A Novel Efficient Visual Object Tracking Retrieving Scheme for Cloud Storage

Prof.Suvarna Patil[1], Netra Kalyanshetty[2], Meghana Kale[3],Snehal Dhole[4], Komal Bobade[5]
Department of Computer Science
[1, 2, 3, 4, 5]Dr.D.Y. Patil Institute of Engineering Management And Research ,Pune-411033

Abstract-*As an important application in local computing, local storage offers user scalable, flexible, and high-quality data storage and computation services. A growing number of data owners choose to outsource data files to the remote server third party storage. Because third party storage servers are not fully trustworthy, data owners need dependable means to check the possession for their files outsourced to remote servers. To address this crucial problem, some remote data possession checking (RDPC) protocols have been presented. But many existing schemes have vulnerabilities in efficiency or data dynamics. In this project we are hiding the file information in the images using visual cryptography and then we are circulating the image to the users. Circulation of the image will be done depending upon the number of users.*

We will divide the image into different parts depending upon the users. Once the image is divided we will send it to all users. Once the image is send to all users we will ask user to upload the image. If users try to download the image and it is not uploaded by all users it will get error message. Once it is uploaded by all the users than it will be downloaded by all. Retrieving objects from large image collection is challenging due to the so-called background-interference, The proposal set with maximum objectness score and minimum redundancy is obtained. Our method is featured by the fine partitioning, which not only removes interferences from background, but also significantly reduces the number of objects to index.

Keywords:Object retrieval, object proposal, partitioning, reranking.

I. INTRODUCTION

As an important application in local computing, local storage offers user scalable, flexible, and high-quality data storage and computation services. A growing number of data owners choose to outsource data files to the remote server third party storage. Because third party storage servers are not fully trustworthy, data owners need dependable means to check the possession for their files outsourced to remote servers. To address this crucial problem, some remote data possession checking (RDPC) protocols have been presented. But many existing schemes have vulnerabilities in efficiency or data dynamics.

In this project we are hiding the file information in the images using visual cryptography and then we are circulating the image to the users. Circulation of the image will be done depending upon the number of users. We will divide the image into different parts depending upon the users. Once the image is divided we will send it to all users. Once the image is send to all users we will ask user to upload the image. If users try to download the image and it is not uploaded by all users it will get error message. Once it is uploaded by all the users than it will be downloaded by all.

Retrieving objects from large image collection is challenging due to the so-called background-interference, i.e., matching between query object and reference images is usually confused by cluttered background, especially when objects are small. So we propose an object retrieval technique addressing this problem by partitioning the images. Specifically, several object proposals are partitioned from the images by jointly optimizing their objectness and coverage. The proposal set with maximum objectness score and minimum redundancy is obtained.Our method is featured by the fine partitioning, which not only removes interferences from background, but also significantly reduces the number of objects to index. In this way, the effectiveness and efficiency are both achieved, which better suits big data retrievalof an IoT based network is the integration of heterogeneous devices causing the complication of the integration process.

II. LITERATURE SURVEY

The first Remote Data Possession Checking (RDPC) was proposed by Deswarte et al. [1] based on RSA hash function. The drawback of this scheme is that it needs to access the entire file blocks for each challenge.The provable data possession (PDP) model was presented by Ateniese et al. [4], which used the probabilistic proof technique for remote data integrity checking without accessing the whole file. In addition, they supplied two concrete schemes (S-PDP, E-PDP) based on RSA. Although these two protocols had good performance, it’s a pity they didn’t support dynamic operations and this scheme still did not support block insert

operation.lots of research works [5]–[7] devoted to construct fully dynamic PDP protocols. For instance, Sebé et al. [5] provided a RDPC protocol for critical information infrastructures based on the problem to factor large integers, which is easily adapted to support data dynamics. Erway et al. [9] first presented a fully dynamic PDP scheme (DPDP) by using authenticated skip list, which allowed data owner to append, delete, insert and update file blocks at anytime. Wang et al. [6] used Merkle hash tree (MHT) to propose another dynamic method for remote data checking, in which each block was hashed to be a leaf node of MHT. By sorting all leaf nodes from left to right, the MHT implicitly identified the block position which is essential for dynamic operations. However, using MHT caused heavy computation cost. Yang and Jia [10] presented an efficient scheme, in which an index table was utilized to support dynamic operations. By the index table, the data owner recorded the logical location and versionnumber for each block for the outsourced file. However, to delete or insert one data block, the verifier had to find the position of the block and shift the remaining entries to insert or delete a row in the index table, which still incurred high computation cost.

Chen et al. provided a dynamic RDPC scheme by using homomorphism hash function defined in [7]. Unfortunately, their scheme was proved insecure by Yu et al. [2]. To overcome the drawback, Yu et al. [11] presented a new RDPC protocol based on RDPC scheme in [7] and proved the security. They also used MHT to achieve data dynamic operations, which caused the same shortcoming of inefficient as in [6].

