computers in server-end. Each maintains a location recognition system that is settled to be identical once finishing one-round adaption. Initially, the status of one system is set as *Active* while the other as *Inactive*. *Active* means this server program is now utilized to provide service; *Inactive* means this server program is now utilized for

incremental indexing. Once the *Inactive* program receives a new batch of scene images, we firstly investigate whether it is necessary for activate incremental indexing process (Detail in Section 3). If so, we conduct incremental indexing and switch its *Inactive* status to *Active*, and vice versa; otherwise, these images are temporally stored and added into consequent new image batches.

3. UPDATING CRITERION BASED ON RELATIVE ENTROPY ESTIMATION

Facing a new image batch, it is not always necessary to activate the incremental indexing process in *Inactive* server. Recalling that the vocabulary tree can be regard as a data driven model, if the distribution of new data is almost identical to that of original dataset, adaption could be postponed with the consequent coming images.

Hence, to reduce the computational cost in gigantic data corpora, we conduct model adaption once either of following cases follows: 1). the volume of new images is large enough; 2).the distribution of new images is extremely diverse from that of original dataset. We present in this section an adaption trigger criterion using Kullback–Leibler diversity based relative entropy estimation.

On its first step, data distribution is measured by its sample density, which is further discretely approximated by point density. Initially, a *Density Field* of current dataset in SIFT space is estimated and approximated by the density of each SIFT point, in which the density of a SIFT point in 128-dimentional SIFT space is defined as:

$$\mathrm{D(i)} = \frac{1}{\mathrm{n}} \sum_{\mathrm{j=1}}^{\mathrm{n}} \mathrm{e}^{-||\mathrm{x_i}-\mathrm{x_j}||_{\mathrm{L2}}} \qquad (1)$$

where $D(i)$ is the point-density of i^{th} SIFT point; n is the total number of SIFT points in this dataset; $\mathrm{x_j}$ is j^{th} SIFT point. We adopt *L2* distance to evaluate the distance between two SIFT points.

To reduce computational cost, we estimate the density of each SIFT point by its local neighbors as an approximation:

$$\widetilde{\mathrm{D}}(\mathrm{i},\mathrm{m}) = \frac{1}{\mathrm{m}} \sum_{\mathrm{k=1}}^{\mathrm{m}} \mathrm{e}^{-||\mathrm{x_i}-\mathrm{x_j}||_{\mathrm{L2}}} \qquad (2)$$

where $\widetilde{\mathrm{D}}(x,m)$ is the point-density of i^{th} SIFT feature in its m neighborhood. By neighborhood approximation, point based density is estimated. Their m nearest neighbors are stored for *Density Field* updating of new data batch.

Consequently, we evaluate the data dissimilarity between original dataset and new data batch by their density-based KL-like relative entropy estimation as:

$$\mathrm{Diver_{Accumulate}} = \sum_{\mathrm{i=i}}^{\mathrm{n}} (\widetilde{D}_{\mathrm{new}}(i,m) \log \frac{\widetilde{D}_{\mathrm{new}}(i,m)}{\widetilde{D}_{\mathrm{org}}(\mathrm{Nearest}(i),m)} \qquad (3)$$

in which $\widetilde{D}_{\mathrm{new}}(i,m)$ is the density of new data at i^{th} data point in m^{th} neighborhood; $\widetilde{D}_{\mathrm{org}}(\mathrm{Nearest}(i),m)$ is the density of old data at the nearest old point of i^{th} new data in m^{th} neighborhood. It can be observed from the above equation that data diversity increases as: (1). The volume of new data batch increase; (2). The distribution of original dataset and new data batch are more diverse.

Based on data diversity evaluation, we control the incremental indexing process by trigger criterion as follows:

Table 2: Trigger criterion test

Input: SIFT feature set of new data batches.
For each point *i* in the new dataset**{**
  Estimate the density of *i* by *m* neighborhood approximation
  Search the nearest original point *Nearest*(*i*)
  Calculate i^{th} part of Eq.3, add to $\mathrm{Diver_{Accumulate}}$**}**
If $\mathrm{Diver_{Accumulate}} \geq \mu_{\max}$ (The maximum tolerant diversity)**{**
  Carry out Vocabulary Tree Incremental Indexing
  Go to **Output }**
Else
{ Add current data to next batch images **}**
Output: Updated Vocabulary Tree

When fusing new data batch into original dataset, the point density in original dataset need not be updated. Indeed, their former density estimations can be partially preserved, only need to be modified by new data as:

$$\widetilde{D}_{\mathrm{Update}}(i,m) = \widetilde{D}_{\mathrm{Org}}(i,k) + \widetilde{D}_{\mathrm{New}}(i,m-k) \qquad (4)$$

k is the number of remaining original points in m nearest neighbors, which is achieved by comparing the new data with the former-stored m nearest neighbors of each point.

