

# Content based Image Retrieval in the Compressed Domain

*by* Suhendro Irianto

---

**Submission date:** 20-Feb-2020 07:40PM (UTC+0800)

**Submission ID:** 1260768036

**File name:** content\_based\_image\_retrieval\_suhendro\_1.pdf (705.17K)

**Word count:** 3857

**Character count:** 20456

# Content based Image Retrieval in the Compressed Domain

Suhendro Y. Irianto  
Department of Informatics,  
Darmajaya Informatics and Business Institute  
Jalan Z.A. Pagar Alam No.93A,  
Bandar Lampung, Indonesia

## ABSTRACT

Given an image with  $N$  blocks of  $8 \times 8$  pixels, we construct an indexing key by overlapping the  $N$  blocks into one combinational block and each block acting as one single plane inside the combinational block. Specific construction of each element inside the indexing key can have a range of alternatives based on such a common platform. These include: (i) average DCT value (ii) energy distributed in DCT domain to construct the indexing key, and (iii) DCT coefficients that can be polarized via exploiting their directional properties, and thus can be processed to construct an energy magnitude to highlight the texture of the input image. In this way, the dimension of the indexing key can be significantly reduced. In this paper we represent DCT descriptors as tools of generating indexing key in compress domain.

## General Terms

Image Retrieval

## Keywords

Compressed domain, DCT domain, CBIR

## 1. INTRODUCTION

In this chapter, JPEG compression technical properties are exploited to propose simple content descriptors for image indexing and retrieval, which are particularly suitable for JPEG compressed images. Given an image divided into  $N$  blocks of  $8 \times 8$  pixels, an indexing key is constructed in the spirit of the bit-plane encoding of images, where each pixel is divided into a bit sequence and each bit of all pixels represents a bit plane. As a result, the indexing key has 64 elements and each element corresponds to one location along the zig-zag route of the JPEG compression standard. The specific construction of each element inside the indexing key has three alternatives and thus three content descriptors can be constructed. These include an energy-based content descriptor (average of DCT coefficients), a binary texture based descriptor, and an orientation based descriptor (horizontal and vertical). The research was also extended to calculate the effectiveness of image retrieval in DCT and by using Adobe Photo Album (APA) as comparison.

Because most images appearing on the World Wide Web are compressed at source, techniques for handling these compressed images directly in their compressed domain has become increasingly important for a range of applications and digital enterprise sectors. As a result, a series of international image and video compression standards have now been developed by International organizations such as JPEG [1] and MPEG [2]. As a matter of fact, the JPEG compression standard is the most popular technique and accounts for more than 95 % of all images on the Web [3]. Since image data size

is significantly larger than traditional text-based information, data compression plays an important role in reducing it to a more manageable one without damaging the quality of images for information delivery [4]. Accordingly, to bridge the gap between the compressed domain and the pixel domain, where the majority of existing image processing algorithms are developed, researchers worldwide have started to develop image analysis and content feature extraction algorithms acting directly on the compressed domain data [5,6,7]

By examining the JPEG compression standard, it can be seen that the core of its compression techniques is the  $8 \times 8$  block-based discrete cosine transform (DCT) [8]. The DCT approximates the principal component analysis (PCA) and decorrelates the pixel data along vertical and horizontal directions. For the existing image processing algorithms, working in the pixel domain, decompression is inevitable, and significant computing cost has to be incurred applying the inverse DCT transform. In compressed domain approaches, direct extraction of content features is conducted by processing the DCT coefficients and thus only partial decompression is required. An overview of such a compressed domain approach is shown in Figure 1. On the other hand, research on image indexing and retrieval in the pixel domain has been actively done over past decades, and many systems have been developed and widely reported in the literatures, such systems include [9,10,11,12,13,14]

The author has carried out extensive work using descriptors proposed in section 2 and the results show competitive performances compared with existing techniques (DCT coefficients based image retrieval). The proposed descriptors provide a number of potential advantages and features, which can be briefly summarized as: (i). low complexity and low computing cost; (ii). high processing speed, particularly suitable for texture classification or clustering inside large image databases; (iii). easy to implement inside the JPEG compressed domain, and thus providing the additional advantage that compressed images can be directly retrieved without full decompression.