III. EXITING SYSTEM

KeyGen(1k,λp,λq,m,s) →(K,sk). The data owner executes this algorithm to initialize the system and generate keys

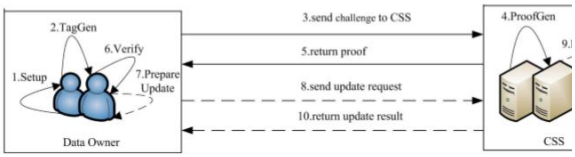

Fig. 2. Work procedure of our RDPC protocol

It inputs security parameters k, λp, λq, the message sector number m and a random seed s, and outputs the homomorphism key K and private key sk. Here the seed s serves as a heuristic "proof", which the hash parameters are selected truthfully [7]. TagGen(K,sk, F) → T: This algorithm is executed by the data owner to produce tags of the file. It inputs the homomorphic key K, private keysk and file F, and outputs the tag set T which is a sequential collection for tag of each block. Challenge(c) → chal: The data owner executes the algorithm to generate the challenge information. It takes the challenged blocks count c as input and outputs the challenge chal. ProofGen(F,T,chal) → P: The CSS executes this algorithm to generate the integrity proof P. It inputs the file F, tag set T and the challenge chal and outputs the proof P. Verify(K,sk,chal, P) →{ 1,0}: The data owner executes the algorithm to check the integrity of the file using the proof P returned from CSS. It takes homomorphism key K, private key sk, challenge chal and proof P as inputs, and outputs 1 if P is correct, otherwise it outputs 0. Prepare Update(Fi,i,UT)→URI:The data owner runs this algorithm to prepare dynamic data operations on data blocks. It takes new file block Fi, the block position i and the update type UT as inputs, and outputs the update request information URI. The parameter UT has three optional elements: insert, modify and delete. Exec Update(URI) →{Success, Fail}: The CSS runs this algorithm to execute the update operation. It inputs URI and outputs execution result. If the update operation is finished successfully, it returns Success, otherwise returns Fail.

IV. PROPOSED SYSTEM

Task partitioning is a way of organizing work that consists in decomposing tasks into sequences of smaller sub-tasks. Task partitioning is usually coupled with task allocation strategies: when a task is partitioned into multiple sub-tasks, workers must be assigned to each of these subtasks. The key difference between task partitioning and task allocation is that task allocation is a way of organizing the workforce, while task partitioning organizes the way work itself is performed.

Our proposed System based on a methods for selecting between a partitioning and a non-partitioning strategy. In this work, the overall task will be pre-partition into two sub-tasks, and one of the two strategies will select on the basis of its costs. In a follow up of this work, we also plan to formulate the problem of choosing whether to employ task partitioning as a dependant problem and show that algorithms employ in the literature to tackle multi-armed(dependant) problems can be successfully employ.

V. METHODOLOGY

Chunking:

Chunking is a process to partition entire file into small pieces of chunks. For any data de-duplication system, chunking is the most time consuming processes since it has to traverse entire file without any exception. The process time of chunking totally depends on how the chunking algorithms break a file. Moreover, the smaller the size of a chunk has, the

better result a de-duplication system has. Increasing the number of chunks, however, results in increasing the processing time for both schemes which we presented in previous section. For the hash-based de-duplication systems, increasing the number of chunk also means increasing the size of lookup table, and then the systems need to spend more time to perform the comparisons. The worst case is that the systems cannot load entire lookup table into memory when the size of the lookup table becomes very huge. In this case, the systems have to pay most expensive costs in disk I/O. The content-aware systems face the similar tradeoffs while performing the block-to-block comparisons. In other words, a good chunking algorithm has to satisfy certain conditions such as minimizing the processing time, balancing the scalability and de-duplication ratio, and controlling the variations of chunk-size These categories have been studied and researchedGenerally speaking, whole-file chunking is the simplest and fastest, but it has the worst results regarding de-duplication ratio. The de-duplication ratio of the fixed-size chunking is totally depending on what the fixed-size is. The smaller the fixed size is, the better de-duplication ratio has. Again, the fixed-size chunking faces the tradeoffs in balancing thecapacity scalability and the de-duplication ratio. Moreover, the most important issue of these two chunking algorithms is the boundary shifting problem . We introduce this issue and then discuss the variable size chunking in next section

