International Journal of Scientific & Engineering Research Volume 10, Issue 6, June-2019
ISSN 2229-5518

Content Based Image Retrieval by Pseudo-Hash in Discrete Wavelet Transform Domain

Mitko Kostov

Abstract— Discrete wavelet transform is used to create pseudo-hashes based on content of images and they are stored in a database. The images are decomposed in few levels and the most important wavelet coefficients are selected. The pseudo-hash contains statistical parameters of the coordinates of the coefficients. When searching a large database for images that match a query-image, the similarity of images pseudo-hashes is considered instead of the images themselves.

Index Terms— Content based image retrieval, ipseudo-hash, query, discrete wavelet transform.

---

1 Introduction

CONTENT-based image retrieval (CBIR) deals with searhing a database for images that correspond to a query image by using visual contents of the images. CBIR requires feature extraction and computation of similarity. The CBIR technology has been used in several applications such as fingerprint identification, digital libraries or medicine.

The property of wavelets to localize both time and frequency makes them very suitable for analysis of non-stationary signals [1]. The basis functions used in wavelet transforms are locally supported; they are nonzero only over part of the domain represented. Hence, an adequately chosen wavelet basis tends to group the coefficients in two groups – one group with a few coefficients with high SNR, and other group with a lot of coefficients with low SNR.

In this paper, a CBIR algorithm based on discrete wavelet transform is proposed. It selects representative wavelet coefficients for an image, which are processed to obtain statistical parameters for the image. They compose an image pseudo-hash information, which is later used for fast searching the database. This approach for searching a database for images in which a query is expressed as a low-resolution image is known as query by content [2], [3], [4], [5].

The paper is organized as follows. After the introduction, the basic definitions of wavelet transform are outlined in Section 2. Section 3 briefly describes the creating of pseudo-hash based on images content and organization of a database. Section 4 presents our experimental results, while the conclusions are given in Section 5.

• *Prof. Dr.sc Mitko Kostov Faculty of Technical Sciences, "St.Kliment Ohridski" University - Bitola, North Macedonia,*
E-mail: mitko.kostov@uklo.edu.mk

2 Discrete Wavelet Transform

Discrete wavelet transform (DWT) decomposes a signal into a set of orthogonal components describing the signal variation across the scale [6]. The orthogonal components are generated by dilations and translations of a prototype function ψ , called mother wavelet:

$$\psi_{jk}(t) = 2^{-j/2}\psi(t/2^{j} - k), \quad k, j \in Z \tag{1}$$

The above equation means that the mother function is dilated by integer j and translated by integer k. A signal f for each discrete coordinate t can be presented as a sum of an approximation plus J details at the Jth decomposed level:

$$f(t) = \sum_{k} a_{Jk}\phi_{Jk}(t) + \sum_{j=1}^{J}\sum_{k} d_{jk}\psi_{jk}(t) \tag{2}$$

where $\phi_{Jk}(t)$ is scaling function. The residual term corresponds to a coarse approximation of $f(t)$ at resolution J.

The estimation of d_{jk} and a_{Jk} can be achieved through an iterative algorithm for decomposition by using two complementary filters h_0 (low-pass) и h_1 (high-pass) [7]. This is illustrated in Fig. 1 for 1D DWT and 2D DWT.

The most popular form of conventional wavelet-based signal filtering, can be expressed by:

$$\hat{d}_{jk} = d_{jk} \cdot h_{jk} \tag{3}$$

where filter h_{jk} describes "hard" or "soft" threshold filtering with a threshold τ_j, known also as wavelet shrinkage [8]:

$$h_{jk}^{(hard)} = \begin{cases} 1, & \text{if } |d_{jk}| \geq \tau_j \\ 0, & \text{if } |d_{jk}| < \tau_j \end{cases}, \quad \text{or}$$

$$h_{jk}^{(soft)} \begin{cases} 1 - \dfrac{\tau_j \operatorname{sgn}(d_{jk})}{d_{jk}}, & \text{if } |d_{jk}| \geq \tau_j \\ 0, & \text{if } |d_{jk}| < \tau_j \end{cases}. \tag{4}$$

