Enhancing Performance of Image Retrieval Systems Using Dual Tree Complex Wavelet Transform and Support Vector Machines

Adeel Mumtaz, Syed Asif M. Gilani, Kamran Hameed and Tahir Jameel

Faculty of Computer Science and Engineering, Ghulam Ishaq Khan Institute of Engineering Sciences and Technology, Topi, Pakistan

This paper presents a novel image retrieval system (SVMBIR) based on dual tree complex wavelet transform (CWT) and support vector machines (SVM). We have shown how one can improve the performance of image retrieval systems by assuming two attributes. Firstly, images that a user needs through query image are similar to a group of images with the same conception. Secondly, there exists non-linear relationship between feature vectors of different images and it can be exploited very efficiently with the use of support vector machines. At the first level, for low level feature extraction, we have used dual tree complex wavelet transform because recently it has been proved to be one of the best for both texture and color-based features. At the second level, to extract semantic concepts, we grouped images of typical classes with the use of one against all support vector machines. We have also shown how one can use a correlation-based distance metric for comparison of SVM distance vectors. The experimental results on standard texture and color datasets show that the proposed approach has superior retrieval performance over the existing linear feature combining techniques.

Keywords: complex wavelet transform (CWT), support vector machine (SVM), content-based image retrieval, texture image retrieval

1. Introduction

Digitization has made a profound effect on our everyday life, from HDTV to digital camera. Global village concept is truly visualized in today's field of communication where boundaries of many independent technologies have been merged to a common technology. With the rapid development of computing hardware, digital acquisition of information has become a popular method in recent years. Every day, G-bytes of images are generated by both military and civilian equipment. Large sets of medical images, architectural and engineering designs, journalism and advertising, are worth mentioning. Consequently, how to make use of this huge amount of images effectively becomes a highly challenging problem [1]. Historically, the only way to search through these collections was text-based. Images were first annotated using text and then traditional Database Management Systems (DBMS) were used to retrieve relevant images when required.

There were two main problems with this approach; at first, the amount of labor involved in manually annotating these images and, secondly, the inherent complexity and richness of image contents made the annotation process difficult. For example, it was not easy to label two images containing similar objects, but with different meanings. In order to overcome these problems and to make the retrieval process automatic researchers from the field of computer vision proposed a new idea [2], which is known as content-based image retrieval (CBIR). CBIR is also known as query by image content (QBIC) and content-based visual information retrieval (CBVIR). These days, many commercial and research CBIR systems are available. IBM's QBIC [7] and UC Berkeley's Blobworld are well known. Detailed comparison of such systems can be found at [3]. These systems are being used mainly for the following application areas:

- Security – e.g. image comparison for corporate security and law enforcement investigations

- Intellectual Property – e.g. information retrieval in patent and trademark data (e.g. technical drawings, logos, photos)

- Brand protection – e.g. tracking proprietary visual data on the Internet

- Media archives – e.g. search, retrieve and manage visual assets (and associated metadata)

- Scientific imaging – e.g. retrieves and classifies images with respect to specific visual content (e.g. life sciences, healthcare, and biotech)

- Military (radar, aerial, satellite target recognition)

- Architectural and engineering design

- Fashion and interior design

- Journalism and advertising

- Medical diagnosis

- Geographical information and remote sensing systems

- Cultural heritage

- Education and training

- Home entertainment

- Web searching.

In content-based image retrieval, we automatically extract features from images and then compare images using these features. Images having similar features would have similar contents. Basic block diagram of a CBIR system is shown in Figure 1.
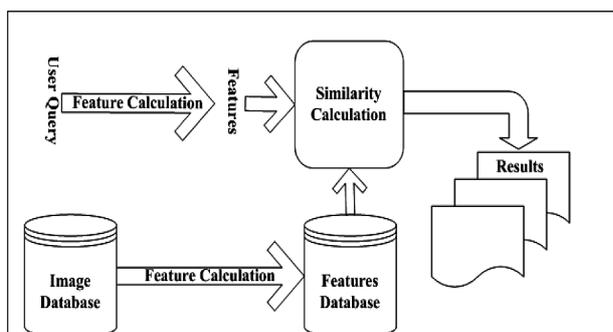
Figure 1. Typical CBIR system.

At first, features from all images in a database are extracted and stored into a feature database. This process is also known as indexing. When a user tries to search some images from the collection, he provides the system with a query image. Different options for query image are possible *i.e.*

- Image

- Rough Sketch

- Color or textural layout

- Verbal or semantic description

Features from query image are extracted by the same indexing mechanism. Then these query image features are matched with feature database using a similarity metric and, finally, similar images are retrieved.

A majority of indexing techniques are based on pixel domain features such as color [4], texture [5], and shape [6]. However, with recent advancements in image compression, compressed domain indexing techniques have gained popularity due to their less complexity. The reason for this is that we can get superior features in these domains. Some frequency domain techniques include wavelet domain features, Gabor transform and Fourier domain features. Comprehensive survey of existing CBIR techniques can be found in [1, 2].

Researchers have shown that texture is one of the most important features for CBIR. Texture refers to the visual patterns that have properties of homogeneity not resulting from presence of only one color or intensity. It is an innate property of virtually all surfaces, including clouds, trees, bricks, hairs, fabric, etc. It contains important information about the structural arrangement of surfaces and their relationship to the surrounding environment. There are many review papers on texture-based image retrieval. Manjunath and Ma [8] evaluated the texture image annotation by various wavelet transform representations and found that Gabor transform was the best among the tested candidates.

