Principal Component Analysis (PCA) and Fast Independent Component Analysis (Fast ICA) are two commonly used methods for denoising images captured by single-sensor digital cameras with color filter arrays (CFAs).

Vedika Paliwal
and
Durga Prasad Yadav

Abstract- **Denoising of natural images is the fundamental and challenging research problem of Image processing. This problem appears to be very simple however that is not so when considered under practical situations, where the type of noise, amount of noise and the type of images all are variable parameters, and the single algorithm or approach can never be sufficient to achieve satisfactory results. Single-sensor digital color cameras use a process called color demosaicking to produce full color images from the data captured by a color filter array (CFA).Normally the quality of images are degraded because of sensor of camera. In this paper we have developed PCA, FASTICA based algorithm with K-means clustering and compared the both .Performance evaluation is done with PSNR, WPSNR, SSIM, Correlation Coefficient.**

Keywords--*PCA, ICA, Kmean, CFA*

I. INTRODUCTION

Digital images play an important role both in daily life applications such as satellite television, magnetic resonance imaging, computer tomography as well as in areas of research and technology such as geographical information systems and astronomy. Data sets collected by image sensors are generally contaminated by noise. Imperfect instruments, problems with the data acquisition process, and interfering natural phenomena can all degrade the data of interest. Furthermore, noise can be introduced by transmission errors and compression. Thus, denoising is often a necessary and the first step to be taken before the images data is analyzed. It is necessary to apply an efficient denoising technique to compensate for such data corruption. Image denoising still remains a challenge for researchers because noise removal introduces artifacts and causes blurring of the images. This paper describes different methodologies for noise reduction (or denoising) giving an insight as to which algorithm should be used to find the most reliable estimate of the original image data given its degraded version. Noise modeling in images is greatly affected by capturing instruments, data transmission media, image quantization and discrete sources of radiation. Most existing digital color cameras use a single sensor with a color filter array (CFA) [1] to capture visual scenes in color. Since each sensor cell can record only one color value, the other two missing color components at each position need to be interpolated from the available CFA sensor readings to reconstruct the full-color image. The color interpolation process is usually called color demosaicking (CDM).Many CDM algorithms [2]-[5] proposed in the past are based on the unrealistic assumption of noise-free CFA data. The presence of noise in CFA data not only deteriorates the visual quality of captured images, but also often causes serious demosaicking artifacts which can be extremely difficult to remove using a subsequent denoising process. Many advanced denoising algorithms which are designed for monochromatic (or full color) images, are not directly applicable to CFA images due to the underlying mosaic structure of CFAs. Our task is to attempt to reduce the noise inherent in the image. A critical concept in signal processing is signal representation the same image is representation in many ways, and thus an important problem is to determine which one is best to be able to effectively denoise signal. The traditional solution is the Fourier representation, but more modern is nonlinear methods employ the wavelet transforms. The common ground of Fourier and wavelet method is that they use signal representation which are pre-determined i.e. they cannot be directly adapted to the signal structure.

The goal is to introduce a nonlinear de-noising method which adapts the representation used to the statistical structure of data to be de-noised. We concentrate on linear data adaptive transform such as the independent component analysis (ICA) and principle component analysis (PCA).

II. ADAPTIVE LINEAR TRANSFORM

In general non- linear transformation function is

$$\mathbf{S=f(x)} \qquad \mathbf{(1)}$$

Where f is vector valued function of a vector argument and **x, s** denotes random vector. The mapping **f** is a fixed function but often we would like to adapt it to structure of x so that we may capture the information in inherent in it. Such type of transform makes it difficult to adapt it to the data.

