Progressive Query Based Search and Retrieval in Large Image Archives

Prof.D. Rajya Lakshmi†, Dr. A. Damodaram††, Dr. B. Raveendra Babu†††, Dr. J.A. Chandu Lal††††
dlakmi@rediffmail.com, adamodaram@jntuap.ac.in, rbhogapathi@yahoo.com, profanandlal@yahoo.co.in

† ANITS Engg. College Visakhapatnam, India
†† JNTU College of Engg. Hyderabad, India
††† RVR & JC Engg. College Guntur, India
†††† GITAM University Visakhapatnam, India

Summary
In this paper, we describe the architecture and implementation of a framework to perform content based search of an image database, where content is specified by the user at one or more of the following three abstraction levels: pixel, feature, and semantic. Query based Image Acquisition System deals with query passing, query parsing, SQL query generation and image retrieval. The image is retrieved just based on the query given by the user. It provides a user friendly interface for user to pass his query. The required image is returned to the user based on his query by performing a search on the database. All this procedure is carried on in semantic level. This framework is well suited for searching scientific databases, such as satellite image, medical, and seismic data repositories, where the volume and diversity of the information do not allow the apriori generation of exhaustive indices, but we have successfully demonstrated its usefulness on still-image archives.
Key words:
Progressive Searching, Content Based, Templates, DWT, Texture Analysis.

1. Introduction

In general, there exist three different levels of abstraction at which images can be defined and searched -semantic, feature, and pixel. For instance, one can search for images containing houses (semantic level), regions with a specified texture (feature level), or pixel-by-pixel. Objects at each level can be either pre-extracted and indexed (an index is a data structure designed to allow efficient retrieval) when new images are placed in the data repository, or evaluated at query time by analyzing pre-extracted information or even the images themselves. Current systems tend to pre-extract as much information as is practical, in order to allow efficient indexing and access to the desired information.

Pixel level cannot be used to retrieve the images because of type and resolution mismatch. This system will overcome this problem by converting the images of different types and resolutions into a raw image by using different decoding techniques and match it with the database images. Because this has been prohibitively expensive, most systems avoid the use of pixel level techniques on large archives.

Feature based search targets particular features such as texture which include either by providing sample images from which feature vectors for search are extracted, or by explicit specification of feature values or ranges.

In the semantic abstraction level of image extraction we have to classify and assign labels to distinct objects.

Semantic objects are typically predefined by applying classification algorithms to image features (such as texture or spectral histograms), or to the original data.

Extraction of features from pixels and of semantic objects from features are lossy operations. Although each higher level of abstraction improves search efficiency by reducing information volume, there are corresponding losses of accuracy. For this reason, it is necessary to provide mechanisms that support searches specified at all of these abstraction levels.

At the present time, there are many systems that perform indexing of image or video through the use of low-level image features such as shape, color histogram and texture. Prominent examples for photographic images include IBM QBIC[1], the MIT PhotoBook[2], VisualSeek from Columbia University[3,6], and the Multimedia Datablade from Infomix/Virage [4]. These techniques have also been applied to specific application domains, such as medical imaging[3], art, and video clips[5,7,8,9]. All of these examples rely on a preprocessing stage in which appropriate features are extracted and indexed. Although pre-extracted features and/or semantic objects permit efficient indexing schemes, they do not allow searching with all possible query semantics. To support a wide range

Manuscript received August 5, 2007
Manuscript revised August 20, 2007

of content-based queries, we must allow the user to form new semantic categories and/or new feature definitions.

The goal is to explore technologies that will facilitate the storage, query, and retrieval of images from large digital libraries by a diverse community of users. It is necessary to provide users the capability to dynamically define new features and objects, and interactively specify and refine the target for content-based search of the image data. Thus, to extract the content dynamically defined by the user, we need the ability to analyze the raw images in the repository at query time. But the large size of the images stored in typical scientific repositories and the complexity of the operations involved make it impossible to use traditional image- processing operations in an interactive system. To overcome this difficulty, an extendible framework have developed (called a progressive framework) in which the search targets are specified at one or more different abstraction levels.