Kingsbury [13] proposed a new complex wavelet transform which allows fast computing Gabor-like wavelets. Peter and Kingsbury [14] in their paper have shown how one can use this new transform to speed up and enhance the image

retrieval process. Kokare et al. [15] have proposed even better extension of this work. Janney et al. [17] have shown how we can enhance the texture extraction capabilities of CWT for color image retrieval. They have shown that we can achieve almost the same precision for color image retrieval as well. These properties of CWT have motivated us to use it as feature extraction for our proposed system.

Mostly during the phase of comparison of features, linear metrics like Euclidean distance etc. were used. Recently, Han et al. [9] have shown how one can improve the performance of image retrieval systems by assuming non-linear relationship among feature vectors and grouping the images into similar classes. We have applied a similar idea for retrieving of texture images. We have used support vector machines for classification of images in the database.

The paper is organized as follows: Section 2 provides a brief introduction to dual tree complex wavelet transform and some of its applications. Section 3 provides an overview of support vector machines and how SVM can be used for classification. Section 4 describes proposed and implemented CBIR systems. Section 5 discusses the results of our technique in comparison with existing techniques. Section 6 gives the concluding remarks.

2. Dual Tree Complex Wavelet Transform

Wavelets are being used in many different areas. Applications of wavelets in signal denoising, image, audio and video compression, image smoothing and differential equation solution are active research topics [20]. Wavelets offer some advantages as a tool for image processing, such as the multiresolution formulation, which allows the reduction of computational complexity. This aspect is of crucial importance, since minimum retrieval time is required for an efficient browsing in large databases.

Kingsbury's [13] dual tree complex wavelet transform (CWT) is an enhancement to the discrete wavelet transform (DWT), with important additional properties. The main advantages, as compared to the DWT, are that the complex wavelets are approximately shift invariant (meaning that our texture features are likely to be more robust
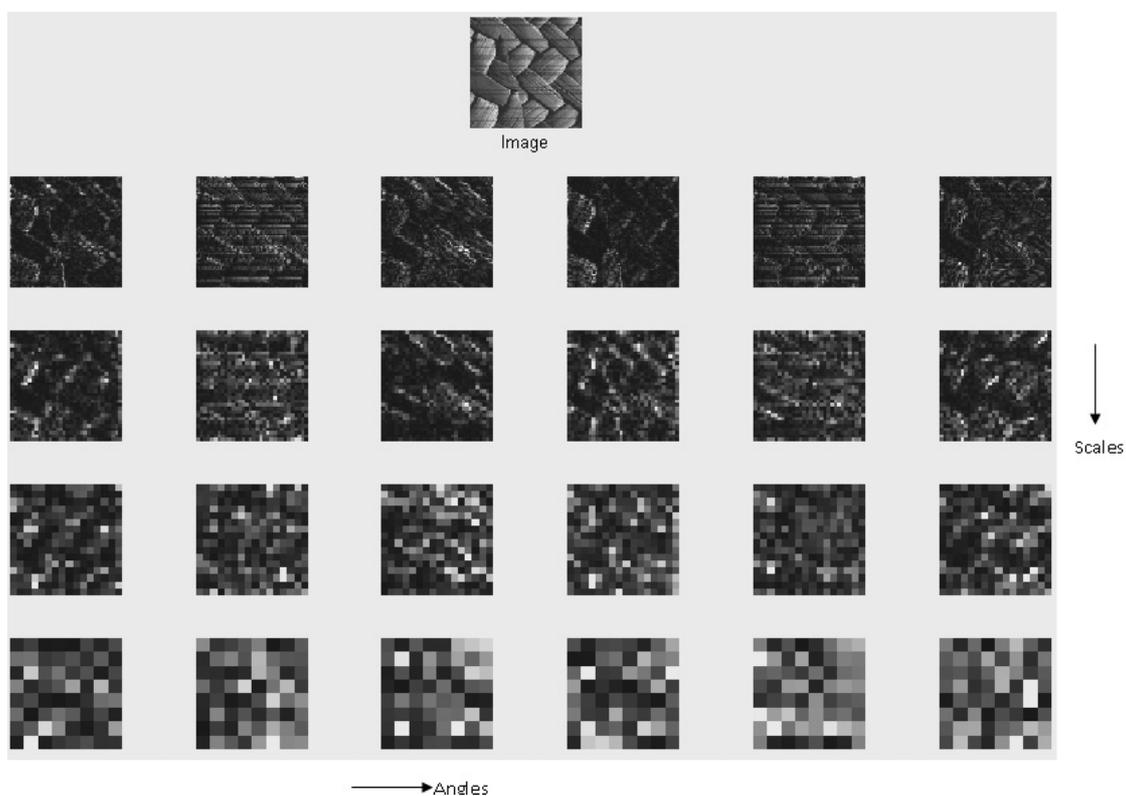
Figure 2. Four-scale CWT of a texture image.

to translations in the image) and that the complex wavelets have separate sub-bands for positive and negative orientations. Conventional separable real wavelets only have sub-bands for three different orientations at each level, and cannot distinguish between lines at $45°$ and $-45°$ respectively.

The complex wavelet transform attains these properties by replacing the tree structure of the conventional wavelet transform with a dual tree. At each scale one tree produces the real part of the complex wavelet coefficients, while the other one produces the imaginary parts. A complex-valued wavelet $\psi(t)$ can be obtained as:

$$\psi(t) = \psi_h(t) + j\psi_g(t) \qquad (1)$$

where $\psi_h(t)$ and $\psi_g(t)$ are both real valued wavelets.