So the simplest solution is to restrict f to be linear. Linearity means it should satisfy the following two constraints:

$$\mathbf{f\,(x_1+x_2) = f(x_1) + f(x_2)} \qquad \mathbf{(2)}$$

$$\mathbf{f\,(\alpha x_1)=\alpha\,.\,f\,(x_1)} \qquad \mathbf{(3)}$$

Where α is any scalar, and x1,x2 are arbitrary vectors. Then

$$\mathbf{s= W.\,x} \qquad \mathbf{(4)}$$

Since any linear function can be written as a matrix multiplication, and multiplication by any matrix satisfies the linearity constraints. Thus if W is n by m matrix the transform has been parameterized by nm parameters which can subsequently be adapted to the statistics of x. so that s forms a representation with relevant properties. Linear transforms have many advantages over nonlinear ones. Most important of which is mathematical simplicity with which linear transforms can be analyzed.

III. PRINCIPAL COMPONENT ANALYSIS

PCA is a classical de-correlation technique which has been widely used for dimensionality reduction with direct applications in pattern recognition, data compression and noise reduction. Denote by $x=[x_1\ x_2 \ldots x_m]^T$ an m-component vector variable and denote by

$$\mathbf{X}=\begin{bmatrix} x_1^1 & x_1^2 \ldots\ldots x_1^n \\ x_2^1 & x_2^2 \ldots\ldots x_2^n \\ & \vdots \\ x_m^1 & x_m^2 \ldots\ldots x_m^n \end{bmatrix} \qquad \mathbf{(5)}$$

The sample matrix of **x**, where x_i^j ,j=1,2,……,n, is the discrete sample of variable x_i ,i=1,2,…..,m. The ith row of sample matrix X, denoted by $X_i= [x_i^1 \quad x_i^2 \quad \ldots\ x_i^n]$, is the sample vector of x_i. The mean value of x_i can be estimated as

$$\boldsymbol{\mu_i}\mathbf{=1/n} \sum_{j=1}^{n} \boldsymbol{X_i}\mathbf{(j)} \qquad \mathbf{(6)}$$

and then the sample vector Xi is centralized as

$$\bar{\boldsymbol{X}}_i = \boldsymbol{X_i} - \boldsymbol{\mu_i} = \left[\bar{x}_i^1 \quad \bar{x}_i^2 \ \ldots\ldots \bar{x}_i^n\right] \qquad \mathbf{(7)}$$

Where $\bar{x}_i^j=x_i^j-\mu_i^j$

Accordingly, the centralized matrix of X is

$$\bar{\boldsymbol{X}} = \left[\bar{X}_1^T \quad \bar{X}_2^T \ \ldots\ldots \bar{X}_m^T\right]^T \qquad \mathbf{(8)}$$

Finally, the co-variance matrix of the centralized dataset is calculated as

$$\mathbf{\Omega = 1/n}\ \bar{X}\bar{X}^T \qquad \mathbf{(9)}$$

The goal of PCA is to find an orthonormal transformation matrix P to de-correlate $\bar{X}$, i.e. $\bar{Y}=P\bar{X}$ So that the co-variance matrix of $\bar{Y}$ is diagonal. Since the co-variance matrix **Ω** is symmetrical, it can be written as:

$$\mathbf{\Omega = \Phi\Phi^T} \qquad \mathbf{(10)}$$

Where **Φ= [**$\Phi_1, \Phi 2 \ldots \Phi_m$] is the m×m orthonormal eigenvector matrix and **Λ**=diag { $\lambda_1\ \lambda_2 \ldots \lambda_m$ }is the diagonal eigenvalue matrix with $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_m$. The terms $\Phi_1\ \Phi_2 \ldots \Phi_m$ and $\lambda_1\ \lambda_2 \ldots \lambda_m$ are the eigenvectors and eigenvalues of **Ω**. By setting

$$\mathbf{P = \Phi\Phi^T} \qquad \mathbf{(11)}$$

$\bar{X}$ can be decor related, i.e. $\bar{Y}=P\bar{X}$ and $\Lambda = (1/n)\ \bar{Y}\bar{Y}^T$

An important property of PCA is that it fully de-correlates the original dataset$\bar{X}$. Generally speaking, the energy of a signal will concentrate on a small subset of the PCA transformed dataset, while the energy of noise will evenly spread over the whole dataset. Therefore, the signal and noise can be better distinguished in the PCA domain[1].