## 2. DCT BASED CONTENT

In this section, the author presents his recent work on content descriptor construction directly from DCT domain which saves computing cost on decompression whilst the performance of content-based image indexing and retrieval remains competitive. There are three proposed descriptors in this work, which are explained in sections 2.1 to 2.3, namely: binary texture, energy-based content, and orientation based content descriptors. In addition to these three methods, image retrieval was also investigated for DCT coefficients based and Adobe Photo Album applications.

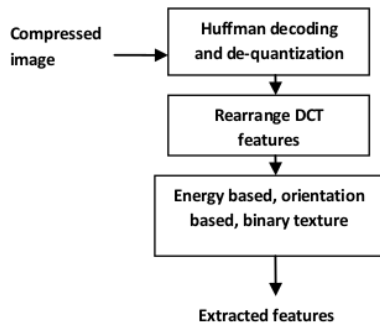


Fig.1. Diagram of DCT feature description extraction

In this paper presents works on energy based, oriented based, and binary texture images. More over, the work also used Adobe Photo Album and full DCT coefficients descriptor.

### 2.1. Binary Texture Descriptor

By taking each block as a pixel, the input image size is reduced to  $W/8 \times H/8$ , where  $W \times H$  is the original size of the input image. In this way, a sequence of 64 sub-images of size  $W/8 \times H/8$  is produced. The first sub-image consists of the DC coefficients and thus is referred to as the DC sub-image. The other 63 sub-images are referred to as AC sub-images since their elements or pixel data are AC coefficients. The texture of each pixel neighborhood can be described by a binary sequence, of length determined by the number of neighbors. This is illustrated in Figure 2.

The block has two binary variables, 0 or 1. It allows communicating with 8 neighbor DCT coefficients: on the left, right, top, and bottom. A simple algorithm is applied to transform DCT values into a binary texture descriptor. The algorithm can be explained as follow: *i*) select a coefficient which is located in the centre of  $3 \times 3$  blocks,  $X_i$  as central value; *ii*) compare every coefficient in the neighborhood; *iii*) if the coefficient is greater or equal to the centre coefficient ( $X_i$ ), assigned 1, otherwise 0.

$x_1$	$x_2$	$x_3$
$x_4$	$x_i$	$x_5$
$x_6$	$x_7$	$x_8$

Fig.2. The illustration of neighbourhood coefficients

An 8 bit binary string for the 8 neighbours shown in Figure 2 is constructed by comparing, in a fixed order, the neighboring elements with the central one, and defining each bit by:

$$Y_i = \begin{cases} 1 & \text{if } x \geq x_i \\ 0 & \text{otherwise} \end{cases}$$

where  $x_i$  corresponds to the  $i^{\text{th}}$  element of the neighbourhood. The proposed texture descriptor is constructed from any type of compressed image, split into  $8 \times 8$  sub blocks with DCT coefficients for each sub image computed. Based on these binary texture coefficients, the proposed texture descriptor is constructed as illustrated in Figure 3.

1	1	0	0	1	1	1	1	0
0	1	1	1	1	1	1	1	0
1	0	1	0	1	0	1	1	0
1	1	1	1	1	1	1	1	0
1	1	0	1	1	1	1	1	0
1	0	1	1	1	1	1	1	0
1	1	1	1	1	1	1	1	0
1	1	1	1	1	1	1	1	0
0	0	0	0	0	0	0	0	0

Fig.3. Binary texture – quantized DCT coefficients conversions of its neighbourhood by padding the last row and column with “0”.

