Image Retrieval in the context of Big Data

MUÑOZ Eduardo
Master of Science, CSDS
IMT Atlantique
Brest, France 29280
eduardo.munoz@telecom-bretagne.eu

HOODA Ashish
Master of Science, CSDS
IMT Atlantique
Brest, France 29280
ashish.hooda@telecom-bretagne.eu

***Abstract*—Image retrieval in big data environments is a very researched topic nowadays as the use of data in the form of images is increasing and the traditional methods for image matching are not suited when considering thousands of millions of images. In this paper we summarize some of the key techniques used for image matching in the field of computer vision. Also, we explore some of the most efficient algorithms and architectures used to increase the performance of the image matching process. Finally we propose an architecture that is able to escalate in storage size without compromising the response time of the system.**

Keywords—CBIR, Feature extraction, Indexing, Clustering, V-T(Vocabulary Trees), SIFT, MapReduce, K-D Tree, Bag of Words

I. Introduction

Context Based Image retrieval is the set of methodologies by which a large data set or collection of images is searched for images related to a given input image.

The first mechanism used for this type of system was based on text annotations inserted to images as meta-data. This text approach was a very simple system but have some limitations as the meaning of the images is subjective. An image carries much more information than text. Moreover, human brain can express and comprehend images faster and with greater depth. Because of this, a way to extract image information was necessary. Currently, we have several techniques in the field of image processing that can extract relevant information out of images. Most of these techniques work perfectly for small data set of images.

But the problem sufficiently evolves for very large image collections. The massive amount of digital data being generated today has created a need for developing mechanisms that are able to deal with these type of large data sets, process them and retrieve useful information with satisfactory delay while using minimum resources. The need is for modifications and adaptations in the current techniques and structures or development of entirely new methods to address this problem.

II. State of the Art

A. *Feature extraction*

1) ***Color****:* Color extraction is one of the most important techniques. It classifies digital images based on their color content. It relies on identifying spatial regions in an image that have same color contributions as the query image. This is achieved using transformations and quantizations over the true image data. Moreover, special structures such as color histograms and predefined color sets help in this process. A color histogram is the joint probability function of the intensities of the three primary color values. [11] Based on the type of query, the color extractions technique varies -

- **Global color indexing**
  It refers to the case where the query takes into account the entirety of the image.With global color indexing we can obtain great matches of panoramas, or some natural photos, where the question is related to global image.
- **Regional color indexing**
  In Localized region color indexing the purpose is to have answers when the users are interested in positional information. For example if the user wishes to find images with a sunset in left half or a car in the center.

2) ***Segment****:* Segmentation is a process by which the image is analyzed and every pixel of the image is label so that it belongs to a particular identified region of the image. This process divides an image into homogeneous units based on one or more characteristics. Basic segmentation can be based on characteristics like surface and edge. One type of segmentation is morphological opening and closing. It employs erosion and dilation using a structure element. Other types employ Markov Random Fields or Neural nets. This technique is an excellent approach to understand the relation and order of the objects within the global space.

3) ***Shape****:* Shape representation uses what is called a shape descriptor which are used to represent shape features. Basically shape descriptors are vectors of not many dimensions.

One of the main problems with shape representation is the current difficulty for the systems to correlate shapes when there exists even subtle differences [4]. Usually the systems making use of shape representation for precise applications are specific purpose systems. General systems cant fit for all applications where shape representation could be a powerful.

Important properties of shape features are:

- Ability to Identify.
- Translation, rotation and scale invariance.
- Affine transforms invariance.

4) ***Textures****:* Texture is the representation of objects which have similar patterns and characteristics as brightness, color, size, slope, etc [12]. This visual elements can be grouped together to simulate human perception.

B. Feature extraction Algorithms

*1) **SIFT (Scale-Invariant feature transform):*** SIFT has become very popular as an algorithm for feature extraction specially for image recognition purposes. The underlying idea behind this algorithm is similar to the functioning of primary visual cortex of primates. SIFT transforms an image into a large collection of local feature vectors which are invariant to scaling, rotation or translation of the image. The main stages for this process are:

- Scale-Space Extrema Detection using differences of Gaussians
- Keypoint Localisation
- Orientation Assignment
- Description Generation

C. SURF (Speeded Up Robust Features)

SURF has many similarities to SIFT in the sense that it too relies on the procedure of finding extreme regions and then eliminating non interest points. Instead of using a Gaussian function, SURF uses a Hessian matrix. Also the gradient is further approximated in the form of Box filters [10]. SURF has found a lot of practical use as it is faster and computationally feasible. But the overhead is that it is able to capture less number of features than SIFT.

