International Journal of Informatics and Applied Mathematics
e-ISSN:2667-6990 Vol. 1, No. 1, 47-55

Indexing Multimedia Data with an Extension of Binary Tree – Image Search by Content –

Kouahla Zineddine[1], Ferrag Mohamed Amine[1], and Anjum Adeel[2]

[1] LabSTIC, Computer Science Department, 8 Mai 1945 Guelma University, POB 401 Guelma, Algeria
(kouahla.zineddine, ferrag.mohamedamine)@univ-guelma.dz
[2] Computer Science Department, COMSATS University Islamabad, Pakistan
adeel.anjum@comsats.edu.pk

Abstract. Searching for similar images in a data collection, based on a query image, is a fundamental problem for many applications that use large amounts of complex data. Image research by content and on a large scale is a current challenge for large image database research and management. Various information can be extracted such as colour, shape or texture. A characteristic represents only a part of the image property, which makes it necessary to combine all this information to improve the efficiency of the systems. This paper aims to propose a new indexing structure that allows to organize as much information as possible about the images in a binary tree to improve the search time. Furthermore, an algorithm for index construction and a search algorithm for kNN type queries are introduced. The concept of containers at the sheet level was used to improve the complexity of algorithms. Experiments on real data sets were conducted to determine its performance.

Keywords: Metric Space, Tree, Large Scale, Image, Intrinsic Dimension

1 Introduction

For several decades, indexing techniques have been developed to address research on large data collections. Due to Internet and multimedia device development, large amounts of images have been used in various fields. As a result, there exists a continuous demand for systems that will be able to store and retrieve appropriate images in a short period of time.
Several systems have been developed to meet these needs. However, sometimes it is difficult to express the visual content of images in words. As an alternative way of searching, content based on image retrieval systems (CBIR) has been developed. Content image search systems use visual content, such as colour, shape and texture. Visual content of the images contained in the database are extracted and described by multidimensional characteristic vectors. To retrieve images, users provide the query image retrieval system with [2, 3, 19, 9].
However, the objects to be indexed are often more complex than simple vectors (e. g. sets, graphs) or cannot be simply and usefully concatenated to give a larger vector (e. g. colour histograms and keywords). Therefore, the development of indexing has partially transferred multidimensional spaces to metric spaces, i.e., to exploit the representation of the data itself to work on the similarities that can be calculated between objects.
Intrinsically, the difficulties of multidimensional spaces remain, in a generalized version, while new difficulties arise due to the lack of information on objects. All indexing structures that work on databases known in advance can be directly rejected, either because they need information about the set before the data indexing step [8, 11, 5] that calculates the median of all data to objects in the group.
Thus, indexing structures that only work in the main memory must be rejected, as the purpose is to have a large-scale indexing structure. Then, data structures that are based on a distance matrix calculation between all objects are also rejected, which will immediately lead to algorithms $O(n^2)$. Finally, indexing structures that require discrete distances (even if they have interesting applications) are also excluded.
This article proposes a system for searching images by content, which meet the constraints mentioned above. Section 2 introduces a short taxonomy on content-based image search systems (CBIR), the second part of this section details the different indexing techniques that exist in metric spaces. Section 3 details the GH-tree variant, its pros and cons, the different characteristics of its construction and also its search algorithm. Section 4 discusses the experimental part and the results obtained. Section 5 concludes the paper and provides some research perspectives.

2 Research Study Case : CBIR System

This section introduces a short taxonomy of some existing software and presented on the Internet. It highlights industrial interest in addressing the problem

of large multimedia databases indexing. The results of these products are very difficult to compare for several reasons: - they all use different image databases, - these databases are known entirely to the user, which makes it impossible to calculate efficiency or accuracy.
Some of the existing CBIR systems [12] include: QBIC or Image Content Query, which is the first commercially based content recovery system [18] for colour, texture and shape. VisualSEEK [13] is a text/image oriented search engine. Turn [7] It supports the corresponding colour and location, as well as the corresponding texture.
Netra[17] system uses colour, shape, spatial arrangement and texture. MARS [15] or Multimedia Analysis and Recovery System, uses the corresponding colour, texture and shape. Viper[1] or Visual Information Processing for Enhanced Recovery System. This retrieves images based on the corresponding colour and texture. The img(Anaktisi) [4, 10] is a web-based CBIR system based on different descriptors that includes powerful colour and texture features.