Padding with an extra column and row in a block, to create a block that has a greater number of DCT coefficients than it had originally. This padding is added to construct a  $9 \times 9$  block instead of  $8 \times 8$ , this make the left, right, top, and bottom coefficients of the centre (big “1”) become symmetric.

This texture examination of all elements inside a sub-image was used to produce a histogram with normalized counts defining the descriptor array  $H$  of the equation below to characterize the binary texture of the sub-image.

$$H_i(\text{binary texture}) = \sum_{i=0}^N \frac{Y_i}{N}$$

Here  $Y_i$  is the count of binary texture value  $i$ , total number of elements across the image, where their binary texture bit stream is the same and  $N$  is the total number of blocks inside the input image.

### 2.2 Orientation based content descriptor

Firstly work was carried out by defining the two edges and spatial edge model and then these two edge features were used to obtain edges defined in the edge model. The distinct horizontal and vertical edge features can be formed from the two-dimensional DCT of a block. Further examination of the DCT transform in the JPEG compression standard indicates that horizontal and vertical coefficients, as shown in Figure 4, can be exploited to detect horizontal and vertical edges from DCT blocks. These edges in an  $8 \times 8$  block can be determined by two edge feature sets:

$$\begin{aligned} \text{Horizontal features} & : H = \{H_i; i=0, 1, 2, \dots, 7\} \\ \text{Vertical features} & : V = \{V_j; j=0, 1, 2, \dots, 7\} \end{aligned}$$

where  $H_i$  and  $V_j$  correspond to the DCT coefficients  $F_{u,v}$ , for  $u, v = 0, 1, 2, \dots, 7$ , in equation (3-6), which describes the 2-dimensional DCT:

$$F_{0,0} = \frac{1}{\sqrt{MN}} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} x_{i,j}$$

$$F_{u,v} = \frac{2}{\sqrt{MN}} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} x_{i,j} \cos \frac{(2i+1)u\pi}{2M} \cos \frac{(2j+1)v\pi}{2N}$$

where  $u = 1, 2, \dots, M-1$ , and  $v = 1, 2, \dots, N-1$ . For an 8x8 block,  $M = N = 8$ . Equation  $F(0,0)$  and  $F(u,v)$  describe DC and AC coefficients of DCT, respectively. Each coefficient of the two-dimensional DCT coefficients  $F_{u,v}$  corresponds to a sinusoidal intensity distribution across its respective image.

The two equations mentioned used to analyze the corresponding DCT coefficients for these models. Furthermore only the horizontal ( $H_i$ ) and vertical ( $V_i$ ) features corresponding to horizontal and vertical edges are utilized. Finally when all edge features have been examined on the basis of overall input image, a histogram can be constructed, where each element simply counts the coefficients of horizontal feature ( $H_i$ ) and vertical ( $V_i$ ) coefficients of each block.

DC	V1	V2	V3	V4	V5	V6	V7
H1							
H2							
H3							
H4							
H5							
H6							
H7							

Fig.4. The horizontal and vertical edge (orientation based) features of DCT coefficients in an 8 x 8 block.

### 2.3 Energy-Based Content

Given an image with  $N$  blocks of 8 x 8 pixels, an indexing key is constructed by combining the  $N$  blocks into one combinational block with each block acting as one single plane inside the combinational block. This is prompted by following the similar spirit of bit-plane encoding in the area of data compression, where each pixel is divided into a bit sequence and each bit of all pixels constructs a bit plane. As a result, the indexing key has 64 elements and each element corresponds to one location along the zig-zag route of JPEG compression. The specific construction of each element inside the indexing key is defined as:

$$h_i = \frac{\sum_{t=1}^N (DCT_t)^2}{N}$$

where,  $h_i$  is the  $i$ <sup>th</sup> element of the feature vector, and the indexing key is written as:

$$H_i = \{h_0, h_1, \dots, h_{63}\}$$

### 3. EVALUATION AND ANALYSIS

To evaluate the proposed content descriptors, work was carried out to index and retrieve images for a database of more than 10,000 images. To provide the ground truth for measuring performance, the image database was prepared from ten image categories which includes *bike, building, car, animal, flower, model, mountains, sky, sunset* and *texture*. A sample of these image categories or classes is shown in Figure 4. In the database, each category is represented by roughly 10,000 images. Ten images from each category were randomly picked from the database to formulate a query data set, of 100 query images for testing the proposed algorithms.



