Set Enumeration Tree based Image Representation for Gray Level Image Storage and Retrieval

Muhammad Suzuri Hitam, Pong Kuan Peng, Wan Nural Jawahir Hj Wan Yussof, Abdul Aziz K Abdul Hamid and Ghazali Sulong

School of Informatics and Applied Mathematics, Universiti Malaysia Terengganu,
21030 Kuala Nerus, Terengganu, Malaysia.
suzuri@umt.edu.my

***Abstract*—The recent growth of communications and multimedia applications had led to the requirement of mass storage space as well as efficient retrieval technique especially for multimedia data. In this paper, a novel approach for representing gray level image for data storage and image retrieval is proposed. The proposed approach used set enumeration tree data structures where only unique image pattern is stored in the image data structure. The overall structure involves two types of tree data structures; the first tree is low-level image pattern tree to store the unique gray level image pattern and the second tree is used to store the image path by referring to the first tree data structure. The low-level image pattern tree is predefined and will not expand throughout the image encoding process. The size of the second tree is gradually expanded as the result of addition of new image path during image encoding. Through unique image pattern encoding into a tree, there will be no redundant image features, thus leading to saving storing space. Caltech-101 gray level image datasets were used to test the proposed approach and the results showed that it could lead to saving storage space while provide promising performance in image retrieval.**

***Index Terms*—Image Compression; Retrieval; Set Enumeration Tree; Storage.**

I. Introduction

The recent growth of communications and multimedia application had consumed lot of mass storage space and transmission bandwidth, thus resulting high demand of efficient storage and retrieval technique. The use of uncompressed image data as well as storage of similar image patterns leads to the requirement of more storage capacity and transmission bandwidth [1]. Thus, this has led to the introduction of image compression technology to reduce data redundancy in image as the demand of efficient data storage is raised [2]. Even though some of the image data are loss in certain image compression technology, image compression still gained great popularity due to the capability to produce smaller image files size [3, 4, 5, 6, 7]. Image compression can reduce the size of images, thus reducing the storage space and transmission cost. It is crucial to maintain the quality of images at the same time reducing image data size. Therefore, ways of retrieving high quality image in a reasonable amount of time and manageable size are a challenging task.

The demand and requirement of efficient storage and retrieval technique for image is increased, but current techniques yield less than desirable results [8, 9]. To keep the image quality as original as possible, instead of compressing the image data solely, it might be better to change the representation of the image data for better storage and retrieval. In this study, a new approach to represent image is proposed. The new approach is to build a structured dictionary for image representation. The conjecture is to represent image using a new set enumeration tree data structures to provide better storage and retrieval.

This paper will be focused on defining a new unique low-level image pattern tree data structures based on gray level images. The low-level image pattern tree is defined as the 8-bit gray level image that could take image value between 0 and 255. The low-level image pattern tree size will be fixed and not expand throughout the image encoding and decoding processes. All possible combinations of low-level image patterns are included in the set enumeration tree data structures. There will be another set enumeration tree data structures to store image path that associates with the low-level image pattern tree. The image tree data are stored in unique hierarchical structure, thus there will be no redundant image pattern stored in the new image data structures. By using this arrangement, it is hoped that it can reduce inefficient searching and retrieval process that normally caused in traditional image storage where repeated patterns are stored. Image retrieval is achieved by simply searching the image tree according to the defined image path.

The paper is structured as follows: Section II presents related works while Section III describes the proposed approach. Section IV presents the results and discussion. Finally, conclusions are drawn in Section V.

II. Related works

There are many ways of representing image information to ensure efficient image storage and retrieval. Sivic and Zisserman [10] presented an idea of representing images the same way as representing text document, i.e. an image is represented by as set of local features called visual word. Similar to text retrieval approach called Bag-of-Words (BoW) [11], where each document is represented by a vector of occurrence frequency of the words contains in the document. The vector representing the document will be organized as inverted files [12]. Thus, efficient object retrieval from an image is possible by searching relevant visual words.

Graph is also used to represent an image relationship. An image is segmented into objects and the relationship between objects is identified. This approach preferable in medical images. Kumar et al. [13] represented a new relational graph-based algorithm to retrieve co-aligned multi-modality positron emission tomography (PET) and computed tomography (CT) images. Gao et al. [14] proposed the weighted graphs to represent optical coherence tomography (OCT) images for both 2D and 3D images. Grauman and Darrell [15] introduced a pyramid match kernel representation that mapped a set of features to multi resolution image histogram that is robust to image clutter for object recognition. This method is further improved by dividing images and compute histogram repeatedly for recognizing natural scene in [16].