ISSN 2229-5518

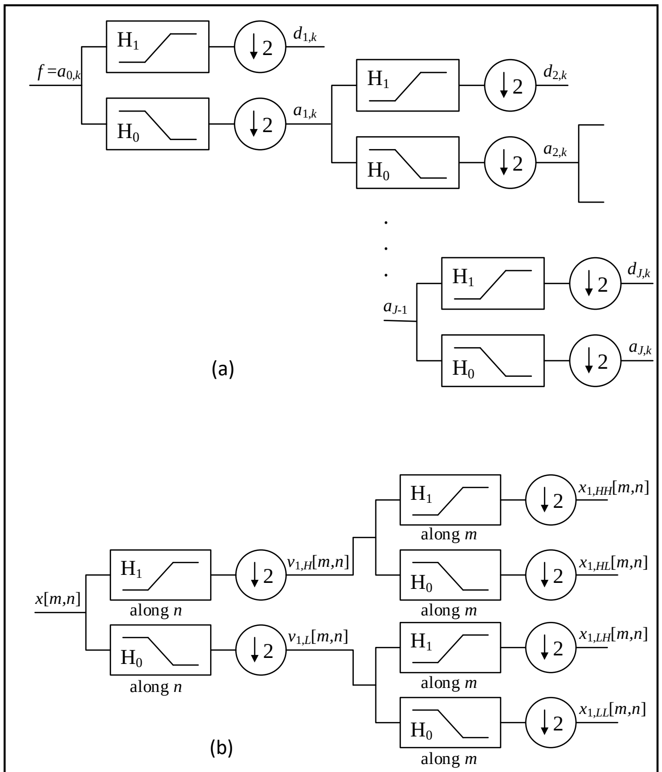

Fig. 2. Discrete wavelet transform tree for a) 1D DWT b) 2D DWT.

3 Image Pseudo-Hash

The main idea is to search a large database for images that correspond to particular image-query on the basis of a small piece of information calculated from the images themselves - pseudo-hash information [3]. By comparing the similarity between the images pseudohashes stored in the database and a query-image pseudo-hash, candidate-images can be selected to be considered visually if some of them correspond to the query-image.

This section presents an algorithm for creating pseudo-hash from an image by using wavelet transform. The wavelet transform is suitable since it tends to concentrate the energy of an image into a small number of coefficients, while a large number of coefficients have small energy. By applying a hard threshold given with (4) the most important wavelet coefficients can be selected.

In order to calculate the pseudo-hash information for an RGB image, the image first is converted into YCbCr colour space, where Y is the luminance (intensity) component and Cb (blue chrominance) and Cr (red chrominance) are the blue-difference and red-difference chroma components, respectively. The wavelet transform is applied over the Y component and the wavelet detail coefficients are filtered in order to keep only the most important coefficients. The mean and standard deviation of x and y coordinates of the coefficients positions compose the pseudo-hash information for the image. Hence, an image pseudo-hash is consisted from four values (mean of the x coordinates, mean of the y coordinates, standard deviation of the x coordinates, standard deviation of the y coordinates) calculated from the postions of non-zero coefficients:

pseudohash[*meanxcoord*, *meanycoord*, *stdevxcoord*, *stdevycoord*].

A database that keeps images information can contain a few relations with their schemas given in Fig. 2. The relation *tbl_image* contains information like *name*, *description* and *location* of the images (the images are picture files in the file system). In this relation, the primary key is the attribute *id_image*. The relation *tbl_coeff* contains positions (x *and y* coordinates) of the representative wavelet coefficients of the images. The primary key is the attribute *id*, while *id_image* is a foreign key that takes its values from the primary key from the relation *tbl_image*. The relation *tbl_query* contains positions (x and y coordinates) of the representative wavelet coefficients of the query image. The pseudo-hash information for all the images in the tables *tbl_coeff* and *tbl_query* can be calculated by running simple SQL queries.