2.1 Indexing Structure

When a massive collection of data is handled, proper indexing will be necessary. There are two main families of structures: one based on clustering objects in space and the other based on partitioning space. A short taxonomy of some techniques can be introduced based on these two partitioning techniques. There are two main cases:

- First class is based on objects clustering using spheres[16, 9]. In this case, the family of M-tree [4] is essentially a balanced index, which allows incremental updates, and it reasonably performs in large dimensions. There is an optimised version of the M-tree, the Slim-trees. Its principle is the reorganisation of objects in order to create a well-balanced index[16].
- Second class is based on the partitioning of space[20] and it is wider than the first class. It contains two sub-approaches: first one uses spheres to share space, such as VP-tree, mVP [14]; second one uses hyper-plan to partition space such as GH-tree [6], GNAT [6].

3 Construction Algorithm

A GH-tree binary tree indexing structure was chosen and it divides the space into two parts, calculating at each iteration two distances d_1, d_2 from the two pivot p_1, p_2. This choice is based on several criteria, as shown in Figure 1:

- Easy to implement.
- Logarithmic complexity in construction and also in research (two distances are calculated each time). As already mentioned, Generalized Hyper-plane (GH) tree uses the principle of the partitioning in metric spaces using hyperplanes. Then, $\mathcal{N}_{GH}$ nodes of a GH-tree are defined in the same method as follows:

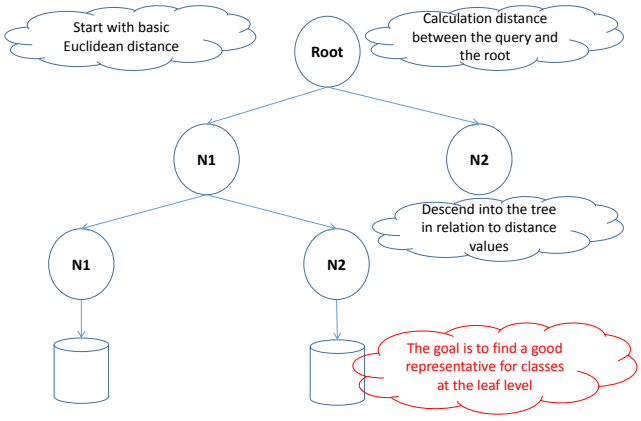

Fig. 1. GH-tree on CBIR system

$$(p_1, p_2, G, D) \in E \times E \times \mathcal{N}_{GH} \times \mathcal{N}_{GH} \tag{1}$$

The size of a leaf node varies between an object and C_{max} objects.
Container concept is a new concept for GH-tree, and did not exist before. This selection is based on several reasons:

- Cluster creation: in this case the leaves in the GH-tree represent similar clusters (image group) with a maximum size of C_{max}.
- Better clusters: the choice of pivots p_1, p_2 depends on the number of images in the container, which allows a proper visualisation of the space.
- Fast access to information: limited leaf size by C_{max}, so minimised tree size, so improved the efficiency of the search algorithm.

3.1 Search Algorithm

Search algorithm responds to a query q to restore k images closest to q when completed. This algorithm is a sequential research version, based on in-depth research and "separation and evaluation" type. As a result, the tree is scanned in pre-order. Initially the search is carried out in the internal nodes, the candidate nodes are selected according to their expected relevance if the image query belongs to the left or right part of the space according to the distance of q from p_1 and p_2.
Once a leaf node is reached, the locally calculated solution is merged with the local pre-solution. It should be noted that the cost of calculation on a container can be controlled, and therefore made relatively low, since it is carried out on a set of objects whose cardinal is limited by the parameter C_{max}.

3.2 Distance Measurements

For image similarity, measurements should be able to measure the similarity between vectors v_1 and v_2 of dimension n. The distance measurement used is the Euclidean distance between different spheres, it is defined as :

$$D\left(v_1,\ v_2\right) = \sqrt{\sum_{i=1}^{n} \left(v_1\left(i\right) - v_2\left(i\right)\right)^2} \tag{2}$$

4 Experiments

To validate the effectiveness of this indexation system, a CoPhIR database of Fliker images was used. CoPhIR is a set of XML files that contains the MPEG7 descriptors. This database has been developed to perform significant tests on the scalability of the SAPIR project infrastructure for similarity research.
As a result, CoPhIR is now available to the research community to compare the different indexing technologies for similarity research, scalability being the key issue.

Table 1. Real data set with their dimensions

MPEG-7 Visual descriptor	**Dimension**
Dominant colour	64
Colour structure	64
Keywords	-
Homogeneous texture	62

Table 2. Index statistics with only DSC descriptor (64 dimension) using corresponding tree

Data	**Construction**	**level**	**Leaves**	**Internal nodes**
1 000 images	14.0s	9	19	40
10 000 images	5m 0 s.	44	95	210
100 000images	50m 0s.	60	211	1120

4.1 Index Quality

Experimentation started with a very important step in the field of indexing, namely index quality. Table 2 presents statistics on the tree structure obtained

when creating an index of three collections, each with 1 000 and 10 000 and 100 000 different images using the dominant 64-dimensional colour.
Next, the authors calculated tree height, internal nodes, leaves number and construction time.
If the maximum size of index container sheets is set at $C_{max} = \sqrt{n}$ objects, a perfectly balanced binary tree will be obtained. However, this depth is no more than twice that of the perfect case.