IV. INDEPENDENT COMPONENT ANALYSIS

To rigorously define ICA (Jutten and Hérault, 1991; Comon, 1994), we can use a statistical "latent variables" model. Assume that we observe *n* linear mixtures x_1, ...,x_n of *n* independent components

$$x_j = a_{j1}s_j + a_{j2}s_j + \ldots a_{jn}s_n \text{ For all } j \quad \textbf{(12)}$$

We have now dropped the time index t; in the ICA model, we assume that each mixture x_j as well as each independent component s_k is a random variable, instead of a proper time signal. The observed values $x_j(t)$, e.g., the microphone signals in the cocktail party problem, are then a sample of this random variable. Without loss of generality, we can assume that both the mixture variables and the independent components have zero mean: If this is not true, then the observable variables x_i can always be centered by subtracting the sample mean, which makes the model zero-mean. It is convenient to use vector-matrix notation instead of the sums like in the previous equation. Let us denote by **x** the random vector whose elements are the mixtures x_1, ...,x_n, and likewise by **s** the random vector with elementss_1,.., s_n. Let us denote by **A** the matrix with elements $a_{i\,j}$.Generally, bold lower case letters indicate vectors and bold upper-case letters denote matrices. All vectors are understood as column vectors; thus $\mathbf{x}^T$, or the transpose of **x**, is a row vector. Using this vector-matrix notation, the above mixing model is written as

$$\mathbf{x} = \mathbf{A}\mathbf{s}. \quad \textbf{(13)}$$

Sometimes we need the columns of matrix **A**; denoting them by $\mathbf{a}_j$ the model can also be written as

$$\mathbf{X} = \sum_{i=1}^{n} a_i s_i \quad \textbf{(14)}$$

The statistical model is called independent component analysis, or ICA model[6]-[9]. The ICA model is a generative model, which means that it describes how the observed data are generated by a process of mixing the components s_i. The independent components are latent variables, meaning that they cannot be directly observed. Also the mixing matrix is assumed to be unknown. All we observe is the random vector **x**, and we must estimate both **A** and **s** using it. This must be done under as general assumptions as possible.

The starting point for ICA is the very simple assumption that the components s_i are statistically independent. It will be seen below that we must also assume that the independent component must have nongaussian distributions. However, in the basic model we do not assume these distributions known (if they are known, the problem is considerably simplified.) For simplicity, we are also assuming that the unknown mixing matrix is square, but this assumption can be sometimes relaxed. Then, after estimating the matrix **A**, we can compute its inverse, say **W**, and obtain the independent component simply by:

$$\mathbf{S} = \mathbf{W}\,\mathbf{x} \quad \textbf{(15)}$$

I.K mean

k-means clustering is a method of cluster analysis which aims to partition *n* observations into *k* clusters in which each observation belongs to the cluster with the nearest mean. This result into a partitioning of the data space into Voroni Cells. The problem is computationally difficult, however there are efficient heuristic algorithms that are commonly employed and converge fast to a local optimum. These are usually similar to the expectation-maximization algorithm for mixtures of Gaussian distributions via an iterative refinement approach employed by both algorithms.[10]-[11] Additionally, they both use cluster centers to model the data, however *k*-means clustering tends to find clusters of comparable spatial extent, while the expectation-maximization mechanism allows clusters to have different shapes

.

Given a set of observations ($\mathbf{x}_1$, $\mathbf{x}_2$, …, $\mathbf{x}_n$), where each observation is a d-dimensional real vector, k-means clustering aims to partition the n observations into k sets ($k \le n$) $\mathbf{S} = \{S_1, S_2, \ldots, S_k\}$ so as to minimize the within-cluster sum of squares

$$\underset{S}{\boldsymbol{arg}\mathbf{min}} \sum_{i=1}^{k} \sum_{x_j \epsilon s_j} \|\boldsymbol{X_i} - \boldsymbol{\mu_i}\|^2 \tag{16}$$

where $\boldsymbol{\mu}_i$ is the mean of points in S_i.