CWT like Gabor transform have six orientations at each of four scales (any number of scales can be used, but the number of orientations is built into the method). The main advantage, as compared to the Gabor transform, is speed of computation. It has a redundancy of only 4 in 2-dimensions and so the post-processing stages (of calculating mean and standard deviations) are also faster as it has less redundancy than the Gabor wavelets.

Figure 2 shows magnitudes of CWT coefficients for a texture image. One can see more details related to orientation and scales. Each row represents one scale and the columns represent angles within that scale.

3. Support Vector Machines

There are many pattern matching and machine learning tools and techniques for clustering and classification of linearly separable and non-separable data. Support vector machine (SVM) is a relatively new classifier and it is based on strong foundations from the broad area of statistical learning theory [11]. Since its inception in the early 90's, it is being used in many application areas such as character recognition, image classification, bioinformatics, face detection, financial time series prediction etc.

SVM offers many advantages over other classification methods such as neural networks. Support vector machines have many advantages in comparsion with other classifiers:

- There are computationally very efficient as compared with other classifiers, especially neural nets.

- They work well, even with high dimensional data, a factor which limits many efficient classifiers.

- They can work well with less number of training data.

- They attempt to minimize test error rather than training error.

- They are very robust against noisy data (noise greatly degrades the performance of neural nets).

- The curse of dimensionality and over-fitting problems does not occur during classification.

Fundamentally, SVM is a binary classifier, but can be extended for multi-class problems as well. The task of binary classification can be represented as having, (X_i, Y_i) pairs of data. Where $X_i \exists X^p$, a p dimensional input space and $Y_i \exists [-1, 1]$ for both the output classes. SVM finds the linear classification function $g(x) = W.X + b$, which corresponds to a separating hyperplane $W.X + b = 0$, where W and b are slope and intersection. SVM, unlike other classifiers, finds the optimal hyperplane. Examples of optimal and non-optimal hyperplanes are shown in Figure 3.

a) Various hyper planes   b) Optimal hyperplane
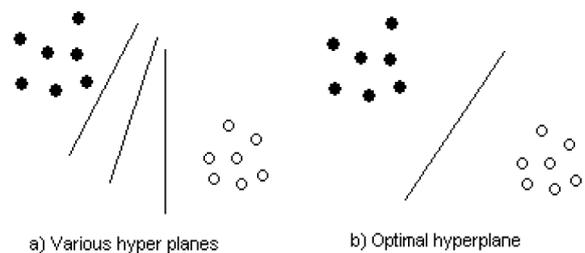
Figure 3. Optimal and non-optimal hyperplanes.

SVM usually incorporates kernel functions for mapping of non-linearly separable input space to a higher dimension linearly separable space. Many kernel functions exist such as radial bases functions (RBF), Gaussian, linear, sigmoid etc. Different options exist to extend SVM for multi-class cases, these include one against all, one

against one and all at once. Figure 4 shows how one against all SVM can be used for grouping of different classes inside an image database. Each support vector machine separates one class of images from the rest of the database, which is shown by non-linear boundaries. A more detailed tutorial on SVM can be found in [19], which is oriented to researchers working on pattern recognition.
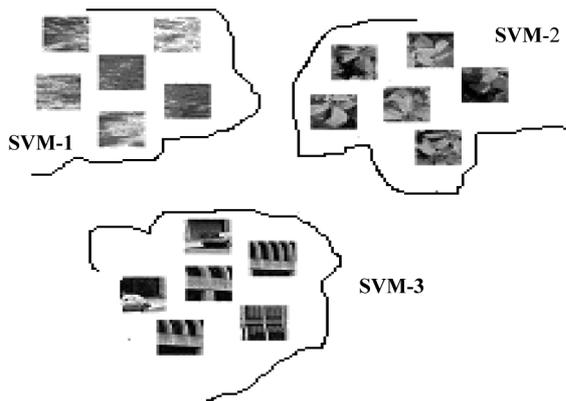
Figure 4. One against all classification showing three support vector machines.

4. CWT and SVM-based Image Retrieval System

In this section, we describe the structure of the proposed SVMBIR system in detail. Figure 5 shows the main components of the proposed system and the control flows among them.

The proposed system is based on SVM classifier. Our system is based on a similar assumption as Han et al.[9] that is images users need are often similar to a set of images with the same conception instead of one query image and the assumption that there is a nonlinear relationship between different features. The following steps show the detail of our proposed algorithm:

Step 1: Features are extracted from each image included in the image database using the following feature extraction process. First, we performed a four scale (six angles) CWT on an image. We got 24 real and 24 imaginary detailed sub-bands, and 2 real and 2 imaginary approximation sub-bands. By taking the magnitudes of
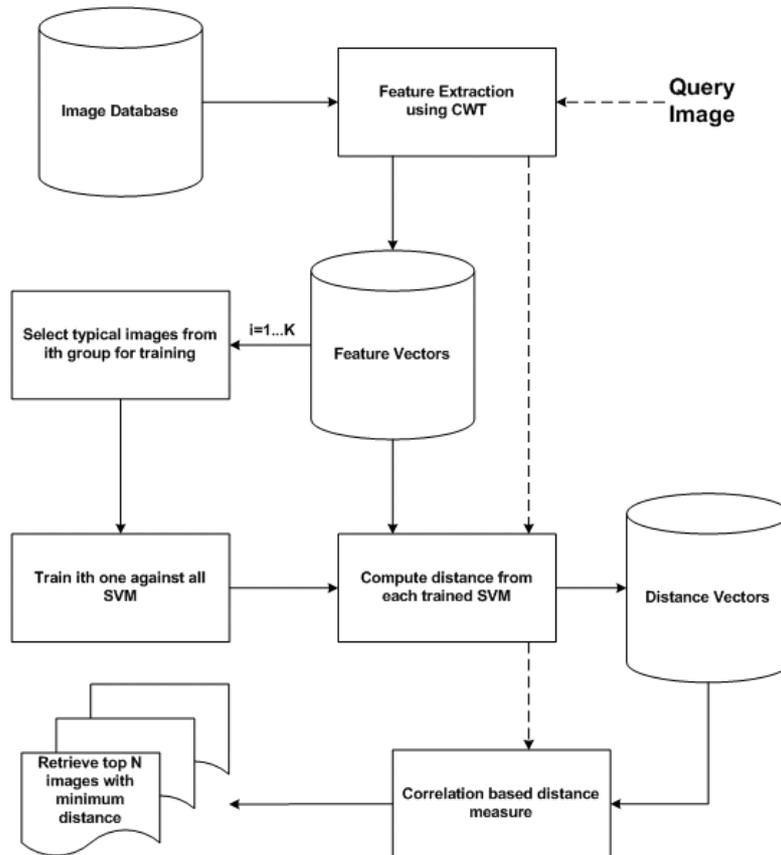
Figure 5. The structure of the proposed SVMBIR system.

corresponding real and imaginary coefficients of both approximation and detailed sub-bands, we got 26 sub-bands. To calculate the features we measured the mean and standard deviation of the magnitude of the transform coefficients in each of 26 sub-bands, in the same way as [14]. For color images we applied above process on each RGB color channels to get the feature vector. These features were then stored in feature database for later comparison.

Step 2: From each class of images included in the image database some typical images (K) were selected for training of support vector machine for that class. Selection of these training images can be done randomly or from a sequence. In our experiments we used first K images for training. We used one against all training method as it is the best when one needs good speed and reliable performance. This is done using *trainlssvm* function of LSSVM [16]. We used 'RBF' as kernel function for training of support vector machines. Optimal parameter selection is always a bottleneck of support vector machines. LSSVM provides a function *tunelssvm* which can be used for stimation of optimal parameters. We used grid search approach for searching optimal parameters. Our used and final parameter values were 100 and 20 for *gam* and *sig2* respectively.

Step 3: In this step, the distance of each image included in the database from each trained SVM is calculated. This is done using *simlssvm* function of LSSVM. Each of this distance is grouped in the form of distance vectors. This distance vector will store distance of each image from each support vector machine. Finally, we store all these distance vectors in distance vectors database.

Steps $1 - 3$ are done offline and after these steps our system is ready to process the user queries.

Step 4: When the user gives the system a query image, features from the query image are extracted. Using this feature vector of query image distance vector of query image is calculated.

Step 5: Query image distance vector is compared with all the distance vectors included in the distance vector database. By looking at the type of distance vectors generated we can easily find that following correlation based metric can be used for this comparison.

$$d_{rs} = 1 - \frac{(x_r - \bar{x}_r)(x_s - \bar{x}_s)'}{\sqrt{\left[(x_r - \bar{x}_r)(x_r - \bar{x}_r)'\right]\left[(x_s - \bar{x}_s)(x_s - \bar{x}_s)'\right]}}$$

(2)

This equation is one minus the correlation coefficient between vectors x_r and x_s. In our case, x_r is the query image distance vector and x_s is the distance vector of images included in the database. s varies from $1 \ldots N$, where N is the total number of images included in the image database. $\bar{x}_r$ and $\bar{x}_s$ are the means of the vectors x_r and x_s.

Step 6: Finally, the top Q images having minimum distance are retrieved and presented to the user.

5. Experimental Results

In this section we have shown some experimental results to evaluate the performance of our proposed system.

For texture images we have used the same dataset as was used by Peter and Kingsbury [14]. The texture data set used in the experiments contains 100 texture images from the Massachusetts Institute of Technology (MIT) VisTex [12] database. Each 512x512 image is divided into 16 smaller images of size 128x128 giving a total of 1600 texture images in the database. Each original image is treated as a single class and so we have 16 examples from each of our 100 classes. For color images we use the same dataset as was used by Janney et al. [17]. The color image data set used in the experiments contains 7200 color images from the Columbia University Image Library (COIL-100) [18]. These images are organized as having 100 objects photographed at 72 different angles giving total of 7200 images.

For performance comparison we have used the same technique as was used by Peter and Kingsbury [14]. They used plots similar to the precision recall graphs. For each of the 1600 images in the texture database we compute the distance to all of the other images and select the N nearest neighbors for each image. We then count how many of these belong to the correct class (up to a maximum of 15) and define the retrieval rate as this number divided by 15. This gives us a

retrieval rate for each of the 1600 images and we simply average these rates to give an overall retrieval rate. In our results, we plot the retrieval rate for a number of choices of N. A good set of features will give a retrieval rate that rapidly rises to 1. For each object present in coil-100 database we selected first instance out of 72 instances giving total of 100 query images and rest of the process is same.

