VP-EMD Tree: An Efficient Indexing Strategy for Image Retrieval

L. Wu, T. Bretschneider

School of Computer Engineering, Nanyang Technological University
N4-02a-32 Nanyang Avenue, Singapore 639798, Singapore
{pg02786079,astimo}@ntu.edu.sg

Abstract: *In order to utilize the large volume of images in databases efficiently, content-based image retrieval (CBIR) has been successfully employed whereby instead of the actual images extracted feature descriptors are used. CBIR systems usually employ multidimensional indexing structures, such as the R*-tree or the SR-tree to index these features and thus to speed-up the query performance. However, the majority of these indexing structures are only suitable for features with fixed length since they mainly employ partitioning methods to divide the multidimensional vector space into sub-spaces. These structures are not appropriate for features with varying characteristic, such as unspecified order of the vector elements and variable lengths. Data with weight information cannot be indexed via current structures either. Hence, this paper introduces a novel indexing structure, the VP-EMD tree. It overcomes the limitations of traditional multidimensional indexing structures and speeds-up the query performance for the features under consideration significantly.*

Keywords: Similarity search, multidimensional indexing techniques, VP-tree, EMD

1 Introduction

Content-Based Image Retrieval (CBIR) is a popular technique in modern image databases. In these systems, features are extracted automatically from images and used to represent those in a more easily utilisable feature space, i.e. the distance between two points that represent features is considered as the degree of similarity for the corresponding images.

In a CBIR system, multidimensional indexing techniques are usually employed to speed up the query performance and many techniques have been introduced, such as the R*-tree and the SR-tree. Most of these existing techniques employ partitioning methods to divide the feature spaces into partitions. Hence they are only suitable for fixed lengths features since boundaries are key parameters that require absolute coordinates. Therefore, current multidimensional indexing structures are limited for fixed features and are not suitable for data with varying characteristic. However, the requirement for features with a fixed length is strict and limits the feature extraction algorithms. On the other hand a dimension reduction is possible as post-processing step. Examples are given by the principal component analysis (PCA) or the utilisation of vector quantisation (VQ). The results are acceptable, but the performance suffers tremendously for CBIR system with a high insertion rate since a permanent adaptation is required, i.e. either by constantly repeating the PCA or optimising the codebook for the VQ. Therefore it is preferable to use the given features directly.

This paper discusses the problems and solution based on the features of the Grid Retrieval System for Remotely Sensed Imagery – $G(RS)^2I$. These features have varying lengths, a partially unfixed order with respect to the vector elements and additional weight information. Current indexing techniques are not suitable for the considered features. This conflict requires a new structure, which is suitable for data with a higher degree of complexity and thus more flexible for the actual retrieval.

This paper is organised as follows: Section 2 briefly introduce the utilised features while Section discusses the employed similarity measure, namely the Earth Mover's Distance (EMD). Section 4 is devoted to discuss the proposed VP-EMD tree and presents indexing measurements. Finally, the conclusion section completes the paper.

2 Indexed Features

This section describes the characteristics of the spectral features and texture features used in the $G(RS)^2I$. Thereby an emphasis is placed on the variable lengths, the partial order among the actual feature's elements and the weight information. Satellite images are used as examples.

2.1 Spectral Features

The spectral features were extracted by locating the dominant spectral characteristics (classes) of an image in the multispectral space. For example, an original satellite image and the corresponding classified image are presented in Figure 1(a) and Figure 1(b), respectively. Table I gives the related statistic information. In total seven dominant spectral classes were located and the centre of each of those was utilised to represent the entire class. The number of corresponding pixels was utilised to represent the weight of the cluster.

Table I: Spectral feature of Figure 1(b)

Cluster No.	Cluster Centre	Weight
1	(20,19,42)	5236
2	(124,37,59)	27907
3	(86,26,47)	34900
4	(165,41,65)	17228
5	(128,169,181)	20
6	(122,107,120)	230
7	(128,63,79)	4479

The classes can be described by the vector $(C_1,W_1),\dots,(C_n,W_n)$, where C_i and W_i are the multidimensional class centres and scalar weights of the i^{th} cluster, respectively. Since two arbitrary C_i and C_j only denote the location in the spectral space, the order of the tuples (C_i, W_i) and (C_j, W_j) is exchangeable. Note that the exchange does not affect the feature, i.e. the spectral feature elements have no order.

(a)

(b)

Figure 1: Classification example:
(a) Original scene, (b) Classified scene

2.2 Texture Features

In satellite images, regions such as urban, agriculture, water as well as mountain areas can be easily described by textures and can provide enormous information for content-based satellite image retrieval. The generated feature vectors have the same characteristic as described previously for the spectral features since the content of a scene is not predefined [2]. Again, the feature's elements do not have a pre-defined order and have arbitrary lengths for different satellite images.