There are many ways of storing image information hierarchically such as using tree data structures. Tree-based index represents the data space of images to form a hierarchical tree structure. The non-leaf nodes act as directory nodes that storing the information of data space, and the leaf nodes storing the information that need to be indexed. The well-known tree-based index is KD-tree [17], R-tree [18], M-tree [19] and B-tree [20].

Quadtree representation [21] is a tree data structure where the image divided into four equal sized quadrants. If the quadrants array not entirely '1' or entirely '0', it will be divided again until entirely '1' or entirely '0' obtained. Tassos and John [22] use quadtree data structures to divide image into variable size and thus can efficiently deal with different image information.

Rymon [23] introduced Set Enumeration tree (SE-tree) to provide a unified search based framework in solving problems where the search space is a subset of the power set, i.e. induces a complete irredundant search technique. SE-tree is simple to use with no redundant data representation. Since its introduction, the SE-tree has been preferred in the field of associative rule mining. Guil and Marin [24] had extended the SE-tree to Temporal Set Enumeration Tree (TSET) extension to mine frequent event-based sequences data. In this paper, we sought to use SE-tree to create a low-level image pattern tree for image representation.

III. SE-Tree Based Image Representation

This section presents the proposed image representation structure using SE-tree based data structure. With this new approach, image will be stored in a SE-tree based data structure. With this format, there will be no redundant image pattern stored, thus anticipated saving storage space. Only the unique image pattern and image pattern path will be stored into the SE-tree based data structure.

In this current implementation, a 1x4 image resolution will be used as the low-level image pattern. With this, the combination of image pattern in each level can be kept to the smallest as possible and thus reduce the complexity of searching the SE-tree image data structure.

A. Set Enumeration Tree

A Set Enumeration tree (SE-tree) structure [23] is a powerful tool for storing and manipulating all kind of data. It has been used across many disciplines for storing and analyzing data. This approach will use two different SE-tree data structure to store image patterns. The first low-level image pattern SE-tree data structure is based on the SE-tree. The expansions of tree depend to a finite set formed by the elements in the first level, see example in Figure 1. Parent nodes link to child nodes to become a new image pattern. As can be seen in Figure 1, the parent node "a" hold three child, parent node "a" link to child node "b" to become a new image pattern "ab". The last parent node will not have a child node. This condition could occur as all possible combination of image patterns had occurred before the last parent node.

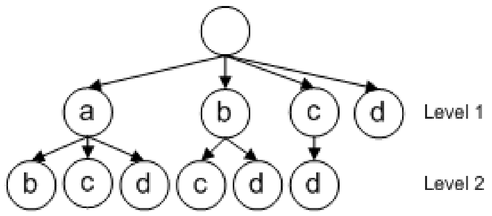

Figure 1: Example of Set Enumeration Tree structure

In the current implementation, the SE-tree based image data structure will have 4 levels. Each node will represent image pattern values of 1x4 resolution. The SE-tree level 1 (parent node) consists of the entire basic images values (Gray level value 0 until 255). The next level will be created following the SE-tree structure, as the tree expand the child nodes will get lesser until the last parent node not holding a child node anymore. This happens as possible combinations image pattern had occurred previously, see example in Figure 2. The combinations of 4 level nodes image value will create a 1x4 low-level image pattern.

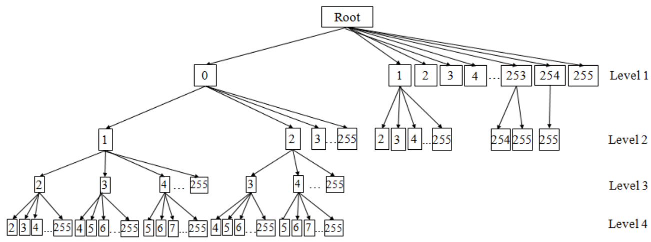

Figure 2: Example of SE-tree image data structure.

B. Image Path Tree

Each of 1x4 low level image pattern can create multiple combinations image pattern. Table 1 illustrates an example of one image pattern (i.e. gray level value 5, 8, 13, 10) where many combinations of image patterns can be created simply by changing the arrangement of the image gray level values. The first tree data structures will only keep the low-level image pattern (i.e. gray level value 5, 8, 13, 10) and the image path will be stored in another SE-tree based data structure.