Figure 6 shows the comparison of our technique with Peter and Kingsbury's technique. In this comparison we used only 25% images from each class for training of support vector machines. As shown in Figure 6, our technique performs better. We believe that the results can be significantly improved if we use more images for training. Figure 7 shows the comparison with 50% training images.
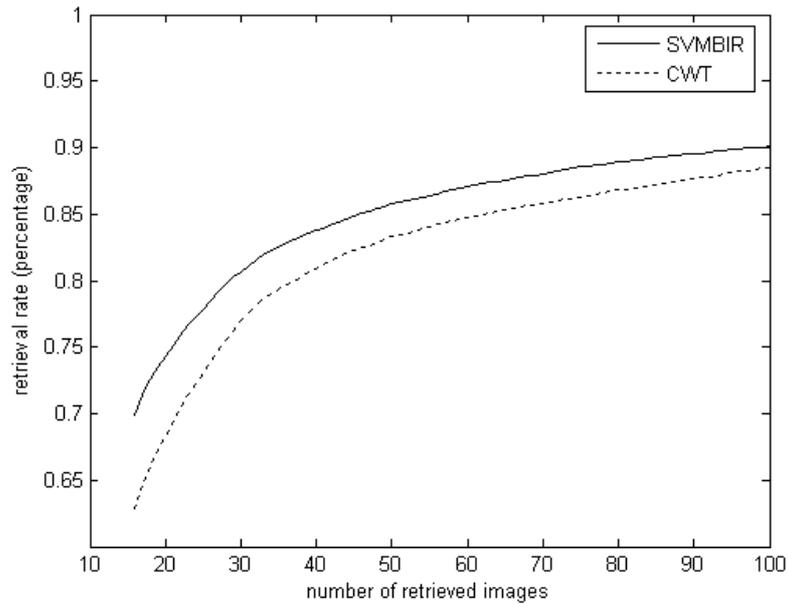
Figure 6. Retrieval rate comparsion using 25% training images.
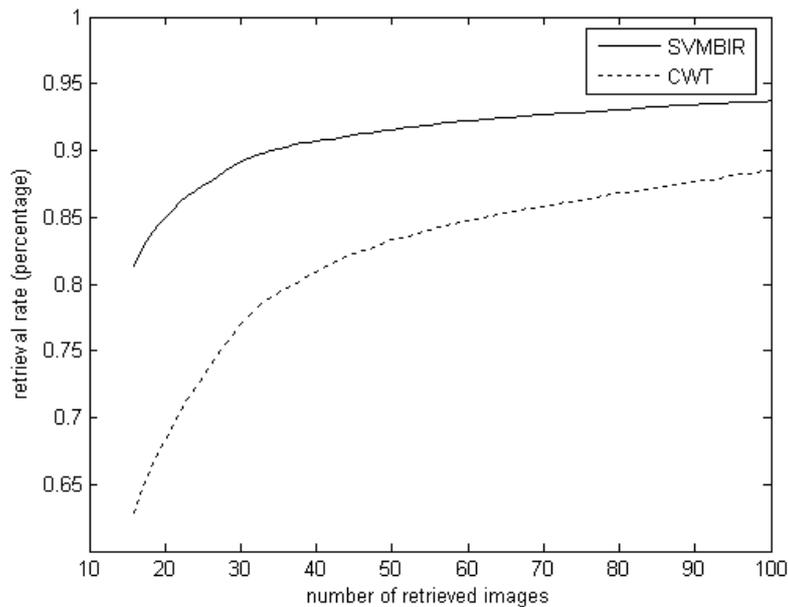
Figure 7. Retrieval rate comparsion using 50% training images.

Table 1 shows error rate (number of mismatched images) comparison of our technique (with 25% training images) and Peter and Kingsbury's technique, for different types of texture images. We used 100 query images, one from each class.

Image Category	% Error SVMBIR	% Error CWT [14]
Bark	12.5	21.87
Brick	15.62	44.79
Buildings	39.84	58.59
Clouds	28.12	50
Fabric	5.46	28.12
Flowers	18.75	33.03
Food	20.62	27.5
Grass	41.66	54.16
Leaves	28.36	39.90
Metal	15.62	33.33
Misc	6.25	43.75
Paintings	22.91	29.16
Sand	11.60	17.85
Stone	6.25	6.25
Terrain	57.81	68.75
Tile	43.75	53.12
Water	13.75	37.5
WheresWaldo	70.83	70.83
Wood	0	0
Average	24.19	37.81

Table 1. Retrieval performance for different categories of images.

Results are accumulated for different categories of images, for example Bricks include 'Brick.0002', 'Brick.0003', 'Brick.0004', 'Brick.0005', 'Brick.0006', and 'Brick.0007' images of the VisTex database. It is very clear from the table that our technique performed well almost in each category.

Figure 8 shows the comparison of our technique with Janney et al. [17]. In this comparison we used only 12.5% images from each class for training of support vector machines. As shown in Figure 8, one can see significance improvement in results.

Here we also believe that the results can be significantly improved if we use more images for training. Figure 9 shows the comparison with 25% training images. This shows that we can almost reach over 90% precision by using the idea presented in this paper.