Organization of rest of the paper is as follows. Section 2 describes about detailed System Architecture, section3 about compression using Discrete Wavelet Transform, section 4 describes about our modified work which includes Texture Analysis, Classification, Semantic Level matching. In section 5 Query Based image retrieval results are shown and section 6 includes discussions and conclusions.

2. System Architecture

Figure 1, shows system architecture and detailed architecture is shown in Figure 2. As shown in the following figure 1, the system has a client-server architecture, with communication between client and server over the internet. The client program is a Java program, which allows the user to navigate through the image database, to construct content-based queries and to specify the format for visualizing query results. The server architecture is centered on a query parser, that operates on features and objects that are both pre-extracted and computed at query execution time. The client issues queries that are translated into programs and parsed by the query parser. In this section, the architecture and the basic functionality of content-based search system is described. As shown in Figure, system is composed of a client and a server using the http protocol to communicate across the Internet.

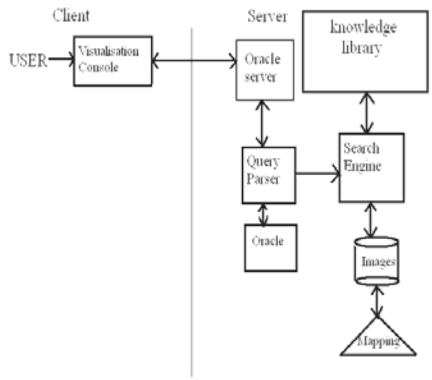


Figure 1:Proposed Architecture

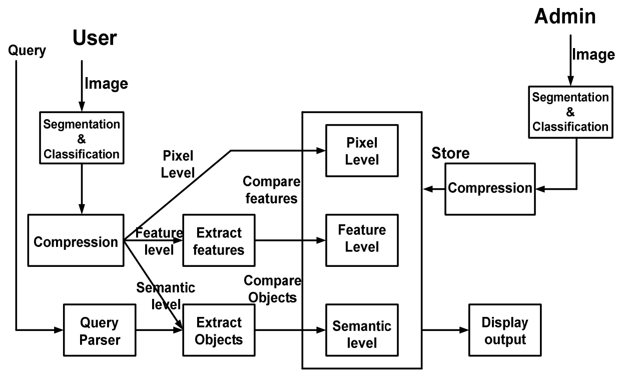


Figure.2 Detailed Architecture

Pre-extracted objects or features are stored in a database. The schema of the database is shown in the following figure, and consists of the following three components:

- Metadata Table: which contains information such as the coordinates of the bounding box of the image, instrument type, date of generation, and other relevant information. This information is used to perform initial pruning of the search space based on location, date and resolution.
- Object Table: which records the object type and geographic coordinates of all the pre-extracted objects.
- Feature Table: which records the feature vectors extracted from the images. Currently, the table stores 8 different texture features including fractal dimensions.

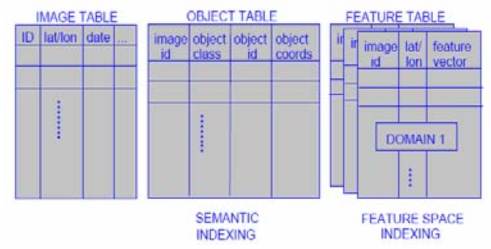


Figure 3: Schema of Database

3. Compression and progression

The first step of a large class of compression algorithms is a transform, such as the discrete cosine transform (DCT) used in both JPEG[10] and MPEG, or the discrete wavelet transform (DWT)[11] that improves the compression performance by concentrating most of the information (i.e., most of the energy) in a few transform coefficients[18]. The transformed data is then "thresholded" (i.e., values less than a specified threshold are changed to zero) in order to eliminate the coefficients that are close to zero, quantized, and finally coded using lossless compression techniques. The thresholding and quantization steps in existing lossy compression standards such as JPEG are usually designed to minimize the perceptual difference between the original and the stored data. However, quantization schemes can be selected to best suit the requirements of the intended applications.

Applying query and retrieval operations directly on lossy compressed data generally leads to improved computational efficiencies along two fronts:

- The required I/O bandwidth between storage units and CPU is significantly reduced.
- The features and properties of the data can be emphasized by the transform step of the compression scheme.