The most common algorithm uses an iterative refinement technique. Due to its ubiquity it is often called the ***k*-means algorithm**; it is also referred to as LIoyd’s algorithm, particularly in the computer science community.

Given an initial set of k means $\mathbf{m}_1^{(1)},\ldots,\mathbf{m}_k^{(1)}$ the algorithm proceeds by alternating between two steps:
Assign each observation to the cluster with the closest mean

$$\boldsymbol{S_i^{(t)}}=\{\boldsymbol{x_p} : \|\boldsymbol{x_p} - \boldsymbol{m_i^{(t)}}\| \le \|\boldsymbol{x_p} - \boldsymbol{m_i^{(t)}}\| \forall \mathbf{1} \le \boldsymbol{j} \le \boldsymbol{k}\} \tag{17}$$

Where each x_P goes into exactly one$s_t{}^{(t)}$even if it could go in two of them.

$$\boldsymbol{m_i^{(t+1)}}=\frac{\mathbf{1}}{|S_i^{(t)}|} \sum_{x_j \ \epsilon \ s_i^{(t)}} \boldsymbol{X_j} \tag{18}$$

The algorithm is deemed to have converged when the assignments no longer change.

V. EXPERIMENTAL RESULTS:

In this paper we have generated the noise with random sequence and that noise is added to reference image taken by single sensor camera. Our work has been implemented in Matlab graphical user interface. We have calculated different parameters like PSNR, WPSNR, SSIM and correlation coefficient to show the performance of our algorithm.The formulas used for calculation are shown below:

Fig 1.Original image

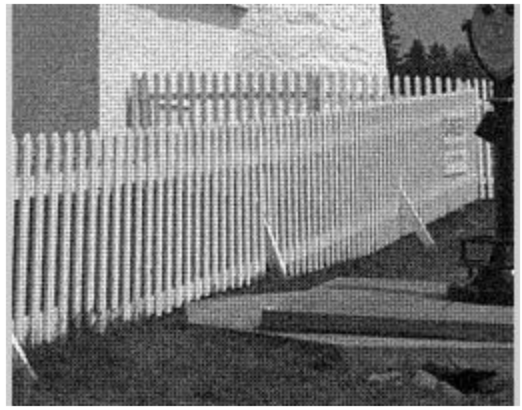

Fig 2. Noisy image

Fig 3.Denoised image

PSNR: The peak signal-to-noise ratio, often abbreviated **PSNR**, is an engineering term for the ratio between the maximum possible power of a signal and the power of corrupting noise that affects the fidelity of its representation. Because many signals have a very wide dynamic range, PSNR is usually expressed in terms of the logarithmic **decibel** scale.

$$\textbf{MSE} = 1/mn \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [I(i,j) - K(i,j)]^2 \tag{19}$$

The PSNR is defined as

$$\textbf{PSNR} = 10 \cdot \log_{10}\left(\frac{MAX_I^2}{MSE}\right)$$

$$= 20 \cdot \log_{10}(MAX_I) - 10 \cdot \log_{10}(MSE) \tag{20}$$

Weighted PSNR: Weighted peak signal to noise ratio (WPSNR) takes into account, the texture of the image and the fact that the human eye is less sensitive to changes in textured areas than in smooth areas. It is nothing but PSNR weighted by an HVS parameter called noise visibility function (NVF).

$$\textbf{WPSNR} = 20 \cdot \log_{10}\left(\frac{255}{\mathrm{NVF} \times \sqrt{\mathrm{MSE}}}\right) \tag{21}$$

The formula to calculate this factor is:

$$\mathbf{NVF = NORM}\left(\frac{1}{1\times\delta_{block}^{2}}\right) \quad \textbf{(22)}$$

where δ block is the standard deviation of luminance of the block of pixels and NORM is the normalization function applied on each block of pixels to normalize the value of NVF in the range from zero to unity. High value of WPSNR indicates that the image is less distorted.