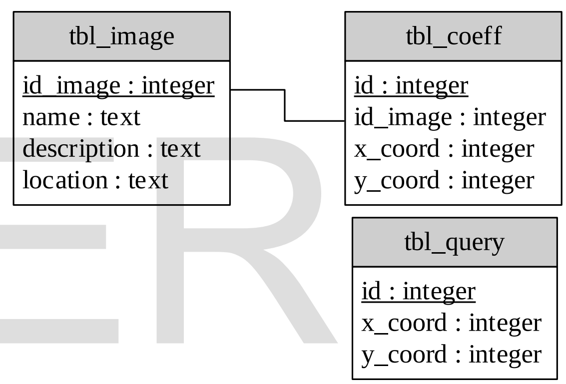

Fig. 2. Relations schemas in the images database.

A list of images candidates can be obtained by processing the images wavelet coefficients or images pseudo-hash information. Comparing of the positions of wavelet coefficients stored in the tables *tbl_query* and *tbl_coeff* can give a satisfactory result. The image with a maximum number of matching points (positions of the most important wavelet coefficients) is likely the image that is looked for.

Also, a list of candidates can be obtained by comparing the distances M_1 between the positions of the wavelet coefficients of the query image with the positions of the wavelet coefficients of the images in the database:

$$M_1 = \sum_{i,j} \frac{1}{\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + \alpha}} \tag{5}$$

where (x_i, y_i) are coordinates of wavelet coefficient of images stored in the database, while (x_j, y_j) are coordinates of wavelet coefficient of the query image. The parameter α is used to avoid obtaining big values or infinity for M_1, when the query image is the same or very similar with some images in the database. Images with bigger values for the similarity measure M_1 are candidates to correspond to the query image.

Next, a list of candidates can be obtained by processing the distance M_2 between pseudo-hash of the query image with the pseudo-hashes of the images in the database:

$$M_2 = \sum_i \frac{1}{\sqrt{\left(p.h.(i) - p.h.q(i)\right)^2 + \alpha}}, \quad (6)$$

where $p.h.(i)$ and $p.h.q(i)$ are pseudo-hash elements for the images in the database and the query-image, respectively:

p.h.(1) – mean of the x coordinates,
p.h.(2) – mean of the y coordinates,
p.h.(3) – standard deviation of the x coordinates,
p.h.(4) – standard deviation of the y coordinates.

The proposed algorithm is given in Fig. 3.

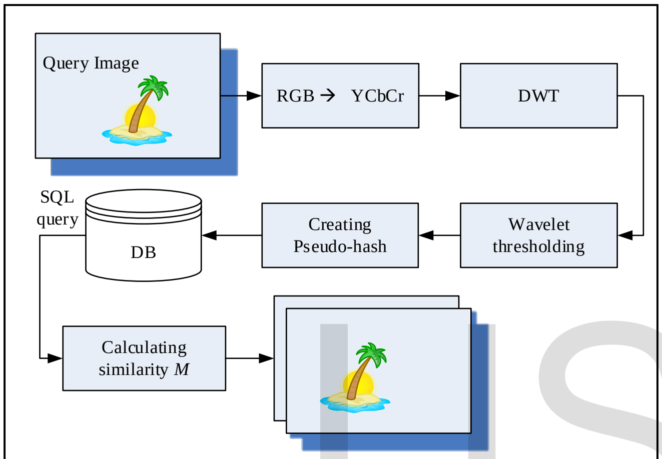

Fig. 3. Block diagram of the proposed algorithm.

4 Expiremental Results

This Section presents the results obtained by experiments performed over a collection of 1001 similar images of people, animals, landscapes, objects, etc. All the images have their own id: from id = 0 to id = 1000. Fig. 4 shows a few of these images. Pseudohash data for all the images are calculated by the proposed algorithm from Section 3 and stored in a Microsoft Access database which schema is given in Fig. 2. The database does not contain the images themselves – the images are picture files stored in the file system.

For calculating the pseudohash data for the images from the collection, the images are converted in YCBCR colour space and the haar wavelet transform in three levels is applied over their Y components. The horizontal, vertical and diagonal detail coefficients from the third level are summed up and the most important 5% coefficients are kept in the table *tbl_coeff*. The pseudohash data are calculated by running a simple SQL statement.