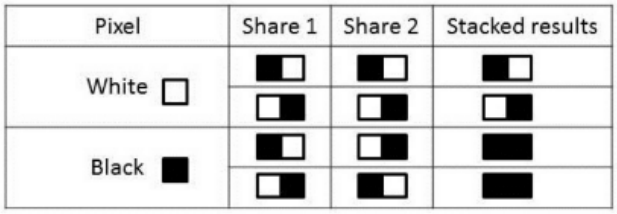

Fig. 1. Basic (2,2) VC scheme with 2 subpixels

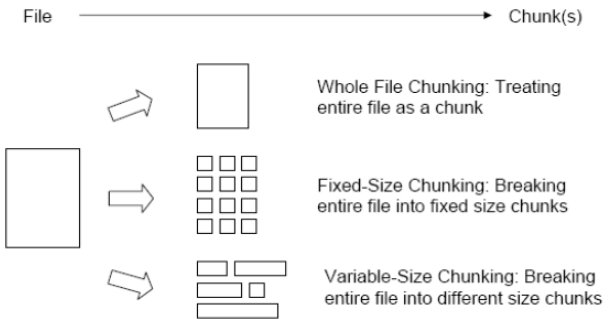

Figure 3: Three Different Chunking Categories.

Visual cryptography:

Visual cryptography schemes allow the encoding of a secret image, consisting of black or white pixels, into n shares which are distributed to a set of n participants. The secret pixels are shared with techniques based on the intelligent subdivision of each secret pixel into a certain number of subpixels. Each share is then composed of black and white subpixels, which are printed in close proximity to each other, so that the human visual system averages their individual black/white contributions. White color means transparent, so that the superposition of white pixels, let the color of the pixel contained in the other shares pass through. The shares are such that only qualified subsets of participants can "visually"recover the secret image, but other subsets of participants, called forbidden sets, cannot gain any information about the secret image by examining their shares.The shares can be conveniently represented with an n × m matrix S where eachrow represents one share, i.e., m subpixels, and each element is either 0, for a white subpixel, or 1 for a black subpixel. A matrix representing the shares is called distribution matrix[11]

VI. CONCLUSION

We have presented a object retrieval framework to address the "un-secure" problem by adapting state-of-the-art image retrieval techniques to object retrieval. Our solution performs in the manner of partitioning, that divides the whole image into tens of object candidates which are retrieved/reranked separately and independently to the background. More effective feature representation and reranking in the context of objet retrieval.

REFERENCES

[1] Hao Yan, Jiguo Li, Jinguang Han, Member, IEEE, and YichenZhang ," A Novel Efficient Remote Data Possession Checking Protocol in Cloud Storage", IEEE Trans. Info.forensic& security, VOL. 12, NO. 1, JANUARY 2017

[2] Zhiyong Chen, Wei Zhang, Bin Hu, Xiaochun Cao, Senior Member, IEEE, Si Liu, and Dan Meng," Retrieving Objects by Partitioning" IEEE TRANSACTIONS ON BIG DATA, VOL. 3, NO. 1, JANUARY-MARCH 2017

[3] Tong Liu, Xiaochun Cao, Senior Member, IEEE, and Jianmin Jiang," Visual Object Tracking With Partition Loss Schemes", IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, VOL. 18, NO. 3, MARCH 2017

[4] G. Atenieseet al., "Provable data possession at untrusted stores,"inProc. 14th ACM Conf. Computer. Commun. Secur. (CCS), 2007,pp. 598–609

[5] F. Sebé, J. Domingo-Ferrer, A. Martinez-balleste, Y. Deswarte, andJ. Quisquater, "Efficient remote data

possession checking in critical information infrastructures," xKnowl. Data Eng., vol. 20,no. 8, pp. 1034–1038, Aug. 2008.

[6] Q. Wang, C. Wang, K. Ren, W. Lou, and J. Li, "Enabling public auditability and data dynamics for storage security in cloud computing," IEEE Trans. Parallel Distrib. Syst., vol. 22, no. 5, pp. 847–859,May 2011.

[7] L. Chen, S. Zhou, X. Huang, and L. Xu, "Data dynamics for remote data possession checking in cloud storage," Comput. Electr. Eng., vol. 39,no. 7, pp. 2413–2424, 2013.

[8] Y. Yu, J. Ni, M. H. Au, H. Liu, H. Wang, and C. Xu, "Improved security of a dynamic remote data possession checking protocol for cloud storage," Expert Syst. Appl., vol. 41, no. 17, pp. 7789–7796, 2014.

[9] C. Erway, A. Küpçü, C. Papamanthou, and R. Tamassia, "Dynamic provable data possession," in Proc. 16th ACM Conf. Comput. Commun.Secur. (CCS), 2009, pp. 213–222.

[10] K. Yang and X. Jia, "An efficient and secure dynamic auditing protocol for data storage in cloud computing," IEEE Trans. Parallel Distrib. Syst.,vol. 24, no. 9, pp. 1717–1726, Sep. 2013.

[11] Stelvio Cimato1, James C.N. Yang2, and Chih-Cheng Wu2," Visual Cryptography Based Watermarking" Y.Q. Shi (Ed.): Transactions on DHMS IX, LNCS 8363, pp. 91–109, 2014