Investigation of Compressed Image Segmentation

S.Raja, S.Sankar, S.Saravanakumar

Abstract— **This paper presents an unsupervised texture image segmentation algorithm using clustering. Two criteria are proposed in order to construct a feature space of reduced dimensions for texture image segmentation, based on selected Gabor filter set. An unsupervised clustering algorithm is applied to the reduced feature space to obtain the number of clusters, i.e. the number of texture regions [1]. A simple Euclidean distance classification scheme is used to group the pixels into corresponding texture regions. Experiments on a mixture of mosaic of textures generated by random field model show the proposed algorithm of using the clustering gives satisfactory results in terms of the number of regions and region shapes. A mathematical programming based clustering approach that is applied to a digital platform of segmentation problem involving demographic and transactional attributes related to the images. It is the collection of sub images corresponding to different image regions and scales is obtained. From the experiments, it is found that the multistage sub image matching method is an efficient way to achieve effective texture retrieval for image segmentation.**

Keywords- **Image segmentation, Texture feature extraction, sub-images matching, Clustering, Texture segmentation, Gabor filter set.**

I. INTRODUCTION

The multistage approach to extract colour features using the Multistage Colour Coherence Vector (MCCV), manages to handle the problem of sub-image colour retrieval effectively. The idea of multistage feature extraction technique has found much success in the area of colour analysis. The multistage approach can be used to extract texture information using any texture feature extraction method, the Discrete Wavelet Frames (DWF). Wavelet-based feature extractors are chosen because of their high classification accuracy and low computational load [1].

In this fundamental process in digital image processing which has found extensive application in areas such as image retrieval, medical image processing, and remote sensing [2]. Texture is an essential key to the segmentation of image and originates from the scale dependent spatial variability, and is observed as the fluctuation of gray levels in images [3]. It is well known that another source of the fluctuations exists in images. Even though the human interpreter often is superior in identifying strange details and phenomena in images, there is still a need to automate this process.

Manuscript received November 01, 2013.

Mr.S.Raja is an Assistant Prof. of CSE, Panimalar Institute of Technolog y, Chennai, TamilNadu, India.
Dr.S.Sankar is a Faculty in Department of EEE, Panimalar Institute of Technology,Chennai.Tamil Nadu ,India.
Dr.S.Saravanakumar is a Prof. of IT, Panimalar Institute of Technology, Chennai, TamilNadu, India.

The clustering leads to the concept of unsupervised segmentation or classification of images. The notion of scale gives an important hint to texture analysis. Wavelet packet transform have received much attention as a promising tool for texture analysis, because they have the ability to examine a signal at different scales. The texture feature set derived from the wavelet de-composition where the sampling between wavelet levels is omitted is superior to that derived from the standard wavelet decomposition [4], [5].The trend probably therefore seems to be a concentration on the wavelet packet transform which basically is the wavelet transform with sub band decompositions not restricted to be dyadic. The discrete wavelet packet transform are critically sampled multi rate filter banks. However, critically sampled filter banks typically imply inaccurate texture edge localization. Therefore, in this paper, wavelet packet decomposition is employed to extract local texture features from image.

From the sub-band filtering point of view, the difference between the wavelet packet transform and the conventional wavelet transform is that the former also recursively decomposes the high frequency components, thus constructing tree structured multi-band extension of the wavelet transform wavelet packet transform decomposes an image into sub images[6,7]. In the Fig.1 shows the standard decomposition. H and L in Fig. 1 denote a high pass and low pass filter respectively. The approximated image LL is obtained by low pass filtering in both row and column directions. The detail images, LH, HL, and HH, contain high frequency components.