In particular, query operations (e.g., analysis, retrieval, evaluation, transmission, and visualization) on image can be staged progressively to minimize the total execution time as follows: Instead of operating on an entire image, the algorithm first analyzes small, selected portions of the transform and, on the basis of this information, decides whether the image can satisfy the query. Images that do not satisfy the query are quickly discarded; the others are further analyzed by accessing additional portions of the transformed data and the process is repeated. At each step, the search space is progressively refined.

Wavelet-based compression

The system uses a transform-based image-coding scheme, which we call multiresolution compression for image analysis[16](MCIA), that simultaneously yields lossless and lossy compression. The structure of our algorithm is shown schematically in Figure 4. It consists of a lossy component and a lossless component; the lossy component uses DWT coding followed by quantization of the transform coefficients and lossless coding of the quantized values. To achieve lossless compression, we compute the difference between the lossy image and the original image, to produce a residual image, which is then coded in a loss less way.

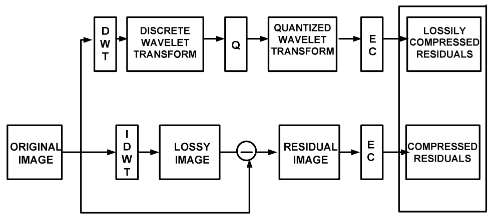


Figure 4 Structure of Compression Algorithm

DWT – Discrete wavelet Transform
IDWT – Inverse Discrete Wavelet Transform
EC – Entropy Coding
Q - Quantization

A Wavelet Transform is used , such as the Daubechies bi-orthogonal symmetric wavelets of order 4 (although the system supports a large variety of wavelets). If the original data is stored in integer format, these filters allow perfect reconstruction; i.e., the transform step is perfectly invertible (of course, if the original data is stored as real numbers and the computation is performed with finite precision, rounding errors might prevent perfect reconstruction). The wavelet transform takes as input an image and produces a matrix of coefficients with the same number of rows and columns. This matrix is divided into portions called sub-bands, which, from the signal-processing viewpoint, are obtained by separating different spectral components using linear filtering operations, and sampling the results.

The coefficient matrix is quantized[18] using a uniform scalar quantization scheme. A different number of bits for each coefficient is allocated for each subband. Since quantization results in a loss of information, this step is the (only) lossy portion of our coding scheme. The quantized subbands are then losslessly coded independently by means of predictive coding followed by a fixed-model two-pass arithmetic coding[10,19]. These two steps are denoted by EC (entropy coding) in the figure. The resulting lossy compressed image requires ten to twenty times less space than the original, without displaying appreciable artifacts.

The residual is computed by inverting the DWT of the quantized wavelet coefficients and calculating the difference between that and the original image; this difference is then losslessly encoded, using DPCM followed by arithmetic coding. The residual contains information that is difficult to predict (hence to compress), such as the sensor noise in satellite images. Thus, the residuals account for most of the storage requirement.

a)

b)

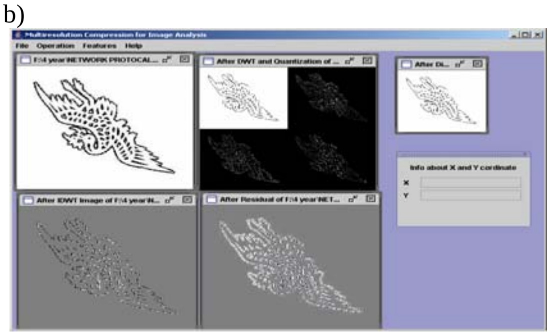


Figure 5

a) Original Black & White Image
b) After compression

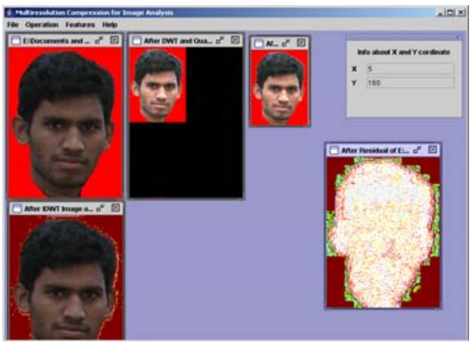


Figure 6: Color Image after compression

Figures 5 & 6 shows the results of DWT, IDWT and residual and compressed images for B/W and color images. Energy, Coarseness, Entropy, Contrast values are defined in eqns.(1),(2),(3),(4). Histogram values, Entropy, Energy, Dispersion, Contrast, Homogeneity values for B/W and color images are shown in Figures 7 & 8