SSIM:Structural similarity index matrix (SSIM) they process all the pixels equally, so they cannot reasonably reflect the Subjective feeling on different scenes based on human visual sensitivity (HVS). Thus, a more comprehensive image quality assessment can be made by considering HVS. The method here is based on structural distortion measurement instead of error measurement. The idea behind this is that the human vision system is highly specialized in extracting structural information from the viewing field and it is not specialized in extracting errors. The structural similarity index correlates with human visual system. Thus SSIM is used as a perceptual image quality evaluation metric. The SSIM is defined as function of luminance (l), contrast(c) and structural components(s) respectively [10].

$$\mathbf{SSIM} = l(x, y), c(x, y), s(x, y) \quad \textbf{(23)}$$

Where,

$$l\,(x, y) = 2\mu x\ \mu y+C1/\mu x2+\ \mu y2+\ C1 \quad \textbf{(24)}$$

$$c\,(x,y) = 2\sigma_x\ \sigma_y+C_2/\ \sigma_x^2+\ \sigma_y^2+\ C_2 \quad \textbf{(25)}$$

$$s(x,\ y) = 2\sigma_{xy}+\ C_3/\ \sigma_x\ \sigma_y+\ C_3 \quad \textbf{(26)}$$

The mean and variance are denoted by μ and σ. The constants C1, C2 and C3 are included to avoid numerical instabilities in the ratios. The SSIM is calculated between two NxN neighbourhoods of original and denoised images. The overall perceptual similarity of an image is obtained by taking mean of SSIM (MSSIM) of several neighbourhoods.

Correlation Coefficient:It is used to compare two images.

$$\mathbf{r} = \frac{\sum_m \sum_n (A_{mn}-\bar{A})(B_{mn}-\bar{B})}{\sqrt{\left(\left(\left(\sum_m \sum_n (A_{mn}-\bar{A})^2\right)\right)\left(\left(\sum_m \sum_n (B_{mn}-\bar{B})^2\right)\right)\right)}} \quad \textbf{(27)}$$

Where, $\bar{A}$ = mean2(A), and $\bar{B}$= mean2(B).

The Correlation Coefficient has the value r=1 if the images are absolutely identical ,r=0 if they are completely uncorrelated and r=-1 if they are completely anti-correlated.
The Table 1 shows the performance comparison of denoising algorithms based on estimated PSNR, WPSNR and SSIM and Correlation Coefficient values of FAST ICA and PCA. The estimated values in Table 1 indicate that the independent component analysis technique is the most effective denoising method compared to PCA.

		PCA				FASTICA			
IMAGES	ORIGNAL SNR	PSNR	CORR COEFF	WPSNR	SSIM	PSNR	CORR COEFF	WPSNR	SSIM
1	26.6987	27.2961	0.9916	38.8532	0.7397	33.0141	0.99771	44.6329	0.8999
2	26.7002	27.4057	0.9836	38.7727	0.7883	33.1462	0.9955	44.7029	0.9239
3	26.6835	27.3481	0.9835	38.8753	0.5981	33.0279	0.9954	44.5311	0.8301
4	26.7002	27.1891	0.9928	38.7607	0.7055	33.0410	0.9980	44.5141	0.8871
5	26.6987	27.2166	0.9915	38.7856	0.8092	32.9141	0.9976	44.3741	0.9250
6	26.6899	27.7531	0.9840	38.7846	0.8907	32.8258	0.9949	44.6122	0.9633
7	26.6834	27.1938	0.9779	38.7634	0.9254	32.1453	0.9928	43.6400	0.9764
8	26.7226	27.0580	0.9831	38.9210	0.6513	33.1550	0.9957	44.8423	0.8703
9	26.6987	27.0928	0.9841	38.7111	0.9630	32.0107	0.9947	41.9592	0.9876
10	26.6899	27.6035	0.9892	38.8047	0.9004	32.7465	0.9966	43.9275	0.9664