3 Similarity Measurement

In the previous section, spectral features and texture features were briefly introduced. Since the two features can be used to efficiently describe satellite images, it is necessary to find a way for measuring the distance, i.e. the similarity, between two spectral features or two texture features. The traditional Euclidean distance is not suitable since both the spectral features and texture features have weights and the lengths of the feature descriptors are not fixed. However, the Earth Mover's Distance (EMD) [4] is an appropriate distance measure function for comparing features that are constructed like described in Section 2, i.e. $x=\{(x_1,w_1),(x_2,w_2),\dots(x_m,w_n)\}$ and $y=\{(y_1,u_1),(y_2,u_2),\dots(y_m,u_n)\}$, where x_i and y_j are the cluster centres with the scalar weight w_i and u_j, respectively. The EMD can be formally defined to find a flow $F=[f_{ij}]$ between x_i and y_j, which minimises the total cost. $D=[d_{ij}]$ is the ground distance matrix where d_{ij} is the ground distance between x_i and y_j, e.g. the Euclidean distance. Thus the EMD is defined as

$$\mathrm{EMD}(x,y)=\sum_{i=1}^{m}\sum_{j=1}^{n}d_{ij}f_{ij}\Bigg/\sum_{i=1}^{m}\sum_{j=1}^{n}f_{ij} \tag{1}$$

with the constraints

$$\begin{aligned}
& f_{ij}\ge 0, \quad 1\le i\le m, \quad 1\le j\le n \\
& \sum_{j=1}^{n} f_{ij}\le w_i, \quad 1\le i\le m \\
& \sum_{i=1}^{m} f_{ij}\le u_j, \quad 1\le j\le n \\
& \sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij}=\min\left(\sum_{i=1}^{m}w_i,\sum_{j=1}^{n}u_j\right).
\end{aligned} \tag{2}$$

The constraints ensure that the flow is always positive and that not more resources from the source i can be obtained than are available. The same holds true for the sink j that cannot be overloaded. The last constraint specifies the maximal possible flow that is bounded by the minimal total energy of the participating vectors.

4 VP-EMD Tree

With the given characteristic of the feature vectors, current multidimensional indexing techniques are not suitable since they employ boxes or spheres, which are the fundamental element of traditional indexing techniques. However, these cannot be built in a space with varying dimension. Hence, the VP-EMD tree, a member of the vantage point (VP) tree family is proposed to solve the indexing problem of the data under consideration.

4.1 Construction of the VP-EMD Tree

The VP-EMD tree combines the idea of both the VP-tree [1] and the EMD [4]. It employs the EMD function to compute the distance between two data objects and uses the computed distance to partition the feature space. A simple 2-dimensional example is shown in Figure 2.

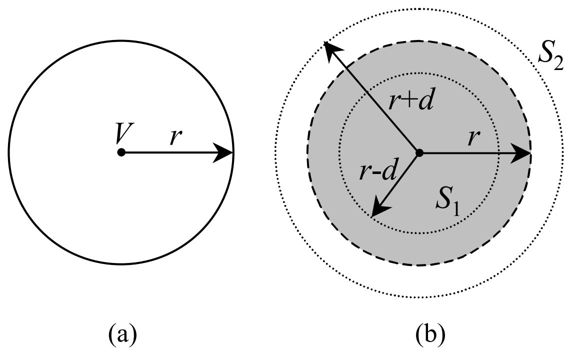


Figure 2: Example for VP-EMD tree:
(a) Partitioning, (b) Querying

In Figure 2, V is the vantage point and r is the median of the EMD values from points to the vantage point V, which is used to partition all the other points into two approximate equal subsets. S_1 and S_2 are chosen according to whose EMD to V is smaller than or equal to r, and larger than r, respectively. Consider the query for the nearest neighbour of querying point q, with the requirement that the maximum distance is smaller than a given value d. With this requirement, the EMD between q and V, denoted as EMD(q,V), is classified into the three cases where EMD(q,V) $\leq r$-d, EMD(q,V) $> r$ and r-$d <$ EMD(q,V) $\leq r$+d. The third case requires the retrieval of both S_1 and S_2. While for the first case, only S_1 needs to be explored. Equivalently for the second case, exploration of S_2 is sufficient, thus effectively pruning one half of the search space. This pruning of the retrieval space is based on the principle of the triangular inequality. Especially, if for $p \in S_2$, EMD(q,V) $\leq r$-d the EMD between p and q is lower-bounded by

$$\begin{aligned} \mathrm{EMD}(p,q) &\geq \left\| \mathrm{EMD}(p,V) \right| - \left| \mathrm{EMD}(q,V) \right\| \\ &\geq \left\| \mathrm{EMD}(p,V) \right| - (r-d) \right| \\ &\geq \left| r - r + d \right| \\ &= d \end{aligned} \tag{3}$$