Energy:

$$\sum_{i,j} P(i,j)^2 \quad ..(1)$$

Coarseness:

$$C = 1 - \frac{1}{1+S_D} \quad S_D = \sum_{i=0}^{L-1} (i - S_M)^2 h[i] \quad ..(2)$$

Entropy

$$S_E = -\sum_{i=0}^{L-1} h(i) log_2 h[i] \quad ..(3)$$

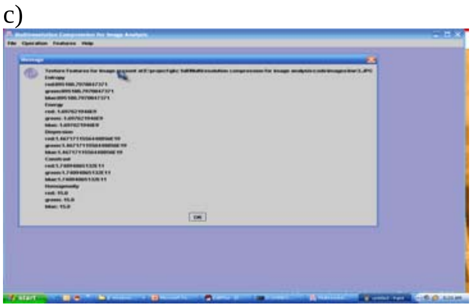
Contrast:

$$\sum_{i,j} (i-j)^2 P(i,j) \quad ..(4)$$

a)

b)

c)

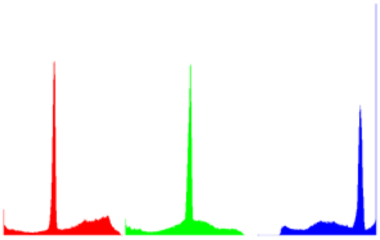


d)

Figure 7

a) Original black and white image
b) Histogram values for B/W Image
c) Entropy, Energy, Dispersion, Contrast, Homogenety, values for B/W Image
d) Color Image

a)



b)

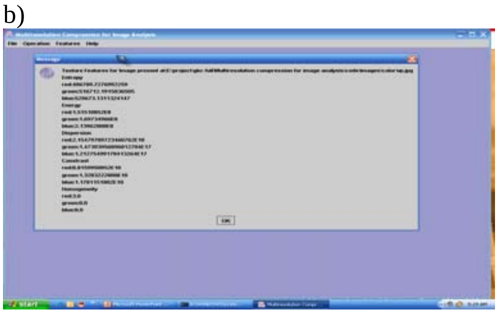


Figure 8

a) Histogram Values for color Image
b) Entropy, Energy, Dispersion, Contrast, Homogeneity Values for color picture

4. Modified Work

The primary objective of this system is to provide a framework for automatic search of an image repository based on the image content.

Combining compression and image-processing operators has proved very effective in reducing the high computational complexity of the feature-extraction task, and sometimes in increasing the accuracy of statistical operators, such as classification.

While the approach has been used in conjunction with a variety of elementary image-processing operators, we discuss in more detail how to apply the progressive framework to three complex operations: texture extraction [12], which retrieves information at the feature level; and classification [15], which attaches semantics to portions of the images.

Texture Analysis

Texture is a perceptual concept that is rather difficult to formalize. It roughly captures regularity and organization, or lack thereof, of the luminance of neighboring pixels. As such, it is a local property of an image; i.e., different regions can have different texture. Many different approaches have been proposed [13] to capture mathematically or numerically the concept of texture. Usually, the image is divided into square, rectangular, or irregular blocks, possibly overlapping, and certain numerical quantities, called texture features, are extracted from each block.

Texture extraction is a rather time-consuming task, since it involves the computation of complex quantities for each block of the image. For interactive situations in which the features cannot be pre-computed, such as content-based search of real-time images and video, feature extraction [12] can become the major bottleneck. In our system, we pre-extract texture vectors, store them in files, and generate indexes. Still, given the large amount of data associated with each image, similarity search is a daunting task.