Fig.5. Sample of the ten image categories in the database

To measure the similarity between the query image and the images inside the database, Euclidean distance is used and defined as follows.

$$d(Q, I) = \frac{\sum_i (Q_i - I_i)^2}{M}$$

Here  $Q$  and  $I$  stand for the query image and database image descriptor feature vector respectively and  $M$  stands for the total number of elements inside the content descriptor.

For each query image, 20 images were retrieved from the database and values of precision and recall calculated. Precision is the number of relevant images retrieved divided by the total number of images retrieved. Recall is the number of relevant images retrieved divided by the total number of relevant images in the category or collection. The experimental results for all ten categories are summarized in Table 1. From these results, it can be seen that the overall performance of the proposed descriptors remains competitive compared to the benchmark, DCT coefficient based retrieval. There are also a significant number of classes, where the proposed descriptors outperform of the DCT coefficient based ones. Adobe Photo Album is tools for managing photos, in this work the tool was used retrieve the images from the database. The retrieval process is based on colour similarity.

Table 1. Precision and recall of the proposed algorithms

Method	Flower	Mount	Sky	Building	Car	Model	Bike	Sunset	Cat	Texture
APR *	0.26	0.31	0.24	0.40	0.55	0.62	0.54	0.38	0.22	0.92
**	0.05	0.07	0.08	0.02	0.09	0.10	0.23	0.11	0.06	0.03
Energy *	0.82	0.72	0.60	0.80	0.43	0.95	0.40	0.35	0.65	0.90
**	0.02	0.05	0.10	0.09	0.25	0.03	0.19	0.04	0.08	0.07
Orientation *	0.29	0.20	0.77	0.69	0.16	0.63	0.65	0.63	0.98	0.88
**	0.04	0.11	0.06	0.05	0.16	0.09	0.08	0.18	0.07	0.19
Binary *	0.50	0.83	0.52	0.60	0.68	1.00	0.72	0.57	1.00	0.50
**	0.05	0.06	0.08	0.07	0.16	0.09	0.14	0.17	0.22	0.19
DCT *	0.37	0.41	0.37	0.39	0.34	0.58	0.64	0.48	0.55	0.69
**	0.09	0.05	0.5	0.10	0.01	0.11	0.04	0.07	0.03	0.06

\*: precision      \*\*: recall

It is not useful to retrieve all relevant images in a database of thousands of images. In the experiment, the evaluation was carried out for the top 20 images ranked in terms of the similarity measures by using *precision* and *recall* parameters. It can be seen in Figures 5,6,7,8, and 9 that the best precision achieved by each method was 100% for binary texture, 98% for orientation based, 92 % for Adobe Photo Album (APA), 95% for energy based, and 69% for DCT coefficients. Moreover, for nine of the ten classes, the *texture* descriptor achieved the highest precision.