4. EXPERIMENTS

Dataset and Experimental Setup: In our scene retrieval experiments, we use *Scity* urban scene dataset taken from Seattle city, which contains over 24,000 street-side photos captured along Seattle streets by a car automatically. Every six successive photos are grouped as a scene (totally 3, 000 scenes). We resized these photos to 400×300 pixels and extracted 306 features from each photo on average.

We divide *Scity* dataset into two parts: (1). The *base set* contains 9,000 images for the construction of original vocabulary tree. There are 3,000 scenes, in each of which we initially pick 3 images of of 6 to represent the scene; (2). The *Incremental Set* contains 15,000 images, which is randomly divided into 15 image batches. These image batches are incrementally sent to our system for model adaption, each of which contains 1,000 images. We initially cover all scenes in *Scity* dataset, based on which new images are sequentially added to these scenes for incremental indexing. To evaluate indexing performance, we sample the images from the remaining *Incremental Set* after each round's incremental indexing as the query set, which is utilized to test algorithm performance.

We build a 2-branch, 12-level vocabulary tree for *Scity*. In tree construction, if a node has less than 2,000 features, we stop its k-mean division, no matter whether it achieves the deepest level or not. The *Density Field* is initially calculated using the overall dataset and consequently updated with new data batch as in Section 3. In tree adaption, the maximum threshold L_{max} is set as 20,000; the minimum threshold L_{min} is set as 2,000. On a computer with Intel PIV Xeon 3.06GHz CPU and 2.0G RAM, a typical time and memory cost for a database with 5K images is: tree constructing time 8 hours, retrieval time 3 seconds per image (including feature extraction), and memory cost 100M.

We use Success Rate at N (*SR@N*) to evaluate system performance. This evaluation measure is commonly used in

evaluating Question Answering (QA) systems. *SR@N* represents the probability of finding a correct answer within the top *N* results. Given *n* queries, *SR@N* is:

$$SR@N = \frac{\sum_{q=1}^{n} \theta(N - pos(a_q))}{n} \quad (5)$$

a_q is the answer of qth query, $pos(aq)$ is its position, θ() is Heaviside function: (x) =1, if x≥0, and $\theta(x)$=0, otherwise.

Results and Discussion: We compare our proposed incremental indexing with methods of (1). Re-clustering of entire renewed dataset. It has the best performance as the performance upper limit; (2). Original VT that only re-generates *BoW* vectors for new images. It has no adaption in tree structure and is the performance baseline. We not only compare their recognition performance using SR@1 (Fig.3) but also compare their adaption computation cost using their computational costs (Fig.4) as follows:

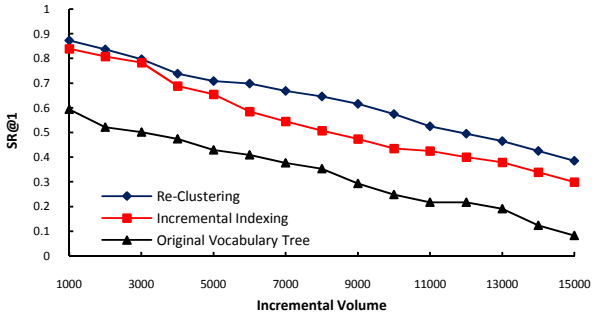

Figure 3: Sequential incremental indexing without trigger criterion

Fig.3 presents the performance comparison between our proposed method and (1) & (2) methods. With the increase of dataset volume, the SR@1 performance is by nature decreased. Although re-clustering performs better than our proposed method, its time cost is unbearable comparing to both incremental indexing and simply using original VT to generate new *BoW* vectors (Fig.4). Our method can maintain comparable performance (less than 10% degradation than re-clustering) while requiring fairly limited computational cost (24% times cost comparing to re-clustering).