To improve the speed of searching an image or video database for texture similarity, we use an approach called progressive texture matching. The goal is to find a specified number of textures in the database that are most similar to an example that the user provides by specifying one or more blocks of existing images. We assume that a progressive representation of the images and the videos being searched already exists, and in the following we identify it with an L-level multi-resolution pyramid created by Sub-band coding or by the wavelet transform. For each resolution level of the pyramid, we can divide the image approximation into regular regions (for example, overlapping square blocks) or into irregular regions according to boundaries determined by edge-detection or image-segmentation techniques, and compute a texture-feature vector for each of the regions. In general, lower-resolution approximations will result in a substantially smaller number of texture vectors. The extracted feature vectors can then be stored separately, level by level, together with other information about the region from which they were extracted, and indexed. The procedure progressively refines the search space by further pruning it at each level.

Classification

Classification of an image is the process of assigning semantic labels to the individual pixels or to distinct regions. In the remote-sensing community [13,14.15], images are often classified by analyzing each pixel location independently, and using as input to the classifier the spectral bands acquired by the instrument (typically four to two hundred). Different land-cover classes have different "spectral signatures"; for instance, water has low reflectance across the visible and infrared portion of the spectrum, vegetation has low reflectance in the red and high reflectance in the green and near-infrared bands, and barren terrain has similar, medium-high reflectance in all bands.

This scheme has two advantages. First, it is faster than the full-resolution, pixel-by-pixel approach, since in images from nature there is significant dependence between the labels of adjacent pixels, which allows large portions of the image to be classified at low resolution. Second, somewhat surprisingly, it is more accurate than the pixel-by-pixel approach.

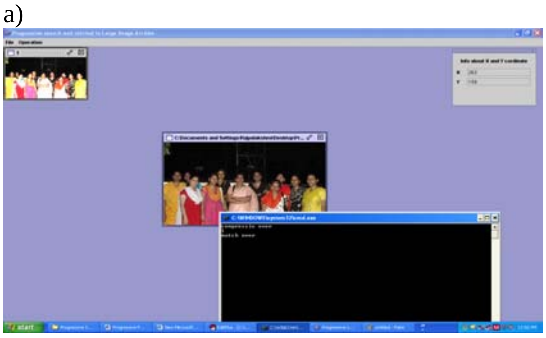
Semantic Level Matching

At semantic abstraction level images are searched based upon content of image required by the user. So, first and fore-mostly, before inserting images into the database each and every content of the image is classified and stored. Images may contain many objects in it and by using the classification technique illustrated in the above section each object is given its tag name. So, at this level user specifies content that is, what he wants in his required image. Based on tag names we search for the image in the database. This level is the best of all techniques and accurate.

5. Query based image retrieval results:

Our system provides a user-friendly access. User need not give the queries in technical terms or search for the efficient words to improve the search .The syntax in which query is to be passed is predefined The query given by the user is parsed into tokens and the required terms are identified i.e., object name, image name etc., The equivalent SQL query is generated. The images are retrieved based on these queries .The images are retrieved by performing a search on the existing databases. The exact features or objects are retrieved from the respective images and the result is returned, thus reducing the time factor. We have given group photo as input as shown in figure 9(a), and it also displayed matched result. Figure 9(b) & (c) represents the outputs of query based search result for user specified format 1 shown in fig.9(b), and equivalent SQL generated query. Figure 10 (a),(b),(c) are query based results of other user specified formats 1 and 2 shown in fig.9(b), and equivalent SQL generated query.

a)



b)

c)

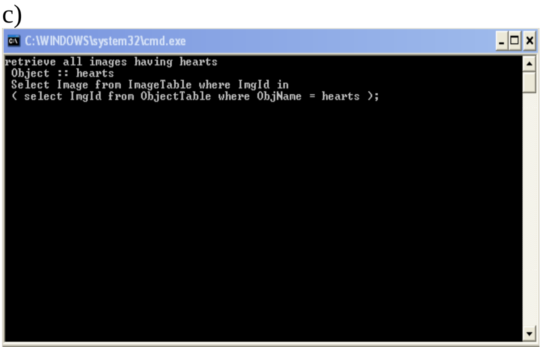


Figure 9:

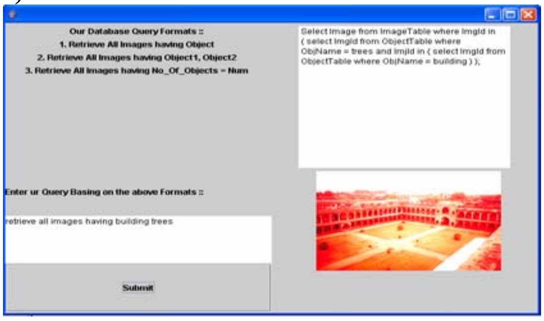
a) Progressive search matching result

<u>Query based retrieval Database Query Formats:</u>

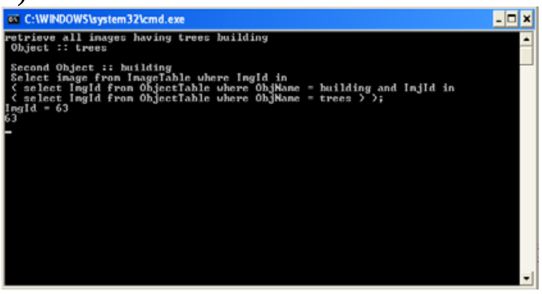
c) Retrieve all images having object1 (hearts)

d) Equivalent Generated SQL Query

a)



b)



c)

Figure 10:

a) Retrieve all images having object1, object 2 (buildings and trees).
b) Equivalent Generated SQL Query
c) Retrieve all images having no. of objects = Num

6. Discussion and Conclusion

Pre-extracting information and generating indices becomes too time-consuming and impractical. We have explored an approach that allows the user to define new features at query-construction time and to use such features to specify new, arbitrarily complex searches. Within our paper, we have developed and demonstrated technologies to support this capability.

The main contribution of our system is the capability of further narrowing the search results by extracting and manipulating user-defined features at query-execution time from this candidate set. Previously, this process has been regarded as impractical, especially in an interactive system, because of the high computational cost of the image-processing operations involved.

To overcome this difficulty, it is proposed a progressive framework that combines image representation (in particular, image compression) with image processing. Progressive implementations of image-processing operators rely on the properties of the compression scheme to reduce significantly the amount of data to analyze during the feature-extraction and manipulation phases.

The progressive operators are capable of extracting user-specified features at query time, and we can use these features to search for new, non-predefined content, thus adding a new dimension to the flexibility of our query engine.

Efficiency of Compression is further improved by using wavelets of order 8 and this work is under progress.

Acknowledgments

We are very grateful and sincerely thank the cooperation given by the team of “Progressive search and retrieval in large image archives” by V. Castelli, L. D. Bergman, I. Kontoyiannis, C.S. Li, J. T. Robinson and J. J. Turek

References

[1] W. Niblack, R. Barber, W. Equitz, M. Flickner, E. Glasman, D. Petkovic, P. Yanker, C. Faloutsos, and G. Taubin, "The QBIC Project: Querying Images by Content Using Color, Texture, and Shape," Storage Retrieval for Image and Video Databases, Proc. SPIE 1908, 173-187 (1993).

[2] A. Pentland, R. W. Picard, and S. S. Sclaroff, "PhotoBook: Tools for Content-Based Manipulation of Image Databases," Storage and Retrieval for Image and Video Databases, Proc. SPIE 2185, 34-47 (1994).

[3] J. R. Smith and S.-F. Chang, "VisualSeek: A Fully Automated Content-Based Image Query System," Proceedings of ACM Multimedia '96, Boston, November 1996, pp. 87-98.

[4] J. R. Bach, C. Fuller, A. Gupta, A. Hampapur, B. Horowitz, R. Humphrey, and R. Jain, "The Virage Image Search Engine: An Open Framework for Image Management," Storage and Retrieval for Still Image and Video Databases, Proc. SPIE 2670, 76-87 (1996).

[5] T. Y. Hou, A. Hsu, P. Liu, and M. Y. Chiu, "A Content Based Indexing Technique Using Relative Geometry Features," Image Storage and Retrieval Systems, Proc. SPIE 1662, 29-68 (1992).