Similarly, if EMD(q,V) $> r$+d for $p \in S_1$, the EMD between p and q is lower-bounded by

$$\begin{aligned} \mathrm{EMD}(p,q) &\geq \left\| \mathrm{EMD}(p,V) \right| - \left| \mathrm{EMD}(q,V) \right\| \\ &\geq \left\| \mathrm{EMD}(p,V) \right| - r \right| \\ &\geq \left| r + d - r \right| \\ &= d \end{aligned} \tag{4}$$

To each resulting partition, this partitioning strategy is recursively employed until the number of data points in a partition becomes small enough and the cost of a linear scan is acceptable. The difference with respect to other spatial data structure is that at each level, a distinct vantage point is chosen to map the other data points in the subset, rather than employing projections based on absolute coordinate values.

If a query for the nearest neighbour of a giving point has to be evaluated, the EMD between the query point and the vantage point associated with the root is calculated to determine which subset to search next. This procedure is repeated recursively until the nearest neighbour(s) is / are found or no such nearest neighbour whose distance to the query point is smaller than the predefined d exists. In deeper levels, the pruning effect can be multiplied by decreasing d if the tolerated similarity distance is larger than the actual space of the sub-set. Thus unnecessary probing can be avoided.

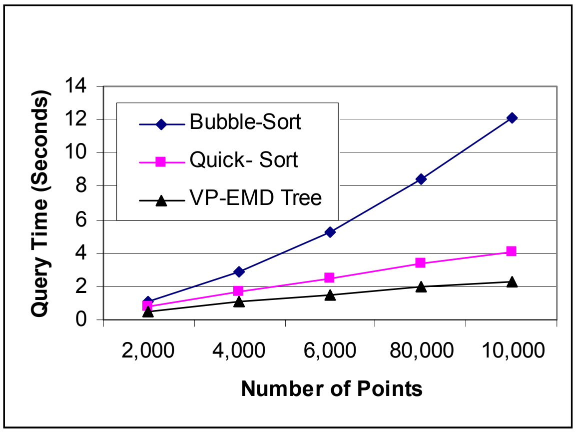
The above binary VP-EMD tree can be generalised to an n-ary VP-EMD tree. The data set S is split into n subsets corresponding to the EMDs between the selected vantage point and other data points. The subsets can be denoted as S_i, i=1…n. Each of the S_i has approximately the same number of data points. As the binary VP-EMD tree, r_i is used to indicate the boundary EMD so that $r_{i\text{-}1} <$ EMD(s,V) $\leq r_i$ $\forall s \in S_i$. With the same partitioning strategy, each of the S_i is recursively partitioned into n smaller subsets.

The choice of vantage points and the value of d affects the performance of the VP-EMD tree notably. A suitable vantage point should divide all points in a set in equally sized subsets. This minimises the number of data points in the concentric regions, and therefore leads the exploration of a smaller number of sub-trees. However, for an increasing data set, it is exceedingly complicate to find the ideal vantage point. Therefore, randomised algorithms are used to approximate the ideal vantage points. The VP-EMD tree shares the algorithm of the original VP-tree proposed by Yianilos [3]. The selection of a suitable d uses the similar solution like the VP-tree and the corresponding algorithm was provided by Chiueh [6].

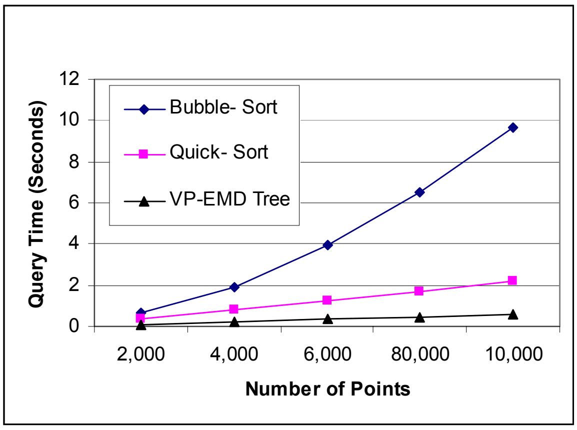
4.2 Experimental Results

The VP-EMD tree was implemented on a Sun Blade1000 workstation with a Sun Ultrasparc III 900M CPU, 1GB memory running a SunOS 5.8 platform. Spectral features and texture features were used to evaluate the query performance, whereby the number of feature points varied from 2,000 to 10,000. Two references were employed to provide the comparisons, i.e. Bubble Sort and Quick Sort. These algorithms were selected since to the author's knowledge there is no other indexing structure that caters for the described data characteristic. Thus for a retrieval of neighbouring feature vectors the sorting algorithms compute the distances and provide a sorted neighbouring list. Figure 3 shows the comparison for texture features and spectral features. For each data set, the ten most similar features were queried.