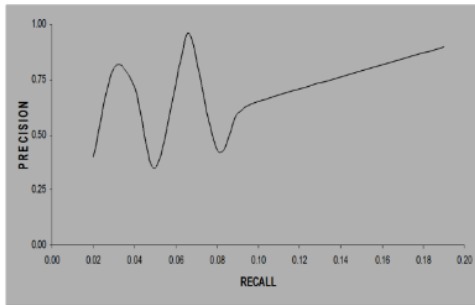


Fig.6. The precision and recall of energy based (average DCT) descriptor

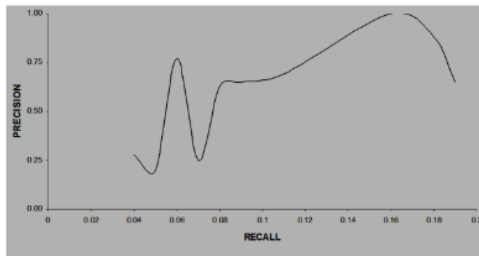


Fig.7. Precision and recall of distribution energy based

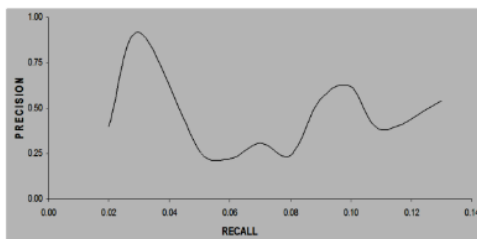


Fig.8. Precision and recall of image retrieval by Adobe Photo Album (APA)

The experimental results also show that binary texture descriptor performs the highest in average precision at 73% compared to the other descriptors. Figure 8 demonstrates the average precision among other descriptors that are 48%, 66%, 59%, and 44% for DCT coefficients based, energy based descriptor, orientation based descriptor and APA respectively. For all ten classes in the database, binary texture descriptor achieves an average precision of 73% and 9 % for recall.

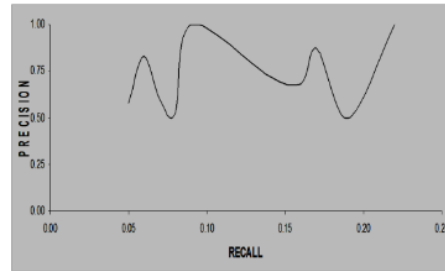


Fig.9. The precision and recall of Binary texture method

Further analysis of the experimental results can be also revealed that, for some categories or classes, a 98% precision retrieval was achieved. These include *cat* for both binary texture descriptor and edge-based or orientation-based descriptor. The experimental result also shows that binary texture descriptor shows the highest in average precision at 73% compared to other descriptors.

As the five descriptors are applied to the image database, the lowest precision achieved by the binary texture descriptor is 50% for *texture* while the lowest precision for the other descriptors are 34%, 35%, 20%, and 22% for DCT coefficient based, energy based descriptor, orientation based descriptor, and APA, respectively. As illustrated in Figure 3.11, the binary texture algorithm achieves the highest average in precision (73%) among the algorithms, while APA shows the lowest average of precision, 44%. An interesting occurrence is that the *texture* class shows the highest average of relevant image retrieved for all the descriptors except for orientation based descriptor that shows *car* as the highest precision.

For this reason, it can be stated that binary descriptor algorithm represents a preferred method over the other four descriptors. The binary texture descriptor works extremely well for *texture*, *sunset*, and *mountain* (greater than 80%), but not quite well for *cat* images. In calculation of ten classes, binary texture descriptor also shows that all its precisions are greater than 50% and almost faultless for *texture* which is 98%. *Texture* images have been retrieved with an average of 98% by using binary texture descriptor. Furthermore, binary texture achieves the best retrieval among all the descriptors with averages of 73% and 9% for precision and recall, respectively.

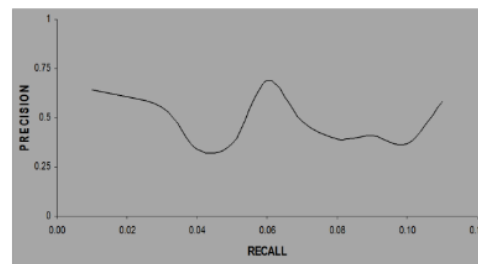


Fig. 10. Precision and recall by using DCT coefficients based method

Compared to the existing method (DCT coefficients based), the proposed descriptors are superior in terms of numbers of images retrieved, except for the orientation based descriptor. For that reason, the proposed methods can be used as alternative techniques to improve the number of relevant images retrieved by image retrieval systems, particularly in the compressed domain.