[6] B. Holt and L. Hartwick, "Visual Image Retrieval for Applications in Art and Art History," Storage and Retrieval for Image and Video Databases, Proc. SPIE 2185, 70-81 (1994).

[7] E. Deardorff, T. D. C. Little, J. D. Marshall, and D. Venkatesh, "Video Scene Decomposition with the Motion Picture Parser," Digital Video Compression on Personal Computers: Algorithms and Technologies, Proc. SPIE 2187, 44-55 (1994).

[8] H. Zhang and S. W. Smoliar, "Content Based Video Indexing and Retrieval," IEEE Multimedia 1, No. 2, 62-72 (1994).

[9] F. Arman, A. Hsu, and M. Y. Chiu, "Image Processing on Compressed Data for Large Video Database," Proceedings of ACM Multimedia '93, 1993, pp. 267-272.

[10] William B. Pennebaker and Joan L. Mitchell, JPEG Still Image Data Compression Standard, Van Nostrand Reinhold, New York, 1993.

[11] Oliver Rioul and Martin Vetterli, "Wavelets and Signal Processing," IEEE Signal Process Magazine 8, No. 4, 14-38 (1991).

[12] Chung-Sheng Li and Vittorio Castelli, "Deriving Texture Feature Set for Content-Based Retrieval of Satellite Image Database," Proceedings of the 1997 IEEE International Conference on Image Processing, November 1997, pp. 567-579.

[13] A. R. Rao, A Taxonomy for Texture Description and Identification, Springer-Verlag, New York, 1990.

[14] C.S. Li and M..S. Chen, "Progressive Texture Matching for Earth Observing Satellite Image Databases," Proceedings of SPIE Photonics East, Proc. SPIE 2916, 150-161 (1996).

[15] Samir R. Chettri, Robert F. Cromp, and Mark Birmingham, "Design of Neural Networks for Classification of Remotely Sensed Imagery," Telematics & Informatics 9, No. 3/4, 145-156 (1992).

[16] Vittorio Castelli, Ioannis Kontoyiannis, Chung-Sheng Li, and John J. Turek, "Progressive Classification: A Multiresolution Approach," Research Report RC-20475, IBM Thomas J. Watson Research Center, Yorktown Heights, NY, 1996.

[17] Vittorio Castelli, Chung-Sheng Li, John J. Turek, and Ioannis Kontoyiannis, "Progressive Classification in the Compressed Domain for Large EOS Satellite Databases," Proceedings of the 1996 IEEE International Conference on Acoustics, Speech and Signal Processing, May 1996, pp. 2201-2204.

[18] Thmos M.Cover and Joy A. Thmos. Elements of Information Theory.Wiely Series in Telecommunications.John Wiely and Sons,1991

[19] Ian H. Witten, Radford M. Neal, and John G. Cleary, "Arithmetic Coding for Data Compression," Commun. ACM 30, No. 6, 520-540 (1987).

D. Rajya Lakshmi is working as Professor in the department of IT at ANITS Engineering College, Visakhapatnam, AP, India. Her research areas includes Image processing , Data mining. She has about 15 years of teaching and research experience. She is now doing her Ph.D in Image processing area.

Dr. A. Damodaram is Vice Principal and Professor of CSE, JNTU College of Engineering, Hyderabad. His research interests include Software Engineering, Computer Networks and Image Processing. Prof. Damodaram was awarded his Ph.D. in CSE from JNTU. He has a rich experience of 17 years in Teaching, Research and mentoring research scholars in his respective areas. He is Member of Academic Council in Cochin University of Science and Technology, Cochin. He is a member of AIEEE, New Delhi and Governing Council, JNTU College of Engineering, Hyderabad.

Dr. B. Raveendra Babu is working as Professor & HOD of CSE in RVR & JC College of Engg., Guntur, AP, India. He has presented the papers like “Impact of IT on Continuing Education”, “The Impact of the Internet on Higher Education”, and “Quality Education Through Continuous Evaluation – Role of IT”. Professor Raveendra Babu was awarded Ph.D by S.V. University, Tirupati. He has completed his M.S. in “Software Systems” from BITS, Pilani. He is life member in Indian Society for Technical Education, New Delhi, India and also in Computer Society of India.