Table 2 shows error rate (number of mismatched images) comparison of our technique (with 12.5% and 25% training images) and Janney's technique, for different types of color images. We used 20 query images from different classes. Names of query image file in coil-100 database are also shown in the table. It is very clear from the table that our technique performed well almost in each category.
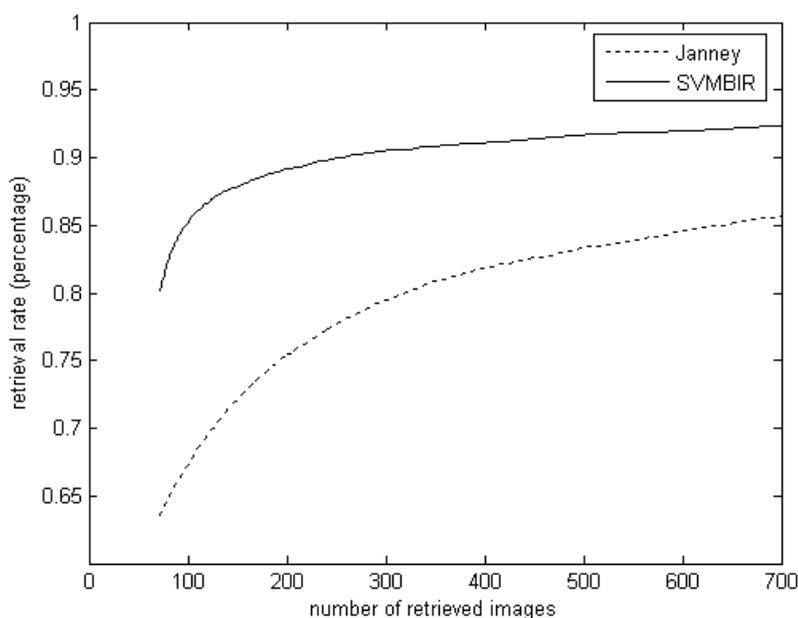
Figure 8. Retrieval rate comparsion using 12.5% training images.
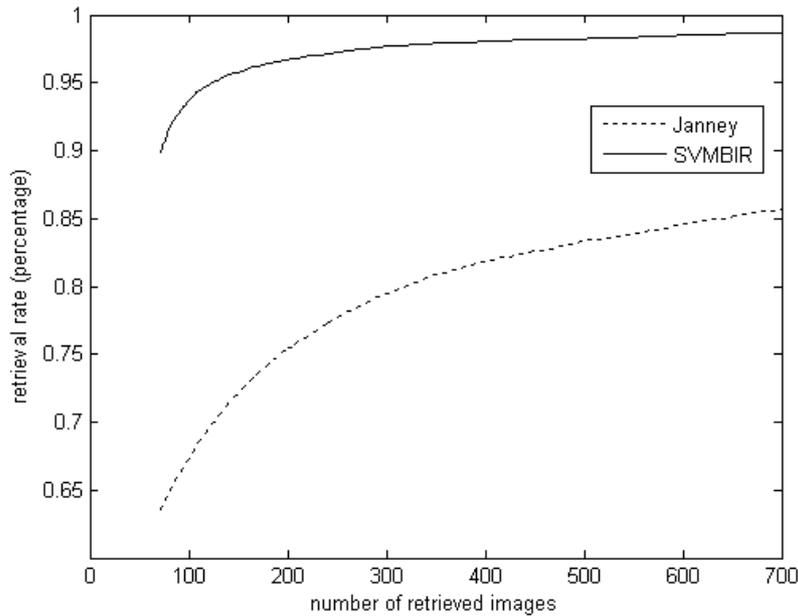
Figure 9. Retrieval rate comparison using 25% training images.

Image file name	% Error Janney [17]	% Error SVMBIR 12.5% training images	% Error SVMBIR 25% training images
Obj5_0.png	6.9444	4.1667	0
Obj10_0.png	22.222	0	0
Obj15_0.png	45.833	22.222	13.889
Obj20_0.png	40.278	19.444	13.889
Obj25_0.png	0	0	0
Obj30_0.png	0	0	0
Obj35_0.png	0	0	0
Obj40_0.png	48.611	16.667	13.889
Obj45_0.png	76.389	69.444	33.333
Obj50_0.png	0	0	0
Obj55_0.png	63.889	30.556	15.278
Obj60_0.png	77.778	61.111	38.889
Obj65_0.png	75	43.056	31.944
Obj70_0.png	0	0	0
Obj75_0.png	30.556	0	0
Obj80_0.png	56.944	58.333	44.444
Obj85_0.png	38.889	30.556	16.667
Obj90_0.png	65.278	25	11.111
Obj95_0.png	0	0	0
Obj100_0.png	40.278	33.333	25
Average	**34.444**	**20.694**	**12.917**

Table 2. Retrieval performance for different categories of images.

Figure 10 and Figure 11 show the retrieved results of the proposed SVMBIR system, in which the first image is the query image. In Figure 10, the query image is a texture image, while in Figure 11 the query is a color image. We can see from these results that the proposed system is very efficient as set of images with same conception can be retrieved, which is more similar to the human visual system.

Table 3 shows execution times for different stages of SVMBIR system. In this simulation we have used coil-100 database and 12.5% training images. Simulation was carried out in Matlab 7.4 running on Dell 360 computer having 3.0 G.Hz CPU and 1.0 GB of RAM.