D. Machine learning

Machine learning systems also can be used in the context of image retrieval, new approaches have emerged allowing the use of more precise systems based on textual annotations.

As we know the main problems of textual annotations if the manual labor and the images perception. With machine learning systems, theses problems can be minimized to get the benefits and expressiveness of textual annotations.

- **Automatic labeling:** with machine learning systems images can be labeled or annotated automatically based on the system knowledge.
- **Relevance feedback:** is process by which the knowledge of the system increments based on user input. There exists a number of classical and new relevance feedback algorithms used for CBIR applications.

E. Multidimensional indexing

The problem with image indexing is in the high number of dimensions that are found in the image analysis approach, we can be talking of dimensions in the order of 1000.

In the process of feature extraction and image index one of the fundamental transformations taken into account is the reduction transform. Various techniques are used to accomplish this - KLT(Karhunen-Loeve Transform) [6] - It focuses on reducing dimension by introducing elements with highest variance possible Clustering - Another technique for dimension reduction

Next come the Multi dimensional indexing algorithms. These include Bucketing algorithm, KBD trees, R-trees. A major challenge for these algorithms is that their performance deteriorates with increasing number of dimensions. Further, the above techniques do not address the case of non-Euclidean similarities. Clustering and neural nets have been used to tackle those kind of features.

III. Image retrieval in Big Data environments

A. Challenges

The traditional approaches used for image retrieval are not suited for large collections of images where the response time is required to be in a certain finite interval.

In the current world the quantity of information in the form of images, and even video is extensive and incrementing everyday. From medical images, to satellite captured images, to the every day pictures uploaded by people, this large set of digital information is collected and stored to be later on processed by the CBIR architecture based on the different use cases defined for the system.

Working with databases of million of images, the methods we've seen can take a considerably large amount of time because of hardware and software constraints. Large tables of multidimensional indexes would take too long to be looked up, even more, sometimes they would not be able to fit into memory.

Having these challenges in mind new techniques are being researched and developed to solve the current issues. The techniques go from a variety of domains, from simpler extraction methods to system architectural considerations. The benefit from this array of improvements is the possibility to use multiple methods in different manners to get the best results.

In the upcoming paragraph we are describing some of the more prominent methods used to day in the context of image retrieval in large image collections. This methods will pave the way to the proposal.

B. Current techniques

*1) **Bag of words:*** The bag of words as described in [1], [3] is a fast method to search for images. It is based on the use of the of bag of words but instead of using words, we will use feature descriptions. After the images features have been extracted (independent of the method), a histogram of features will be build and this histogram will be used to describe the image.

Bag of words can be implemented in different ways to achieve better performance regarding particular contexts:

- **Inverted file** is a common indexing technique used to associate words or features to files.
- **Min hash** is used to describe the images in a less redundant manner than when using IF. Using minimization hashes similar features are treated as the same creating less mappings. This reduces the universe providing faster searches, although could also reduce the precision of the search.

There exists other variations of this method created specifically for very large image collections as those we can find in big data environments. One know reported method is the use of Multiple Dictionaries of Bag of Words as described in [1].

The main advantage using MDBOW is partitioning of the feature space into different dictionaries.

*2) **Scalable Vocabulary Trees**:* The concept of a vocabulary tree is based on treating the features in an image as words. These words are represented hierarchically in trees which are formed by applying clustering algorithms such as K-means to the set of features extracted from the images.

Initially a k-means is run on the root node to branch out k children. Further, each of these children undergo the same procedure to subsequent nodes with k branches at each level. This structure significantly decreases the fraction of images in the database that have to be explicitly considered. In [14], a vocabulary tree with around 10000 visual words is used, with on the order of 1000 visual words per frame, this means that approximately a tenth of the database is traversed during a query, even if the distribution of visual words in the database is uniform. Therefore, while processing the query the closest child is chosen at each level by employing a suitable scoring mechanism. This reduces the query to just traversing the tree and considering only the nodes along the path.

A different type of clustering, Predictive Clustering, has been proposed in this domain. It produces clusters that are homogeneous in both the target and the descriptive space. This architecture reduces the time for similarity value calculation as the computation is required only for the leafs in which the query image descriptors are sorted into. A key point in this technique is that it employs a forest of Scalable Vocabulary Trees(SVTs) rather than a single tree. This prevents major changes in tree structure due to the training data.

This technique has several advantages: it simplifies the searching process, it addresses the notion of updating the architecture to accommodate new information. It could be further improved if integrated with a distributed file system which would bring down the processing time as well as tackle the storage limitations.