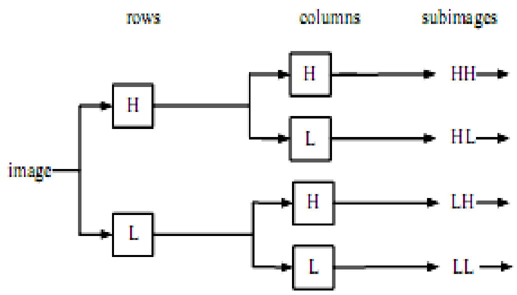


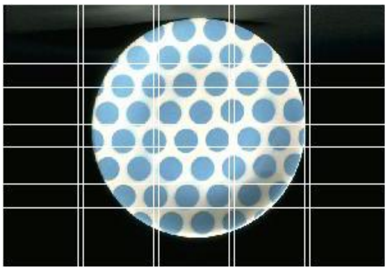
Fig.1 Decomposition of images into sub-images

Sub-image classification is done by indexing sub regions and no specific objects. The image is partitioned in overlapping rectangles of different sizes and this allows dealing with the different scales of objects in the image. Each of these regions is represented by global information and, for efficiency reasons, instead of storing the whole global features, only the differences of the feature vectors are stored.

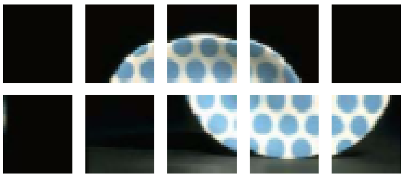
The segmentation process is repeated at several different resolutions or scales of the image. The result of this process is a collection of sub-images each corresponding to different regions of the parent image.

A. Sub-Image Querying

The sub-image querying problem can be defined as follows: Given as input query a sub-image Q of an image I and an image set S, retrieve from S those images Q' in which query Q appears (denoted Q<= Q').The problem is made more difficult than image retrieval by a wide variety of effects (such as changing viewpoint, camera noise, etc.) that cause the same object to appear different in different images . The patterns belonging to each category will form a cluster in the feature space which is compact and isolated from clusters corresponding to other texture categories.



First stage: original image



Sub-image tilling

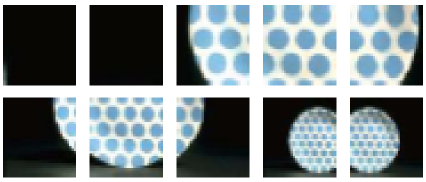


Fig.2 Sub-image querying

The above condition however might be suitable for colour features, but might not work for texture features. By non-linearly re-scaling the resolution of the input image, the texture characteristics of the image might be affected, and features extracted from it might not reflects an accurate representation of the texture, hence resulting in incorrect retrieval. Because of this, multiscale resolution is used from the original size of the input image in the proposed algorithm. Because of the possibly non-dyadic nature of the image size, the algorithm ends up with a collection of slightly overlapping (64×64) sub-images at the current resolution.

A suitable texture feature extraction can be performed on all sub-images separately. Each time a feature vector of a sub-image is produced, it is added to the final feature vector, which represents the multiscale feature vector of the original image. After all sub-images of the current scale have been processed, the image is linearly rescaled and the whole process is repeated.

B. Salient Location Identification

The wavelet decomposition is used because of its computational simplicity. The decomposition algorithm is based on the non-standard decomposition procedure.

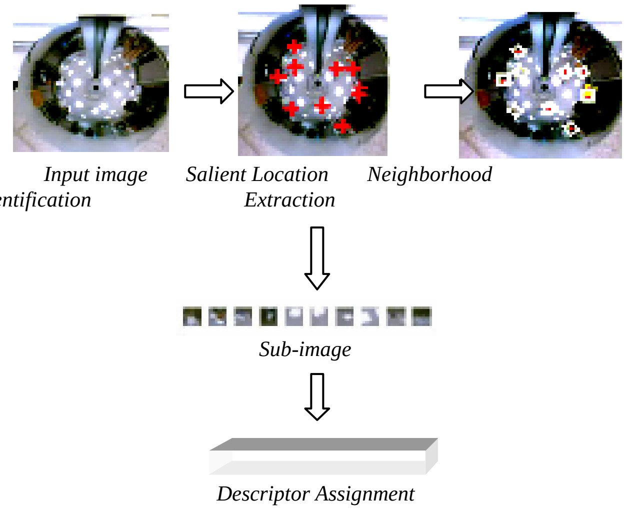


Fig.3 *Neighborhood identification and extraction*

The wavelet decomposition of the feature extraction process from the sub-images are obtained from the neighborhood region, and the descriptor vectors are computed from each sub-image.