4.2 *k*NN Search Algorithm Performance

This section presents a variety of performance results for the kNN search algorithm to respond queries. For this task three parameters were varied.
Therefore, each time the average time to respond to a type of one hundred (100 requests) (5NN, 10NN, 30NN) was calculated. As a result, the version without index is the most expensive and the version with index is the most efficient.

4.3 *k* Effect

If k parameter is increased, the number of calculations increases, which results in an increase in search time. But it is observed that the increase is not too large and this parameter does not have a negative influence on this algorithm.

4.4 Results Quality

This section presents the effectiveness of the proposed approach in relation to descriptors' dimensionality. Table 3 shows index construction duration and search

Table 3. Search time with different descriptors for 1000 images

Descriptor	Search time Without Index			Search time With Index		
	k=5	k=10	k=30	k=5	k=10	k=30
Dominant colour	1.15	1.21	1.36	0.8	0.94	1.15
Keywords	0.54	0.55	0.65	0.2	0.25	0.36
Homogeneous texture	1.1	1.2	1.40	0.7	0.85	1.01
Colour structure + Dominant colour Structure de couleur + Couleur dominante	1.2	1.36	1.4	1.21	1.35	1.56
Dominant colour + Colour structure + Text Couleur dominante + Structure de couleur+ Text	1.8	1.89	2.01	1.5	1.58	1.61

time with different k values for five dimension variants. As a result, the construction duration increases with a high dimensionality, as well as the search time.

During the second part of the statistics, presented in Table 4, the precision recall report was calculated in relation to the quality of results obtained. Prototype is more efficient when the size of the descriptor vectors is large, which is perfectly logical. This type structure allows to store visual characteristics, also annotations (keywords). However, maximum use increases the search time.

Table 4. Cophir base accuracy and recovery values

Descriptor	Dominant colour	Text	Homogeneous texture	Colour structure + Dominant colour	Dominant colour + text
Number of relevant images among 10 images	8	9	6	8	9
Precision	90.33%	94.03%	88.66%	89.66%	94.6%
Reminder	13%	15%	5%	8%	16%

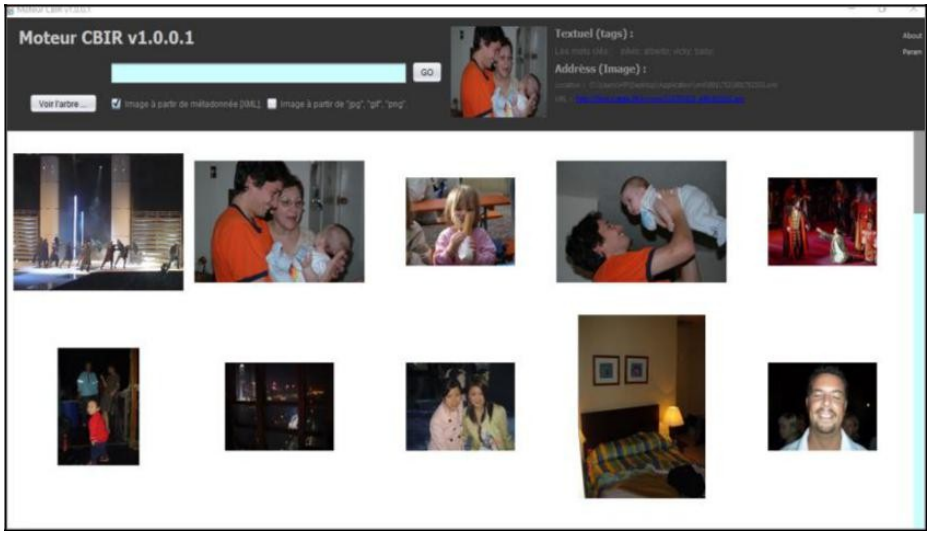

Fig. 2. Running a Cophir base image

Figure 2 shows an example of a query to execute the proposed application using a descriptor mixture.