*3) **K-D trees**:* In the context of image retrieval the use of K-d trees and optimized K-d trees is very common as this type of structures are an excellent fit to represent geometric data.

Currently K-d trees are most of the time used to store SIFT features [8] which are multidimensional vectors insensitive of scale variations. As the K-d structure is very well fitted to solve the nearest neighbor problem image matching can be performed with very precise results in respect to the query feature.

One of the problems when searching for the nearest neighbors in the k-d structures is the backtracking event that can occurs when searching on the 2 branches of a particular node. Depending on the query vector information and the tree, a lot of backtracking can degrade the run time of the base algorithm used to traverse the tree. With this problem in mind, optimizations have applied [5] in order to get the nearest neighbor in a much faster way, without the need to visit all possible nodes. This action can be also performed heuristically using an error margin.

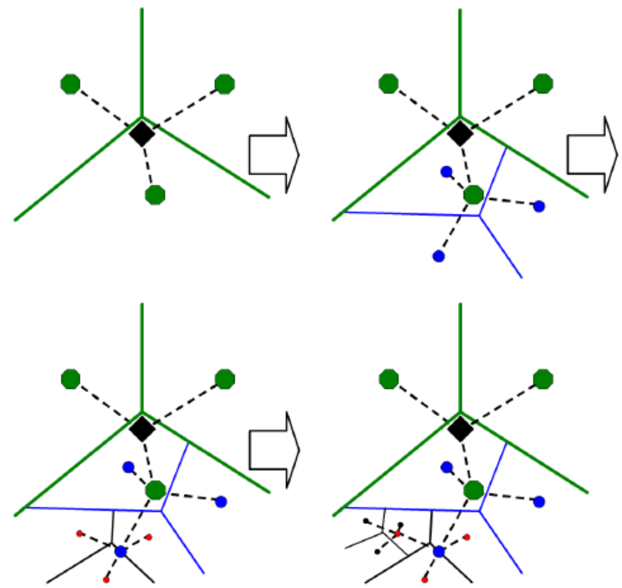

Fig. 1. An illustration of the process of building the vocabulary tree. The hierarchical quantization is defined at each level by k centers (in this case k = 3). [9]

K-d trees can even expand between multiple computers, given that they are basically binary search trees indexed by geometric points, we can make sense of the whole tree as a composition of multiple trees which can live independently of each other. One such method can be seen in [2] where the concept of distributed K-d trees is presented.

In [2] we have two approaches to distributed k-d trees, namely **Independent K-d trees** and **Distributed K-d trees**.

- **Independent K-d trees** are a type of segmentation. In this architecture the total databases of images is partitioned between different nodes. A controller will be the interface to the outer world and every query received by the controller will be distributed to all the nodes. Each node will have an independent tree created from the images in it's own partition.
  Using independent K-d trees the search is parallelized within all the nodes of the cluster. As stated in [2] one drawback is that the controllers can be overloaded, nonetheless this can be solved installing more controllers.
- **Distributed K-d trees** implements a real distributed approach where nodes take charge of their own part of the tree. After the whole tree is computed by the root node, the lower branches the image sets represented by those branches are assigned to clusters of computers. When the root node receives a request it will relay the request to the appropriate peer based on the branch the vector is probably on. One disadvantage of this method is the use of an error margin to prevent backtracking.

*4) **Map/Reduce**:* One of the most used techniques considered nowadays to process large amounts of information is the Map Reduce process. This process has its roots in the

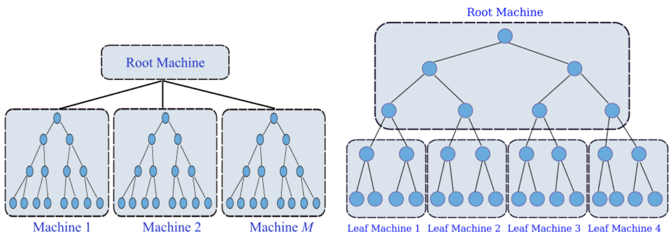

Fig. 2. This figure taken from [2] shows an independent K-d tree to the left and distributed K-d tree to the right

functional programming models where the concept of data transformation is clearly distinguished.

In MapReduce systems, input data is processed in batches in a parallel and distributed environment. The main advantage in this type of architecture is the almost infinite scalability of the system (most of the time only limited by the hardware in place).

When processing input data, each input item is processed by a map control job which maps the input item to a composed key-value pair understandable by the system. The key-value pairs are relayed in orderly fashion to the reduce jobs, which takes care of processing the key-value pair. Every piece of action described in this flow is done asymmetrically.