Stage	Time
Feature extraction	0.2 sec/image
SVM training	18sec/class
Distance vectors	14 sec/svm
Query execution	8 sec/query

Table 3. Execution times for different stages of SVMBIR system.
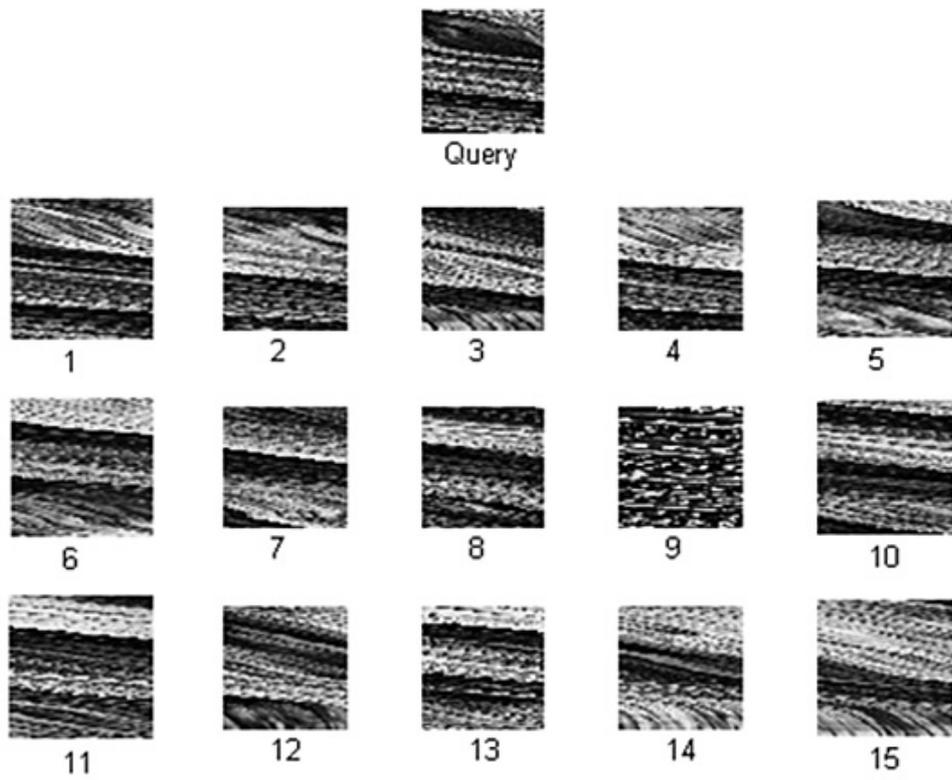
Figure 10. Retrieved results of SVMBIR system for "*Query*" image.
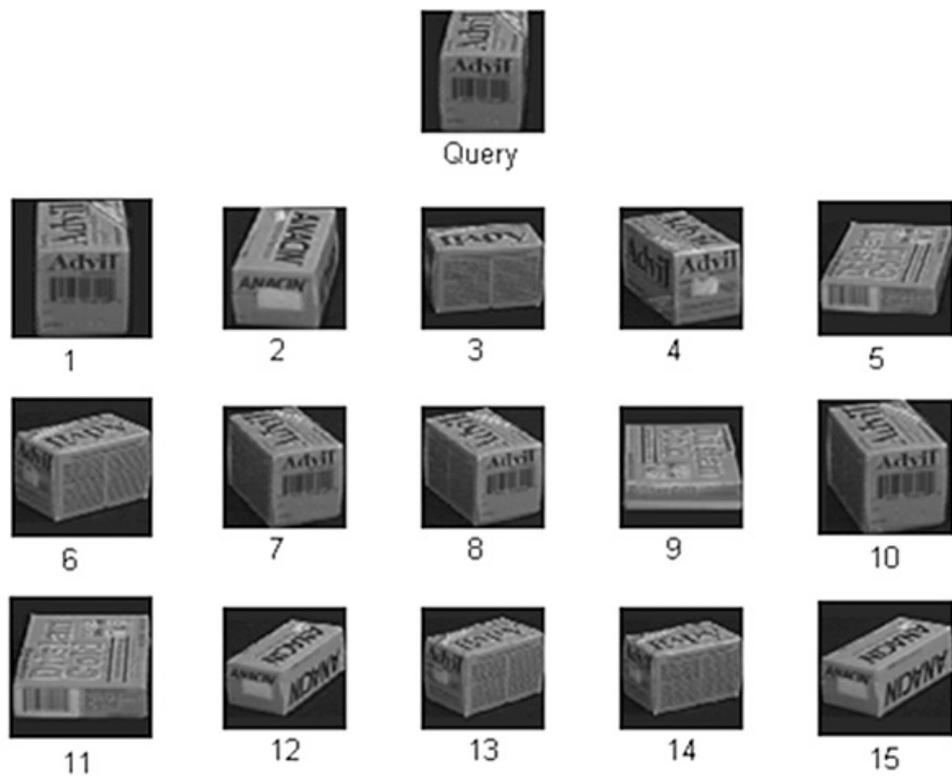
Figure 11. Retrieved results of SVMBIR system for "*Query*" image.

6. Conclusion

In this paper, we presented a novel dual tree complex wavelet transform and support vector machine-based image retrieval system. The proposed system is based on the observation that the images users need are often similar to a set of images with the same conception instead of one query image and the assumption that there is a nonlinear relationship between different features. In addition, we have shown how a correlation-based distance measure can be used to enhance the retrieval accuracy. Finally, we compare the performance of the proposed system with other image retrieval system. Experimental results show that it is more effective and efficient to retrieve visually similar images having non-linear relationship among their feature vectors.

7. Acknowledgment

References

[1] Y. RUI, T. S. HUANG, Image Retrieval: Current Techniques, Promising Directions, and Open Issues. *Journal of Visual Communication and Image Representation*, **10** (1999), 39–62.