Table 1
Example of combinations of image patterns

Low-level image pattern	Possible combinations	Image Path
[5,8,13,10]	[5,13,8,10]	[5_1,3,3,2]
	[5,10,8,13]	[5_2,3,3,3]
	[8,13,5,10]	[5_3,3,1,2]
	[13,8,10,5]	[5_3,3,2,1]

The second tree data structures will store image path associates with low-level image pattern. The second SE-tree data structure consists of many combinations image path referring to low-level image pattern. As can be observed in Figure 3, each sub-tree will hold different number of image path that associates with low-level image pattern.

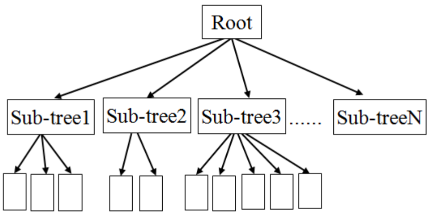

Figure 3: Example of SE-tree image pattern representation.

C. *Tree-like Data Structure Algorithm*

Before image encoding process starts, a low-level image pattern is created. The SE-tree image data structures expand with the images values [0 until 255] following the rule of set enumeration tree structures until no more image patterns is to be stored. During image decoding process, i.e. image retrieval, the final SE-tree will be used.

The process of encoding the SE-tree image data structure is as follows. Firstly, an image *M* is partitioned into a square of 16x16 image pattern blocks. The image pattern blocks may consist of redundant image pattern blocks. A validation process is performed to remove duplicate image pattern block. This is why the number of unique image pattern is different for image pattern blocks, as the number of duplicate image pattern removed are different. Next, a unique 1x4 image patterns will be extracted from 16x16 image pattern blocks. This 1x4 image patterns will go through the low-level image patterns tree to retrieve an image path. Each 16x16 image block will be assigned an image path. Each block will use image path go through the image path tree searching for a match, if image path is not existed in the image path tree, new path will be added into image path tree. Table 2 shows the encoding algorithm of image into a SE-tree image data structure.

Table 2
The encoding algorithm

Algorithm 1: Encoding	
INPUT	Image *M*,
STEP 1	Partition image *M* into *N* blocks of 16x16 images called image pattern blocks. Then, remove redundant image pattern.
STEP 2	A unique 1x4 binary image patterns will be extracted from image pattern blocks.
STEP 3	Matching each image pattern to the low-level image tree. The search results return image path.
STEP 4	Go to STEP 2 until *N* blocks of image block matched. A list of image path is retrieved.
STEP 5	Validate image path. If image path is not existed, then add image path into image path tree.
OUTPUT	Image path tree updated.

D. *Euclidean Distance*

To retrieve similar images from the SE-tree image data structure, Euclidean distance is defined (i.e. Equation 1) to measure the difference between the query image patterns with the image patterns stored in the SE-tree data structures. To simplify the computation process, each image pattern's frequency of occurrence has been normalized into the range of [0, 1].

$$d_{x,y} = \sqrt{\sum_{n=1}^{N} (x_n - y_n)^2} \tag{1}$$

where *x* is query image, y is y= SE-tree image data structure and *N* is the total number of image patterns.

IV. Result and Discussion

In this section, the performance of the proposed approaches is tested on the Caltech-101 datasets [25]. This dataset contains 101 categories image and there are 40 to 800 images per image category. Each image has different image size, as to simplify the computation and standardize it throughout the experiment, all the images in datasets were resized into 256x256 image resolution. A total of 9144 images had gone through the encoding process to be stored in the SE-tree based image data structure.

Figure 4 shows the comparison between the total size of image storage between original image and image stored in SE-tree based image data structure. The size of the image path tree increased as new image path are added in into image path tree. At 2000 images, the tree size had increased around 38% compared to 1000 images. From 2000 images to 3000 images, the tree size had increased around 19%. However from 3000 images to 4000 images the tree size had increased around 23% which is higher increment compared to previous. This mostly due to the lesser redundant image path are found, thus many new image paths are added into the tree. The image path tree grows slowly as the number of image increase. Starting 7000 image upwards the tree grows slowly, the tree size only increase below 10%. It can be seen that an incremental of the image path tree size is less consistent. This is the result of certain image contains less redundant image pattern as compared with others. Thus leading the tree size didn't grow steadily as compared with the original image size which will continue to grow as the number increases.