An input image yields three coefficient matrices, namely the Horizontal (**H**), Vertical (**V**), Diagonal (**D**), and the compressed image to a quarter of the size of the original image. The decomposition can be recursively performed by using the output image from the previous iteration as the input. At each decomposition level, wavelet coefficient matrices are combined into one signal matrix S:

$$S(n, m) = |H(n, m)| + |V(n, m)| + |D(n, m)| \quad (1)$$

The salient locations are identified by each element of the last signal matrix (last wavelet decomposition) to the untransformed input image. The localization approach determines the location of a android when it travels near the place that a reference image was taken as in (1). Hence, the current query image is different from the corresponding reference image by certain rotation angles. The neighborhood region around each salient location is extracted to make it rotationally invariant using the coordinates transformation in equation (2). As a result, the sub-images of the same salient point extracted from this method are having the same appearance regardless of rotational deviations from an original image.

$$X_n = X_f + (u - M/2)\cos a_f \quad (2)$$
$$Y_n = Y_f + (v - M/2)\sin a_f$$

Where (u, v) are the transformed (new) coordinates, (x_n, y_n) are untransformed coordinates in the original image, and (fx, fy) is the coordinate of the salient point $a_{f=tan}^{-1}(X_f / Y_f)$. Classically, image segmentation is defined as the partitioning of an image into overlapping, constituent regions which are homogeneous with respect to some characteristic such as intensity or texture. If the domain of the image is given by I, then the segmentation problem is to determine the sets S<=I whose union is the entire image I.

II. BACK PROPOGATION SYSTEM

In this back-propagation algorithm is a widely used learning algorithm in Artificial Neural Networks. The Feed-Forward

Neural Network architecture is capable of approximating most problems with high accuracy and generalization ability. This algorithm is based on the error-correction learning rule. Error propagation consists of two passes through the different layers of the network, a forward pass and a backward pass. In the forward pass the input vector is applied to the sensory nodes of the network and its effect propagates through the network layer by layer. The actual response of the network is subtracted from the desired response to produce an error signal. This error signal is then propagated backward through the network against the direction of synaptic conditions. The synaptic weights are adjusted to make the actual response of the network move closer to the desired response.

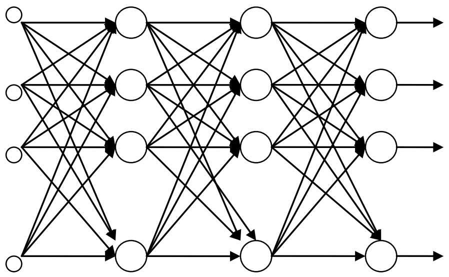


Fig.4. Input layer   Hidden layers   Output layer

The back propagation neural network is essentially a network of simple processing elements working together to produce a complex output. These elements or nodes are arranged into different layers: input, middle and output. The output from a back propagation neural network is computed using a procedure known as the forward pass. The circuit of propagation network is as shown in the Fig.4. The pictorial representation of different steps as follows;

1.The input layer propagates a particular input vector's components to each node in the hidden layers.
2. Hidden layer nodes compute output values, which become inputs to the nodes of the output layer.
3. The output layer nodes compute the network output for the particular input vector. The forward pass produces an output vector for a given input vector based on the current state of the network weights.
4.Since the network weights are initialized to random values, it is unlikely that reasonable outputs will result before training. The weights are adjusted to reduce the error by propagating the output error backward through the network. This process is where the back propagation neural network gets its name and is known as the backward pass:
5. Compute error values for each node in the output layer. This can be computed because the desired output for each node is known.
6. Compute the error for the middle layer nodes. This is done by attributing a portion of the error at each output layer node to the middle layer node, which feed that output node. The amount of error due to each middle layer node depends on the size of the weight assigned to the connection between the two nodes.
7. Adjust the weight values to improve network performance using the Delta rule.
8. Compute the overall error to test network performance.