The methods have been carried out on a compressed images database to verify their performance in a JPEG standard stream line. Thus it would be expected that among the five descriptors used, the binary texture descriptor would have the largest number of images retrieved during query process. In addition to the binary texture descriptor, the energy based descriptor and the horizontal-vertical or orientation based descriptors show better performance over the existing method in term of average precision.

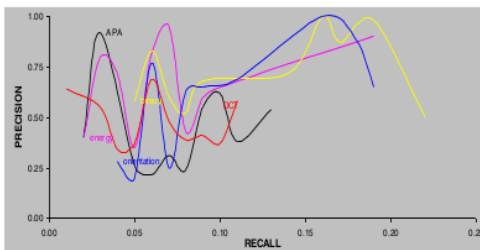


Fig.11. The precision and recall for three method descriptors, DCT based and Adobe Photo Album (APA)

Experimental results show that retrieval effectiveness of over ninety percent has been achieved by the proposed scheme. Although this new image searching method has substantial advantages over the existing approach, it is not perfect and its ultimate utility may depend on the size of the image database and the number of image classes being searched. In this paper, an image retrieval system based on various descriptors in compress domain was presented. It is presented as alternate methods developed for image indexing by using of DCT texture descriptors. We used a discrete cosines transforms to compose the image indexing from DCT coefficients properties to compute the energy features of the image.

The methods have been carried out for compressed images database to verify its performance in JPEG standard stream line. Thus it could be concluded that among five descriptors used, Binary Texture descriptor has the highest number of images retrieved during query process. In addition to Binary Texture descriptor, Average DCT and Horizontal-Vertical descriptors show better performance over the existing method in term of average precision.

It must therefore be recognized that an effective indexing method is introduced which expands the work in image indexing using of Euclidean distance algorithm. Our experiments verify that the propose methods can be used as substitute methods to achieve higher number of image retrieved by utilizing the description of DCT coefficients. The descriptors we have proposed are uncomplicated, involve only a small amount of data to be stored for each target image, and reasonably effective. It is also fairly easy to understand and implement in compress domain.

Experimental results show that retrieval efficiency of over ninety percent can be achieved by the proposed scheme. Although this searching method has substantial advantages over existing approach, it has not been perfect yet and its ultimate utility may depend on the size of the image database and number of image class being searched.

The retrieval efficiency has been shown better that the existing method, but more experiments should be done toward flawless image retrieval in the near future. We are setting up these methods on DC domain, and to special databases such as biomedical database and very large database from World Wide Web.

#### 4. CONCLUSION

In this paper, an image retrieval system based on the baseline JPEG international compression standard has been proposed, where three content descriptors are extracted. Extensive experiments showed that the proposed techniques achieved competitive performances in comparison with an existing DCT based method. The proposed methods have a significant advantage over pixel domain method by requiring only partial decompression. The proposed content descriptors are also suitable for real-time implementation and applications.

The main focus of this paper is presented content based image retrieval by employing a various descriptors. The energy based, edge-based and binary texture descriptors have been used in the DCT domain to improve precision and recall of image retrieval. Precision and recall have also been investigated for image retrieval based on DCT features and Adobe Photo Shop as comparisons for the three proposed descriptors.