Fig. 4. Part of a collection of 1001 images.

Next, the images shown in Fig. 5 are used as query-images (id = 1, 1503 and 1089, respectively). All they have same resolution 256x384. The sparse matrices obtained from the most important wavelet coefficients of the three images after applying the haar wavelet transform in three levels are stored in table *tbl_query* and they are also shown in Fig. 5. Their resolutions are 48x32, which means that only a few coefficients were taken into consideration for the calculation of the pseudohash. The first query image (id = 1) is already contained in the collection and accordingly its peseudohash is already stored in the database. The second query image (id = 1503) is obtained from the image with id = 501 when some parts from the original image are erased, so the query image (id = 1503) contains missing (black) parts. The third query image (id = 1089) is obtained by filtering an original image from the collection (id = 87). By running a simple SQL statement, the pseudohash data are calculated for the three query images.

The next step is searching the database for candidates images that match the three query images. First, this is carried out by comparing the filtrated wavelet coefficients of the images stored in the database (*tbl_coeff*) and the filtrated wavelet coefficients of the query images (*tbl_query*). The best candidates are the images that have the most overlapping (matching) wavelet coefficients (points). The results are shown in Table 1. For the query-image with id = 1, the candidate image with the most matching points (77) is the image with id = 1 (the same image). For the query-image with id = 1503, the candidate image with the most matching points (56) is the image with id = 501, and for the query-image with id = 1089, the candidate image with the most matching points (71) is the image with id = 87.

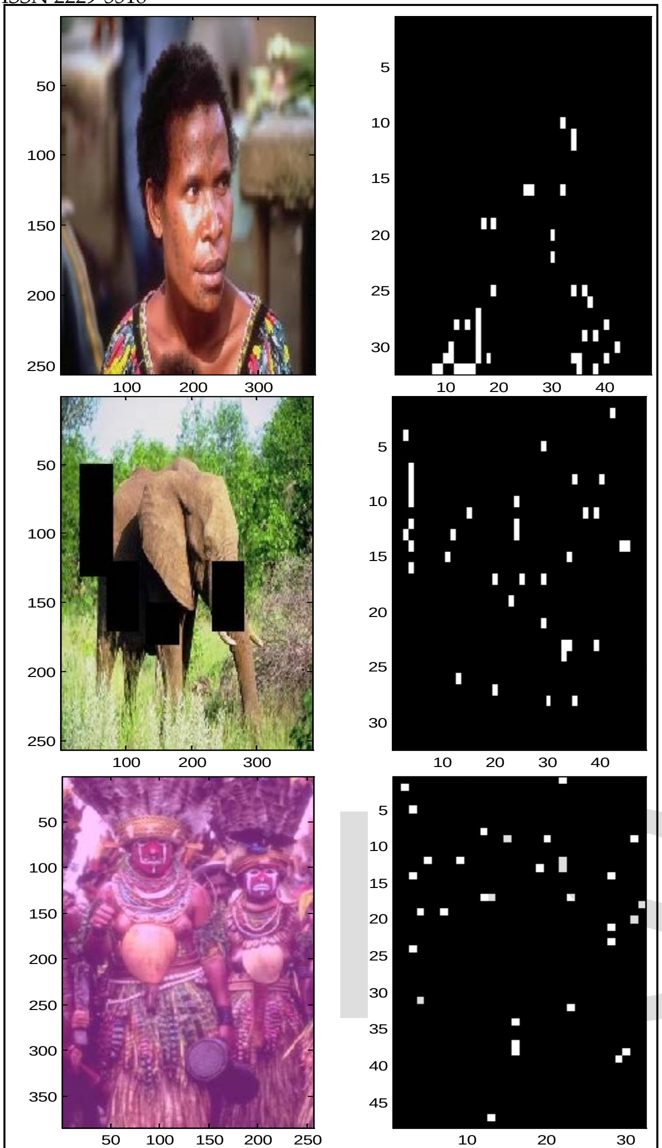

Fig. 5. Query-images (with id = 1, 1503, 1089) and their most important wavelet coefficients at level 3.