[2] S. DEB, Y. ZHANG, An Overview of Content-based Image Retrieval Techniques. *Aina*, **1**(1) (2004).

[3] R. C. VELTKAMP, M. TANASE, Content-based Image Retrieval Systems: A Survey. UU-CS-2000-34, Department of Computer Science, Utretch University, October 2002.

[4] A. DEL BIMBO, M. MUGNAINI, P. PALA, F. TURCO, L. VERZUCOLI, Image Retrieval by Color Regions. *ICIAP 97, Image Analysis and Processing*, 9th International Conference, Ember, Florence, Italy.

[5] G. L. GIMEL'FARB, A. L. JAIN, On retriving textured images from an image database. *Patter Recognition*, **29**(9) (1996), 1416–1483.

[6] V. N. GUDIVADA, V. V. RAGHAVAN, Content-based Retrieval Systems. *Computer*, (1995), 24–33.

[7] M. FLICKNER, H. SAWHNEY, Query by Image and Video Content: The QBIC System. *IEEE Computer*, (1995).

[8] B. S. MANJUNATH, W. Y. MA, Texture features for browsing and retrieval of image data. *IEEE Trans. Patt. Anal. Mach. Int. Special Issue on Digital Libraries*, **18**(8) (1996), 837–842.

[9] J.-H. HAN, D.-S. HUANG, T.-M. LOK, M. R. LYU, A Novel Image Retrieval System Based on BP Neural Network. *International Joint Conference on Neural Networks (IJCNN 2005)*, (2005) Montréal, Québec, Canada.

[10] P. JANNEY, G. SRIDHAR, V. SRIDHAR, Enhancing capabilities of Texture Extraction for Color Image Retrieval. *WEC*, **5** (2005), 282–285.

[11] V. VAPNIK, *Statistical Learning Theory*. Wiley, New York, 1998.

[12] MIT-VISTEX-DATABASE, http://vismod.media.mit.edu/vismod/imagery/VisionTexture/vistex.html

[13] N. G. KINGSBURY, The Dual Tree Complex Wavelet Transform: A New Efficient Tool for Image Restoration and Enhancement. *Proc. European Signal Processing Conf.*, (1998), 319–322.

[14] R. PETER, N. KINGSBURY, Complex Wavelets Features for Fast Texture Image retrieval. *Proc IEEE Int. Conf. on Image Processing*, (1999), 25–28.

[15] M. KOKARE, P. K. BISWAS, B. N. CHATTERJI, Texture Image Retrieval Using New Rotated Complex Wavelet Filters. *SMC-B*, **35**(6) (2005), 1168–1178.

[16] J. A. K. SUYKENS, T. VAN GESTEL, J. DE BRABANTER, B. DE MOOR, J. VANDEWALLE, *Least Squares Support Vector Machines*. World Scientific, Singapore, 2002.

[17] P. JANNEY, G. SRIDHAR, V. SRIDHAR, Enhancing Capabilities of Texture Extraction for Color Image Retrieval. In *Proceedings of World Enformatika Conference (Turkey)*, (2005).

[18] COLUMBIA OBJECT IMAGE LIBRARY (COIL-100), http://www1.cs.columbia.edu/CAVE/software/softlib/coil-100.php

[19] C. J. C. BURGES, A Tutorial on Support Vector Machines for Pattern Recognition. *Data Mining and Knowledge Discovery*, **2**(2) (1998), 955–974.

[20] M. G. ALBANESI, M. FERRETTI, A. GIANCANE, A Compact Wavelet Index for Retrieval in Image Database. *Iciap*, **00** (1999), 927.

Contact addresses:
Adeel Mumtaz
Faculty of Computer Sciences
GIK Institute, Pakistan
e-mail: adeel@giki.edu.pk,
adeelmumtaz@hotmail.com

Syed Asif M. Gilani
Faculty of Computer Sciences
GIK Institute, Pakistan
e-mail: `asif@giki.edu.pk`

Kamran Hameed
Faculty of Computer Sciences
AJK University, Pakistan
e-mail: `kjarral71@hotmail.com`

Tahir Jameel
Faculty of Computer Sciences
GIK Institute, Pakistan
e-mail: `teejay@giki.edu.pk`

ADEEL MUMTAZ received his BS (CIS) from Pakistan Institute of Engineering and Applied Sciences, Pakistan. At present, he is a graduate student at GIK Institute, Pakistan. His area of interest includes content-based image retrieval, data mining and digital image watermarking.

DR. SYED ASIF GILANI received his M.Sc from Islamia University Pakistan and Ph.D in digital image watermarking from University of Patras, Greece. He is assistant professor at Ghulam Ishaq Khan Institute of Engineering Sciences and Technology, Pakistan. His research interests include digital image watermarking, steganography and image authentication. He has published a number of research papers internationally. At present, he is supervising many MS/Ph.D students at GIK Institute. He is also one of HEC and PCST approved Ph.D supervisors.

KAMRAN HAMEED received his MS in computer science from GIK Institute Pakistan At present, he is assistant professor at the AJK university Pakistan. His area of interest includes content-based image retrieval, data mining and digital image watermarking.

TAHIR JAMEEL received his BSE from Bahria University, Pakistan At present, he is a graduate student at GIK Institute, Pakistan. His area of interest includes content-based video retrieval, data mining and digital image watermarking.