Currently there is an active research regarding the use of MapReduce systems in different type of big data environments because of the scaling and fast processing benefits [13].

Reports as the one in [7] try to leverage the MapReduce architecture for the purpose of image retrieval, some of the key aspects are:

- **Feature extraction** can be a difficult process to parallelize. Taking into consideration the types of structures used to hold the information. A simple approach will be to store the information in a plain document database and perform brute force image matching.
  Storing the vector information in other type of structures as k-d trees in any of it's variations is not an easy process to parallelize without sharing the memory state.
- **Search process** the search process doesn't couple very well with the MapReduce architecture because of the interactive nature of the search process. As explained before the MapReduce is usually used with data sets that can be batch processed.
  This would mean that the search process should be implemented outside of the MapReduce architecture as was done in [3] or that a specific set of the reduce nodes with very customised properties would be used to perform this operation.

IV. Proposal

Image retrieval is an extensive topic, from features extraction mechanisms, to indexing techniques, there are a plethora of methods that make the process of designing a CBIR system a challenge.

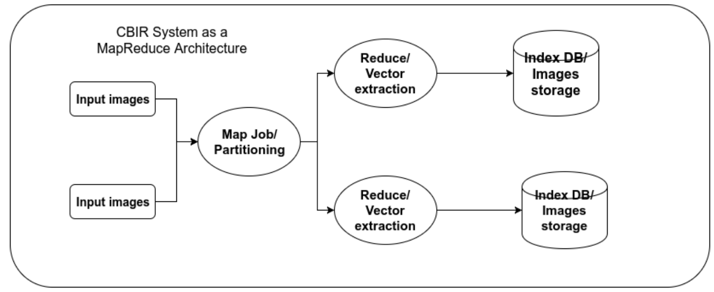

Fig. 3. Image retrieval system in a MapReduce architecture. A map control process receives the input images and partition them. Reduce jobs assigned to specific partitions will extract the vector information to store in that partition database

We propose a parallelized and distributed architecture using a partition scheme to cap computer resources. Figure 4.

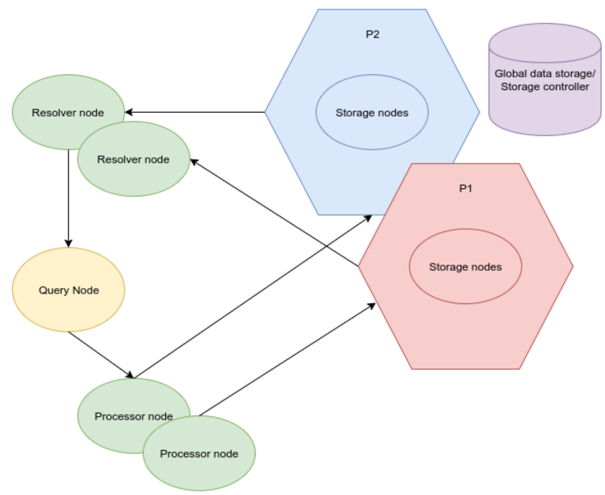

Fig. 4. Customized architecture. Types of nodes and their basic communication directionality

Architecture terminology

- **Global datastore** is a document database holding the images, and their indexes. This database also holds a data field used as the partition ID.
- **Partitions** are basically a subsection of the global image store. Each partition has an associated number of storage nodes and images. Storages nodes compute trees and lookup for image matching taking only into account the images associated to the partition.

Components

- **System controllers** holds information about the state of the system. Systems controllers known the number of images associated at any time to a particular partition, and the number of queries being processed.
- **Storage controller** is used by the storage nodes to retrieve the images associate to a partition. The storage controller is also used by the query node to retrieve the images by the index.
- **Storage nodes** only hold the V-T structure for a partition. This structure is computed from the images

associated to a partition. For caching purposes the storage nodes also holds a list of images already processed and inserted by in the tree. The storage nodes receives queries from the processing nodes in the form of feature vectors that are used in the image matching process.

- **Processing nodes** are the only components aside from the query node that receives the query image in raw form. The processing nodes extract and quantize the image features which are then used to query the storage nodes for possible matches.
- **Query nodes** is actually the input point into the system. The query node will receive query images which are relayed to the processing nodes. Also the query node is the one responsible to return the proper answer to the users of the system.
- **Resolver nodes** receives matching images histograms and index. They use this information to sort results based on histogram matching and build an ordered array of indexes that it sends to the query nodes for image retrieval.