Next, the searching the database for candidate images is carried out by calculating the measures M_1 and M_2, according to (5) and (6), respectively, and the results are shown also in Table 1. The parameter α in (5) has value 0.1, while in (6) it is 0.01. The measures M_1 and M_2 can be calculated by running simple SQL commands over the tables and queries shown in Fig. 2.

It can be seen again that the images with id = 1, 87 and 501 have the highest similarity with the query images.

5 Conclusion

In this paper a CBIR algorithm based on the discrete wavelet transform is presented. The most important wavelet coefficients for images are used to create images pseudo-hash information. A database is searched for images that correspond to a query image by comparing respective images pseudo-hashes. The experiments validate the algorithm, showing that it works properly and delivers the expected results.

TABLE 1
Results of Searching for Images in the Database

Image id	Match. points	Image id	M_1	Image id	M_2
Query image = 1 (there is image with id = 1 in the database)					
1	77	1	996,8978	1	100
544	9	92	334,0799	63	0,3500
686	9	544	328,2449	580	0,2479
126	9	126	326,6215	30	0,2284
252	8	202	326,0941	49	0,2202
797	8	290	325,7217	290	0,2012
202	7	686	324,0086	744	0,1917
763	7	738	311,7879	205	0,1914
853	7	206	304,5836	65	0,1869
161	6	937	303,3439	34	0,1868
Query image = 1503 (filtered variant of image with id = 501)					
501	56	501	774,1331	501	0,6247
719	11	565	396,2749	251	0,4932
386	9	719	385,9561	867	0,4655
415	8	490	375,9287	222	0,4556
434	8	497	360,9072	534	0,4299
497	8	415	359,0868	830	0,3662
490	8	405	354,2778	761	0,3639
440	7	523	350,8746	432	0,3555
862	7	732	349,0866	549	0,3305
Query image = 1089 (variant of image with id = 87)					
87	71	87	908,4499	87	0,3023
86	6	808	297,3539	54	0,2056
13	5	13	290,356	4	0,1852
808	5	271	286,6985	923	0,1569
47	5	943	275,2746	555	0,1468
215	5	103	274,2945	340	0,1464
916	5	850	273,3468	11	0,1441
109	5	842	271,9997	5	0,1433
0	4	722	268,5459	255	0,1365
941	4	86	266,4457	370	0,1350

Our future work will be focused on extending the algorithm for searching when the image-query is a blurred or rotated image, a low-resolution image from a scanner or video camera or it is manually drawn picture.

References

[1] Vetterli M, Kovacevic J. *Wavelets and subband coding*. Prentice-Hall, 1995.

[2] Liu Y, Zhang D, Lu G, Ma WY. “A survey of content-based image retrieval with high-level semantics”, Pattern recognition 40 no. 1,pp. 262-82, Jan. 2007.

[3] Barber, R., W. Equitz, W. Flickner, W. Niblack, D. Petkovic, P. Yanker. "Efficient query by image content for very large image databases", Compcon Spring'93, Digest of Papers, pp. 17-19. IEEE, 1993.

[4] C. E. Jacobs, A. Finkelstein, D. H. Salesin, "Fast multiresolution image querying", SIGGRAPH '95 Proceedings of the 22nd annual conference on Computer graphics and interactive techniques.

[5] Wang, J.Z., Wiederhold, G., Firschein, O. and Wei, S.X., "Content-based image indexing and searching using Daubechies' wavelets", International Journal on Digital Libraries 1.4,pp. 311-328, 1998.

[6] G. Strang, T. Nguyen, *Wavelets and Filter Banks*. Wellesley-Cambrige

Press, 1996.

[7] P.P.Vaidyanathan, *Multirate Systems and Filter Banks*. Prentice – Hall, 1992.

[8] D. L. Donoho, "Wavelet Thresholding and W.V.D.: A 10-minute Tour", Int. Conf. on Wavelets and Applications, Toulouse, France, June 1992.