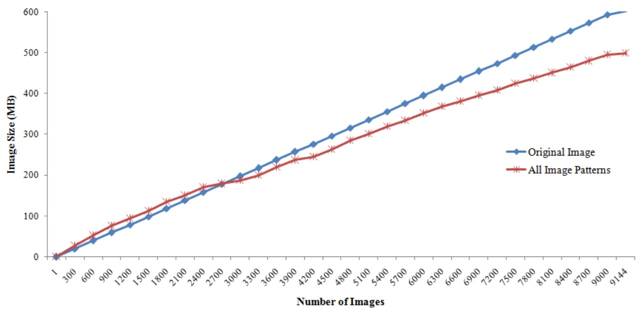

Figure 4: Total size of image path tree for 9144 images.

Figure 5 shown the top ranking of similar images retrieved based on the given query images. In this experiment, only 10 categories were selected to go through image retrieval testing process. The image retrieval process is based on similarity of image pattern stored in SE-tree based image data structure. It can be observed that when the background of the object used different gray level value, it will result in different image pattern been created. Therefore, the image will not be considered as similar images. As seen in Figure 5, image category number 10 only able to retrieve 2 similar images as the other image in the category have different background gray level environment and the object size also differs a lot.

However, for image category number 2 and image category number 3, similar images with different background gray level environment is still able to be retrieved when the object (brain/cup) size are similar and the object have strong features compared to the other images. All the query images are able to retrieve the exact same image. However, the similar images retrieved is not relevant when the image background and object size are too much different. These results are obtained due to the predefined size of the image resolution used is too small.

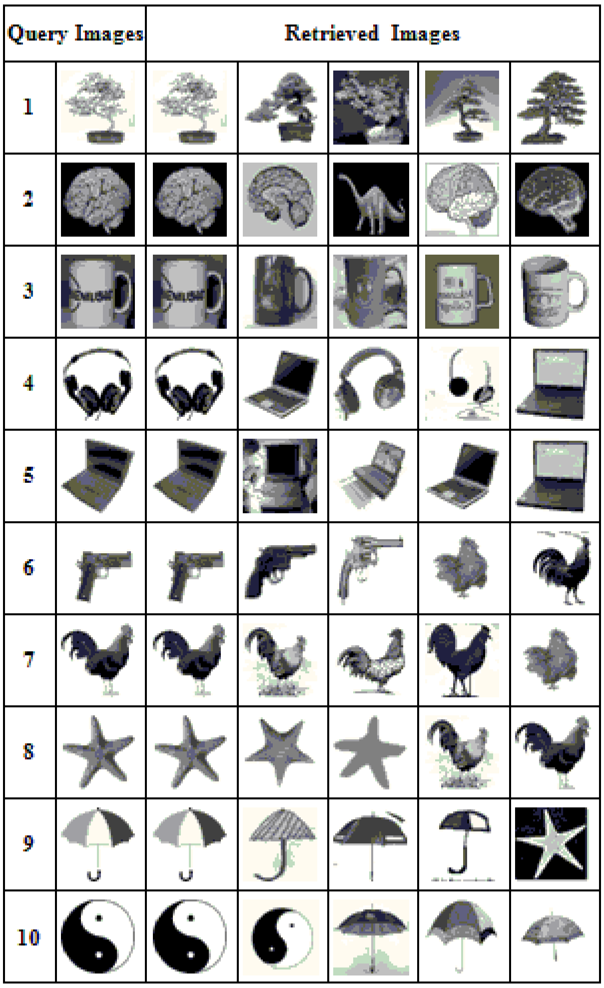

Figure 5: Top 5 similar images retrieved for 10 query images from different image categories

Table 3
Precision and recall for 10 query images

Query Images	Precision (%)	Recall (%)
1	60.73	49.5
2	58.31	49.5
3	65.31	55.5
4	37.31	27.5
5	61.60	51.5
6	41.46	28.5
7	57.86	45
8	41.46	28.5
9	50.22	37.0
10	30.38	23.5
Average	50.47	39.6

V. Conclusions

In this paper, a new approach is introduced to represent gray level image in a SE-tree image data structure so that storage, indexing and retrieval are facilitated. In general, the overall structure of the approaches is performed in three steps. First, a list of low-level image pattern is defined. Next, the unique low-level image patterns will be searched and stored into a SE-tree image data structure. Finally, the SE-tree data structure will act as image dictionary for getting image path. Experimental results showed that the proposed approaches could significantly reduce image storage size. However, this approach still needs improvement to provide more promising approach in retrieving similar objects from the image data.

Acknowledgment

The authors acknowledge the financial support from Ministry of Higher Education Malaysia under Fundamental Research Gant Scheme (FGRS) (vot no. 59288) that has resulted in this research.