5 Conclusion

This paper presents a known hierarchy of indexing methods with a variant of the balle-partitioning family, inspired by the recent GH-tree technique, for efficient multimedia data recovery. The proposed technique can be considered as a parameterization of the GH-tree. The number of clusters is based on an analysis of the complexity of the overall research problem.
The algorithms were validated on real-world data sets. Then, the usefulness of collecting as much information as possible about the indexed images was also investigated. The index remains the same, as it is based only on the distance function.
Although a parallel search algorithm on a machine cluster will certainly be beneficial and will be part of the work in the future.

References

1. Baldonado, M., Chang, C.C.K., Gravano, L., Paepcke, A.: The stanford digital library metadata architecture. International Journal on Digital Libraries **1**(2), 108–121 (1997)
2. Berchtold, S., Böhm, C., Kriegal, H.P.: The pyramid-technique: towards breaking the curse of dimensionality. In: ACM SIGMOD Record. vol. 27, pp. 142–153. ACM (1998)
3. Böhm, C., Berchtold, S., Keim, D.A.: Searching in high-dimensional spaces: Index structures for improving the performance of multimedia databases. ACM Computing Surveys (CSUR) **33**(3), 322–373 (2001)
4. Bruce, K.B., Cardelli, L., Pierce, B.C.: Comparing object encodings. In: International Symposium on Theoretical Aspects of Computer Software. pp. 415–438. Springer (1997)
5. Ciaccia, P., Patella, M.: Bulk loading the m-tree. In: Proceedings of the 9th Australasian Database Conference (ADC'98). pp. 15–26. Citeseer (1998)
6. Flickner, M., Sawhney, H., Niblack, W., Ashley, J., Huang, Q., Dom, B., Gorkani, M., Hafner, J., Lee, D., Petkovic, D., et al.: The qbic system. IEEE computer **28**(9), 23–32 (1995)
7. Frankel, C., Swain, M.J., Athitsos, V.: Webseer: An image search engine for the world wide web (1996)
8. Guttman, A.: R-trees: a dynamic index structure for spatial searching, vol. 14. ACM (1984)
9. Hoseinitabatabaei, S.A., Fathy, Y., Barnaghi, P.M., Wang, C., Tafazolli, R.: A novel indexing method for scalable iot source lookup. IEEE Internet of Things Journal **5**(3), 2037–2054 (2018). https://doi.org/10.1109/JIOT.2018.2821264, https://doi.org/10.1109/JIOT.2018.2821264
10. Jabeen S, Mehmood Z, M.T.S.T.R.A.M.M.: An effective content-based image retrieval technique for image visuals representation based on the bag-of-visual-words model. PLoS ONE (13) (2018)
11. Katayama, N., Satoh, S.: The sr-tree: An index structure for high-dimensional nearest neighbor queries. In: ACM Sigmod Record. vol. 26, pp. 369–380. ACM (1997)

12. Niblack, C.W., Barber, R., Equitz, W., Flickner, M.D., Glasman, E.H., Petkovic, D., Yanker, P., Faloutsos, C., Taubin, G.: Qbic project: querying images by content, using color, texture, and shape. In: Storage and retrieval for image and video databases. vol. 1908, pp. 173–188. International Society for Optics and Photonics (1993)
13. Ogle, V.E., Stonebraker, M.: Chabot: Retrieval from a relational database of images. Computer **28**(9), 40–48 (1995)
14. Ogle, V.E., Stonebraker, M.: Chabot: Retrieval from a relational database of images. Computer **28**(9), 40–48 (1995)
15. Pentland, A., Picard, R.W., Sclaroff, S.: Photobook: Content-based manipulation of image databases. International journal of computer vision **18**(3), 233–254 (1996)
16. Satyanarayanan, M., Gibbons, P.B., Mummert, L., Pillai, P., Simoens, P., Sukthankar, R.: Cloudlet-based just-in-time indexing of iot video. In: 2017 Global Internet of Things Summit (GIoTS). pp. 1–8 (June 2017). https://doi.org/10.1109/GIOTS.2017.8016212
17. Smith, J.R., Chang, S.F.: Visualseek: a fully automated content-based image query system. In: ACM multimedia. vol. 96, pp. 87–98. Citeseer (1996)
18. Srihari, R.K.: Automatic indexing and content-based retrieval of captioned images. Computer **28**(9), 49–56 (1995)
19. Weber, R., Schek, H.J., Blott, S.: A quantitative analysis and performance study for similarity-search methods in high-dimensional spaces. In: VLDB. vol. 98, pp. 194–205 (1998)
20. Yang, Y., Zheng, Z., Bian, K., Song, L., Han, Z.: Real-time profiling of fine-grained air quality index distribution using UAV sensing. IEEE Internet of Things Journal **5**(1), 186–198 (2018). https://doi.org/10.1109/JIOT.2017.2777820, https://doi.org/10.1109/JIOT.2017.2777820