#### 5. REFERENCES

- [1] Javed . M, P. Nagabhusan, and B.B. Chaudhuri. Direct Processing of Run-Length Compressed Document Image for Segmentation and Characterization of a Specified Block, International Journal of Computer Applications., Vol.83, No. 15, 2013, Pp.1-6.
- [2] Lew, H, Nicu Sebe , Chabane Djeraba ,And Ramesh Jain .Content-based Multimedia Information Retrieval.State of the Art and Challenges., ACM Transactions on Multimedia Computing, Communications, and Application, Vol.2, No.1, 2006. Pp.1-19.
- [3] Gregory K. Wallace, The JPEG still picture compression standard communications of the ACM, *Special issue on digital multimedia systems*, Vol.34, Issue 4, ACM Press, New York, NY, USA, April 1991, pp. 30-44.
- [4] Didier Le Ga, MPEG: a video compression standard for multimedia applications, *Communications of the ACM, Special issue on digital multimedia systems*, Vol. 34, Issue 4, ACM Press, New York, NY, USA April 1991. pp. 47 – 58.
- [5] R.Uma . FPGA Implementation of 2-D DCT for JPEG Image Compression. International Journal Of Advanced Engineering Sciences D .2011, Vol No. 7, Issue No. 1, pp. 001 – 009.
- [6] E. Feig and S. Winograd, Fast algorithms for the discrete cosine transform, *IEEE Transaction on Signal Processing*, Vol. 40, Sept.1992, pp.21-74.

- [7] Shivang Ghetia, Nagendra Gajjar, and Ruchi Gajjar. Implementation of 2D Discrete Cosine Transform Algorithm on GPU. *International Journal of Advance Research in Electrical, Electronics and Instrumentation Engineering*. Vol. 2, Issue 7, 2013. Pp. 2024-3030.
- [8] Huazhong Shu, Yuan Wang, Lotfi Senhadji, and Limin Luo, Direct computation of type-II discrete Hartley transform, *IEEE Signal Processing Letters* 05/2004; Vol. 14, No. 5, Pp. 329 – 332.
- [9] Shen, B, and Ishwar, S. Direct feature extraction from compressed images, SPIE vol. 2670, *Storage & Retrieval for Image and Video Databases IV*, 1996.
- [10] Vaishali A. Choudhary and Preeti Voditel, Extraction of Region of Interest in Compressed Domain, *International Journal of Computer and Information Technology*, Vol. 02 Issue 04, July 2013, Pp. 594-602. ISSN: 2279 0764
- [11] Reeve, R., Kubik, K., and Osberger, W. Texture characterization of compressed aerial images using DCT coefficients, *Proceeding SPIE, Storage Retrieval Image Video Databases V*, Vol. 30, Feb. 1997, pp. 398-407.
- [12] W.G. Kropatsch, M. Kampel, and A. Hanbury (Eds.): *A New Wavelet-Based Texture Descriptor for Image Retrieval*, CAIP 2007, LNCS 4673, pp. 895–902, 2007. Springer-Verlag Berlin Heidelberg.
- [13] McIntyre, A. R., and Heywood, M. I., Heywood, Exploring content-based image indexing techniques in the compressed domain, *Proceedings of the 2002 IEEE Canadian Conference on Electrical & Computer Engineering*, Canada, 2002, pp. 957 – 962.
- [14] Padmashree Desai, Jagadeesh Pujari, and Goudar R.H. Image Retrieval Using Wavelet Based Shape Features *Journal Of Information Systems And Communication* ISSN: 09768742 & e-ISSN: 0976 8750, Vol. 3, Issue 1, 2012, Pp. 162-166.

# Content based Image Retrieval in the Compressed Domain

---

## ORIGINALITY REPORT

---

15%

SIMILARITY INDEX

5%

INTERNET SOURCES

13%

PUBLICATIONS

5%

STUDENT PAPERS

---

## MATCH ALL SOURCES (ONLY SELECTED SOURCE PRINTED)

---

2%

★ M.H. Lee, S. Nepal, U. Srinivasan. "Edge-based semantic classification of sports video sequences", 2003 International Conference on Multimedia and Expo. ICME '03. Proceedings (Cat. No.03TH8698), 2003

Publication

---

Exclude quotes      On

Exclude matches      Off

Exclude bibliography      On



# Content based Image Retrieval in the Compressed Domain

---

## GRADEMARK REPORT

---

FINAL GRADE

**/0**

GENERAL COMMENTS

**Instructor**

---

PAGE 1

---

PAGE 2

---

PAGE 3

---

PAGE 4

---

PAGE 5

---

PAGE 6

---