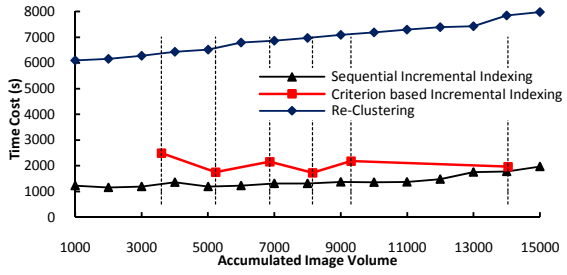

Figure 4: Computational comparison with/without trigger criterion, each node represents the time cost of an occurred VT adaption.

We also compare our proposed adaption trigger criterion (Section 3) with simply sequential adaption in this subsection (Fig.4-5). Fig.5 presents the SR@1 performance comparison of three above methods, in which the startup of each of above three methods are triggered by the relative-entropy-based criterion. Comparing with the sequential incremental indexing method in Fig.5, criterion-constrained method performs equally well. Furthermore, our method reduces large amount of computational cost. As presented in Fig.4, only 6 adaption operations are conducted comparing to former 15. It saves over 70% computational cost while performing better than sequential incremental indexing at 4, 6, 7, 14 batches (Fig.3 vs. Fig.5), which is the case that the trigger criterion is satisfied and the criterion-based incremental indexing is finished.

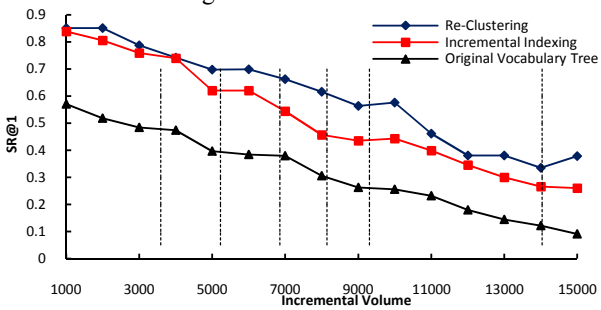

Figure 5: Incremental indexing with trigger criterion, each node indicates a test performance, a dash lines indicate a VT adaption

5. CONCLUSION

Our work in this paper enables vision-based recognition system to be adaptive to scalable dataset in an incremental scenario. Two key issues in proposed method is *How* and *When* to make such adaption, which are addressed by vocabulary tree incremental indexing as well as relative entropy based distribution diversity evaluation. In system architecture implementation, we propose a server switch mechanism to maintain durative workability during incremental indexing. Experimental results over 30K city scene images demonstrate that: Incremental indexing o profitable and essential for maintaining vision-based location recognition system adaptive to dataset variance.

6. REFERENCES

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

[2] Duncan Robertson and Roberto Cipolla. An image-based system for urban navigation. *BMVC*, 2004

[3] Hao Shao, Tomas Svoboda, Tinne Tuytelaars, and Luc J. Van Gool. Hpat indexing for fast object/scene recognition based on local appearance. *CIVR*, PP.71-80, 2003.

[4] W. Zhang and J. Kosecka. Image based localization in urban environments. *International Symposium on 3D Data Processing, Visualization and Transmission*, 2006.

[5] K. Mikolajczyk, B. Leibe, and B. Schiele. Local features for object class recognition. *ICCV*, p1792-1799, 2005.

[6] D. Nister and H. Stewenius. Scalable recognition with a vocabulary tree. *CVPR*, 2006.

[7] D. G. Lowe. Distinctive image features form scale-invariant keypoints. *IJCV*, 20(2):91-110, 2004.

[8] Menglei Jia, Xin Fan, Xing Xie, Mingjing Li, Wei-Ying Ma, Photo-to-search: Using camera phones to inquire of the surrounding world. *Mobile Data Management*, 2006.

[9] L. Kennedy, M. Naaman, S. Ahern. et.al. How flickr helps us make sense of the world: context and content in community contributed media collections. *ACM Multimedia*, 2007.

[10] Igor Cadez, and P. Bradley. Model based population tracking and automatic detection of distribution changes. *NIPS* 2001.

[11] C. Campbell and K. P. Bennett, A linear programming approach to novelty detection. *NIPS* 2001.