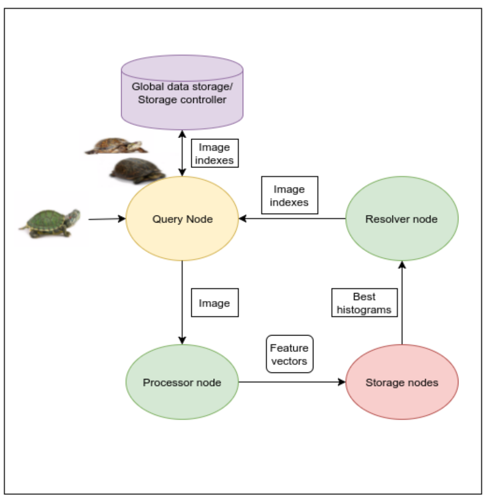

Fig. 5. Message flow in the CBIR system. Transactions from the query node to storage controller when looking up for matching images

The purpose of this architecture is to hold a large amount of images while providing fast response times for query searches. The query node will receive an input query which it will pass to the processing nodes. The processing nodes extract and quantize the features which will be passed to the storage nodes for image matching.

Extracting the features on a separate component makes the relay process lighter and more manageable. All partitions receive the features request and the storage nodes use this information to find the best matching images.

Once finished the storage nodes send the computed histogram of the matching images to the resolver nodes which merge the results off all partitions and sorts the result list. The sorted result list is later on send to the query node which uses the indexes to retrieve the actual images from the storage controller. Figure 5.

V. Conclusion

We have presented in this paper some of the basic concepts of image retrieval in the field of computer vision. Feature extraction techniques as color, texture, shape extraction have been summarized. We also described the SIFT extraction method, which is widely used in the image processing domain.

As the paper mostly takes into consideration the process of image retrieval in big data environments, we presented some of the key challenges of this problem. Later on we have examined some techniques used for fast feature matching. K-d trees have been described in their basic form, and we have described results from other authors regarding tree optimizations. Vocabulary trees were also took into consideration, and the advantages of using V-T's over other structures have been pointed out.

In this paper we also propose a solution to work around the problem of scalability. Our solutions leverages the concepts of distribution and parallelization. We use some of the already presented concepts and define a customized architecture with self awareness capabilities. In our proposal we adopt mainly the use of SIFT vectors with vocabulary trees, a decision that can be changed based on context.

References

[1] Mohamed Aly, Mario Munich, and Pietro Perona. Bag of words for large scale object recognition properties and benchmark. 2011.

[2] Mohamed Aly, Mario Munich, and Pietro Perona. Distributed kd-trees for retrieval from very large image collections. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2011.

[3] Mohamed Aly, Mario Munich, and Pietro Perona. Multiple dictionaries for bag of words large scale image search. *Probe*, 1(D1):P1, 2011.

[4] Ritendra Datta, Jia Li, and James Z. Wang. Content-based image retrieval: Approaches and trends of the new age. In *Proceedings of the 7th ACM SIGMM International Workshop on Multimedia Information Retrieval*, MIR '05, pages 253–262. ACM, 2005.

[5] Rinie Egas, Nies Huijsmans, Michael Lew, and Nicu Sebe. Adapting kd trees to visual retrieval. In *International Conference on Advances in Visual Information Systems*, pages 533–541. Springer, 1999.

[6] Rafael C. Gonzalez and Richard E. Woods. *Digital image processing*. Addison-Wesley Publishing Company, 1992. Autre(s) tirage(s) : 1993 (avec corrections).

[7] Chunhao Gu and Yang Gao. A content-based image retrieval system based on hadoop and lucene. In *Cloud and Green Computing (CGC), 2012 Second International Conference on*, pages 684–687. IEEE, 2012.

[8] David G Lowe. Object recognition from local scale-invariant features. In *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, volume 2, pages 1150–1157. Ieee, 1999.

[9] David Nistr and Henrik Stewnius. Scalable recognition with a vocabulary tree. In *IN CVPR*, pages 2161–2168, 2006.

[10] Edouard Oyallon and Julien Rabin. An analysis of the SURF method. *IPOL Journal*, 5:176–218, 2015.

[11] J. Mukherjee P. Haldar. Content based image retrieval using histogram, color and edge. volume 48, pages 888–975, 2012.

[12] Yong Rui, Thomas S. Huang, and Shih fu Chang. Image retrieval: Current techniques, promising directions and open issues. *Journal of Visual Communication and Image Representation*, 10:39–62, 1999.

[13] Kyuseok Shim. Mapreduce algorithms for big data analysis. *Proceedings of the VLDB Endowment*, 5:2016–2017, 2012.

[14] J. Sivic and A. Zisserman. Video google: A text retrieval approach to object matching in videos. 2003.