The Quick Sort is known as one of the most effective sorting algorithm and it was employed in the comparison to referee the improvement of the VP-EMD tree. The comparison shows that the VP-EMD tree has a much better query performance and proved that the VP-EMD tree accelerated the querying performance efficiently. However, an advantage of the sorting approach is that a complete similarity list with respect to the query point is provided while the VP-EMD tree returns only a small sub-set. Nevertheless this is sufficient for most practical case and especially for databases with a large amount of data.



(a)



(b)

Figure 3: Query performance of VP-EMD tree: (a) Texture features, (b) Spectral features

The precision of the VP-EMD tree is another important parameter to evaluate the performance. The experimental results for both spectral and texture showed that the precision of the VP-EMD tree is always 100%. The high precision guarantees that the query results are correct.

The building cost of a VP-EMD tree is a notable parameter as well. Table II shows the building costs for texture and spectral features. For every column in the table, the texture data set and the spectral data set contain the same number of points. But the cluster number of texture features is always ten and the lengths of every cluster is always eight, while the number of spectral features varies from three to eight and each element always contains three entities. Hence, texture features always have more data than the spectral features and the texture features always need more building time. For texture features, the building cost varies from 152 seconds to 639 seconds according to the number of points varying form 2,000 to 10,000. The corresponding building cost of spectral features is in the range of 84 seconds to 385 seconds.

Table II: Building cost of a VP-EMD tree (in seconds)

	2,000	4,000	6,000	8,000	10,000
Tex. feature	152	273	399	518	639
Spec. feature	84	169	239	308	385

5 Conclusions

In this paper, the VP-EMD tree was introduces as an effective multidimensional indexing structure for data with varying lengths, partial order and additional weight information. The VP-EMD tree combines the VP-tree and EMD. It supports *k* nearest neighbour search and explores a new way for the multidimensional indexing techniques by focusing on the distance similarity between data objects, while most current structures use hyper-rectangles or hyper-spheres to partition the feature spaces and focuses on the portioning. Another contribution of the VP-EMD tree is that it employs the EMD, which is an effective function to compute distances, in the research of indexing techniques. The experimental results show that the VP-EMD tree has a good query performance, high precisions and acceptable building costs. The VP-EMD tree solves the indexing problem of multidimensional data with varying characteristic. However, the experimental results show a significant dependency between the building costs and the data size, which is mainly due to the selection of appropriate vantage points.

References

[1] J.K. Wu, "Content-based indexing of multimedia databases", *IEEE Transactions on Knowledge and Data Engineering*, Vol. 9, No. 6, pp. 978–989, 1997.

[2] Y. Li, T. Bretschneider, "Supervised content-based satellite image retrieval using piecewise defined signature similarities", *Proceedings of the IEEE International Geoscience and Remote Sensing Symposium*, 2003 (in press).

[3] P.N. Yianilos, "Data Structures and algorithms for nearest neighbour search in general metric spaces", *Proceedings of the ACM-SLAM Symposium on Discrete Algorithms*, pp.311-321, 1992.

[4] Y. Rubner, C. Tomasi and L.J. Guibas, "The Earth Mover's distance as a metric for image retrieval", *International Journal of Computer Vision*, Vol. 40, No. 2, pp. 99–121, 2000.

[5] B. Liu, T. Bretschneider, "A distributed unsupervised classfication algorithm for optical satellite imagery", *Proceedings of the IEEE International Geoscience and Remote Sensing Symposium*, 2003 (in press).

[6] T. Chiueh, "Content-based image indexing", *Proceedings of the International Conference on Very Large Databases*, pp. 582–593, 1994.

[7] N. Katayama, S. Satoh, "The SR-tree: An indexing structure for high-dimensional nearest neighbor queries", *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pp. 369–380, 1997.

[8] N. Bechmann, H. Kriegel, R. Schneider and B. Seeger, "The R*-tree: An efficient and robust access method for points and rectangles", *Proceedings of the SIGMOD Conference*, pp. 322–331, 1990.