References

[1] S. Dhawan, "A review of image compression and comparison of its algorithms," *International Journal of Electronics & Communication Technology*, vol. 2, no. 1, pp. 22-26, 2011.

[2] M. Marimuthu, R. Muthaiah, and P. Swaminathan, "Review article: An overview of image compression techniques," *Research Journal of Applied Sciences, Engineering and Technology*," vol. 4, no. 24, pp. 5381-5386, 2012.

[3] J. Ziv, and A. Lempel, "A universal algorithm for sequential data compression," *IEEE Transactions on Information Theory*, vol. 23, no. 3, pp. 337-343, 1977.

[4] T.A.Welch, "A technique for high-performance data compression," *Computer*, vol. 17, no. 6, pp. 8-19, 1984.

[5] M. Rabbani, and P. W. Jones, *Digital Image Compression Techniques*. Society of Photo-Optical Instrumentation Engineers, Bellingham: SPIE Opt. Eng. Press, 1991.

[6] I.H. Witten, R.M. Neal, and J. G. Cleary, "Arithmetic coding for data compression," *Communication*, vol. 30, pp. 520-540, 1987.

[7] D.A. Huffman, "A method for the construction of minimum-redundancy codes," in *Proceedings IRE*, vol. 40, 1962, pp. 1098-1101.

[8] R.J. Clarke, *Transform coding of images*. San Diego: Academic Press, 1985.

[9] G.K. Wallace, "The JPEG still picture compression standard," *IEEE Transactions on Consumer Electronics*, vol. 38, no. 1, pp. xviii-xxxiv, 1992.

[10] J. Sivic, and A. Zisserman, "Video Google: A text retrieval approach to object matching in videos," in *Proceedings of the Ninth IEEE International Conference on Computer Vision*, vol. 2, 2003, pp. 1470-1477.

[11] P. Indyk, and R. Motwani, "Approximate nearest neighbors: Towards removing the curse of dimensionality," in *Proceedings of 30th annual ACM symposium on Theory of computing*, ACM, 1998, pp. 604-613.

[12] R. Shekhar, and C. V. Jawahar, "Word image retrieval using bag of visual words," in *10th IAPR International Workshop on Document Analysis Systems*, 2012, pp. 297-301.

[13] A. Kumar, J. Kim, L. Wen, M. Fulham, and D. Feng, A graph-based approach for the retrieval of multi-modality medical images," *Medical Image Analysis*, vol. 18, no. 2, pp. 330-342, 2014.

[14] Z. Gao, W. Bu, Y. Zheng, and X. Wu, "Automated layer segmentation of macular OCT images via graph-based SLIC superpixels and manifold ranking approach," *Computerized Medical Imaging and Graphics*, vol. 55, pp. 42-53, 2017.

[15] K. Grauman, and T. Darrell, "The pyramid match kernel: Discriminative classification with sets of image features," in *Proceedings of the 10th IEEE International Conference on Computer Vision*, vol.2, 2005, pp. 1458–1465.

[16] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, 2006, pp. 2169-2178.

[17] C. Silpa-Anan, and R. Hartley, "Optimised KD-trees for fast image descriptor matching," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2008, pp. 1-8.

[18] V.P. SubramanyamRallabandi, and S.K. Sett, "Image retrieval system using R-tree self-organizing map," *Data & Knowledge Engineering*, vol. 61, no. 3, pp. 524-539, 2007.

[19] T. Skopal, and J. Lokoč, "New dynamic construction techniques for M-tree," *Journal of Discrete Algorithms*, vol. 7, no. 1, pp. 62-77, 2009.

[20] C. Douglas, "The ubiquitous B-tree," *ACM Computing Surveys*, vol. 11, no. 2, pp.121-137, 1979.

[21] H. Samet, "The quadtree and related hierarchical data structures," *ACM Computing Surveys*, vol. 16, no. 2, pp. 187-260, 1984.

[22] T. Markas, and J. Reif, "Quad tree structures for image compression applications," *Information Processing & Management*, vol. 28, no. 6, pp. 707-721, 1992.

[23] R. Rymon, "Search through systematic set enumeration," in *Proceedings of 3rd International Conference on Principles of Knowledge Representation and Reasoning*, 1992, pp. 539-550.

[24] F. Guil, and R. Marin, "A tree structure for event-based sequence mining," *Knowledge-Based Systems*, vol. 35, pp. 186-200, 2012.

[25] L. Fei-Fei, R. Fergus, and P. Perona, "Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories," *Computer Vision and Image Understanding*, vol. 106, no. 1, pp. 59-70, 2007.