The training set is repeatedly presented to the network and the weight values are adjusted until the overall error is below a predetermined tolerance. Since the Delta rule follows the path of greatest decent along the error surface, local minima can impede training. The momentum term compensates for this problem to some degree. Neighborhoods from the sub-band images and with the same centre coordinates are represented. This process is repeated for all unknown wavelet coefficients at the highest level. Once all of the wavelet coefficients have been generated, the synthesised image is inverse transformed to give an image that should resemble that of the sample texture. Note that, in order to avoid problems with boundary conditions, it is necessary to pad each sub image with zeros before performing the algorithm. This padding should be removed prior to inverse transform.

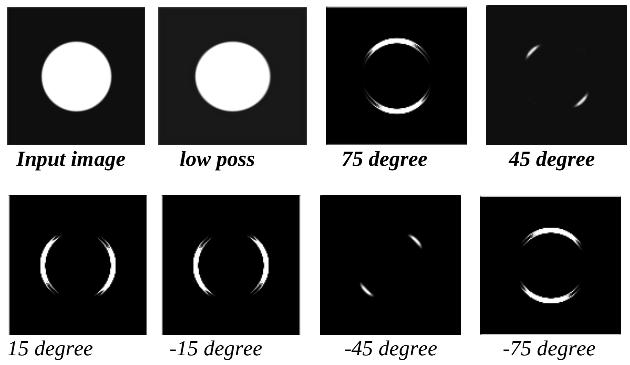


Fig.5 Synthesized image transformation

The unsupervised texture segmentation algorithm, in terms of low computational load is considerable. The Fig.5 of size m * n and the image to be synthesised I_s of size M * N. Let p 2 I_s be a pixel to be synthesised and let w be the width of the neighbourhood of the square spatial neighbourhood surrounding p. For each pixel to be synthesised in algorithm needs to search up to nm locations.

At each of those locations, $4w_2$ operations need to be performed to calculate the weighted sum squared difference. This is $4nmw_2$ operations in total for each searched region. Therefore to generate I_s of size M* N the algorithm will have to perform $4NMnmw^2$ operations. In comparison, the algorithm proposed in this paper synthesises texture at the level of the complex wavelet transform.

At this level the dimensions of the sample image are nm=16 and the image to be synthesised are NM=16. Since all the sub images at this level must be searched, the total number of operations is given as NM=16 /nm=16 /$4w_1^2$ *6. Here w_1 is the neighbourhood size for this process and is typically smaller than the needed.

The multi-resolution compositing technique is to turn into an image which is a combination of an *original* image and a synthetic texture derived from *target*; the combination is controlled by a mask. The algorithm remains the same as before, except that at each iteration of the algorithm, the *original* is composite back into the *noise* image according to the *mask* using that avoids blurring and aliasing.

For nonuniform textures, the texels within a region may vary randomly in orientation and the position and shape of the texels may be perturbed by the end of the algorithm, *noise* will have been converted into a synthetic texture. First, *Match-Histogram* is used to force the intensity distribution of *noise* to match the intensity distribution of *target*. Then an analysis is constructed from the *target* texture. The pyramid representation can be chosen to capture features of various sizes by using the features of various orientations. Then, the *noise* are modified in an iterative manner. At each iterations, a synthesis is constructed from *noise*, and *Match-Histogram* is done on each of the sub-bands of the synthesis and analysis.

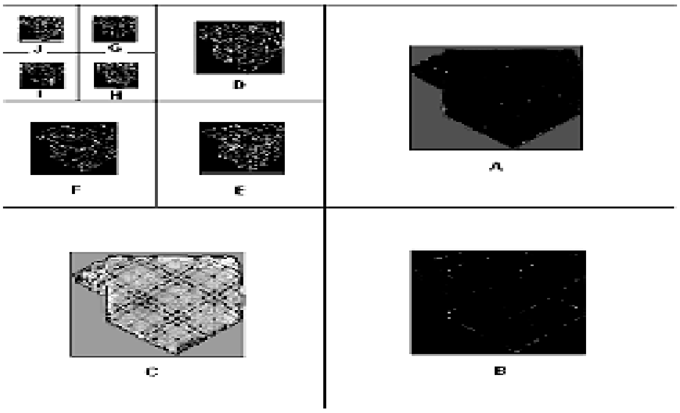


Fig.6 matching of sub-textures.

In the Fig. 6, shows that sub-texture field corresponding to the concrete between the bricks. That show some of the connected components have been used to generate this sub-texture field left of Fig 6. The sub-texture fields are synthesized in preprocess by using any existing texture synthesis technique.

III.RESULTS AND DISCUSSIONS

In this section, the results obtained with the previously described texture synthesis and sub image matching technique. Fig.5 illustrates an example of synthesis result. The top row shows from left to right: the input, the resulting cells and the corresponding mesh. The second row shows the arrangement map and the resulting synthesized mesh, that row is the resulting texture for a low (left) and high (right) amount of classes. Using a high amount of classes causes connected components to be very small, which results in nearly no computed sub textures. The left column represents the model, the middle shows existing techniques from top to bottom: texture quilting the feature matching synthesis technique of pixel the right column the result shows. In all cases, the pre-processing time required only once for a given texture is below m*n this includes segmentation, texture-mesh generation and sub texture field synthesis.

Simulations have been carried out using a database of aquatic images. The proposed scheme for indexing the still texture content is employed on the image database.

IV. CONCLUSIONS

This method of each textured regions are obtained. Simulation results show that the proposed algorithm is effective in texture image retrieval on the bases of texture content. The segmentation and it allows efficient proposed algorithm can be applied to large real-world image database for efficient textured-based image indexing. In our propagation algorithm, features of the segmented regions are calculated as the mean of features of all blocks it contains, features more efficient for indexing purpose can be found later.

BIOGRAPHY

Mr.S.Raja completed his B.Tech degree in Information Technology at Sri Balaji Chokalingam Engineering College, Arni and M.Tech degree in Computer Science and Engineering at Dr.M.G.R University, Chennai. He has presented number of papers in National and International Conferences. His areas of interests are Computer Networks, Data Mining, Java Programming & Database Technologies. He is working as an Assistant Professor in Department of Computer Science and Engineering at Panimalar Institute of Technology, Chennai.

Dr.S.Sankar obtained his B.E Degree in Electrical & Electronics Engineering at Sri Venkateswara College of Engineering, from Madras University and M.E (Power System) Degree from Annamalai University Chidambaram. He has done his Ph.D in the area of FACTS controllers in 2011. His research interests are in the area of FACTS, Electrical Machines, Voltage stability, power quality, Power system security and Power System Analysis.

Dr S.SARAVANAKUMAR has more than 10 years of teaching and research experience. He did his Postgraduate in ME in Computer Science and Engineering at Bharath engineering college,anna university chennai, and Ph.D in Computer Science and Engineering at Bharath University, Chennai. He has guiding a number of research scholars in the area Adhoc Network, ANN, Security in Sensor Networks, Mobile Database and Data Mining under Bharath University Chennai, , Sathayabama University and Bharathiyar University.

References

1. Li-Yi Wei and Marc Levoy. Fast texture synthesis using tree-structured vector quantization. Proceedings of SIGGRAPH 2012, pages 479–488, 2012.

2. Unser, M.: Texture Classification and Segmentation Using Wavelet Frames. IEEE Trans. Image Processing. Vol. 4 (11) (2005), pp. 1549-1560.

3.Fukuda, S., Hirosawa, H.: A Wavelet-based Texture Feature Set Applied to classification of Multifrequency Polarimetric SAR Images. IEEE Trans. Geosci. Remote Sensing. Vol. 37 (2009), pp. 2282-2286

4.B. S. Manjunath and W. Y. Ma, "Texture Features for Browsing and Retrieval of Image Data", IEEE .

5 W. Niblack, "The QBIC Project: Querying Images by Content using Color, Texture and Shape", In SPIE

6.K. Jain and F. Farrokhnia, "Unsupervised Texture Segmentation Using Gabor Filters", Pattern Recognition, Vol. 24, No. 12, 1991, pp. 1167-1186.

7.Y. Rubner and C. Tomasi, “Texture Matrices". Proceedings of IEEE International Conference on Systems, Man and Cybernetics, San-Diego, USA, 1998, pp. 4601-4607.