Table1: PEROFRMANCE COMPARISION OF DENOISING ALGORITHMS

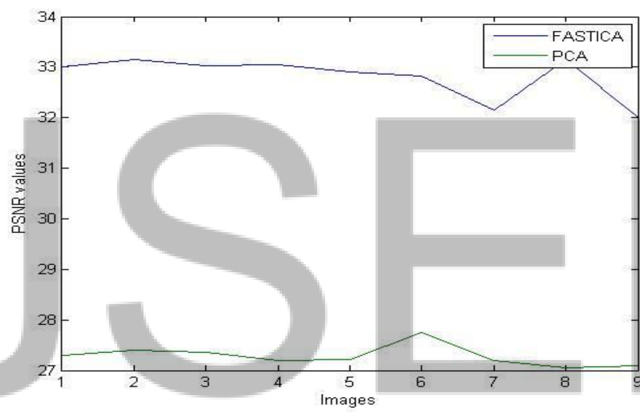

Fig 4. Comparision of PSNR values of FASTICA and PCA denosing algorithm

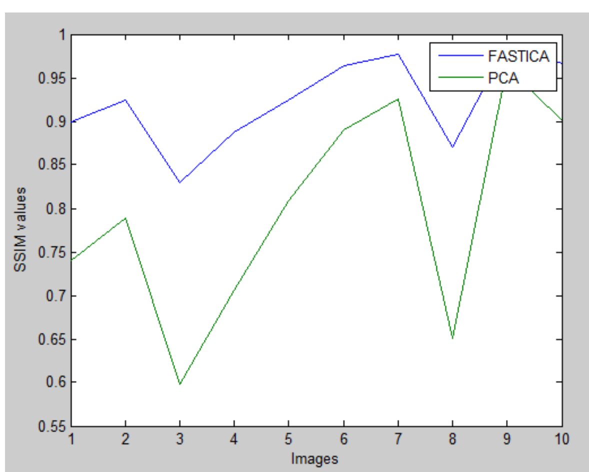

Fig 5. Comparision of SSIM values of FASTICA and PCA denosing algorithm

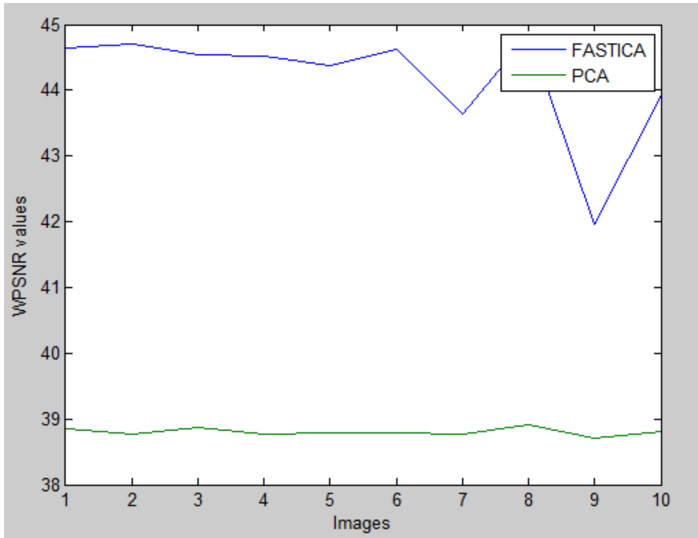

Fig 4. Comparision of WPSNR values of FASTICA and PCA denosing algorithm

VI. CONCLUSION:

This paper presented a PCA and Fast ICA based CFA image denoising scheme for single-sensor digital camera imaging applications. To fully exploit the spatial and spectral correlations of the CFA sensor readings during the denoising process, the mosaic Samples from different color channels were localized by using a supporting window to constitute a vector variable, whose Statistics are calculated to find the PCA and ICA transform matrix. A synthetic noise is also generated so the algorithmic difference can be evaluated. Kmeans Clustering approach has been considered The denoising performance has been evaluated by PSNR, WPSNR and SSIM and Correlation Coefficient values. In comparison with the other methods, the image denoised by ICA has good signal - to - noise ratio and structural similarity index is almost close to that of original image. Thus ICA method provides better quality image which is close to human perception.

REFERENCES

[1] Lei Zhang, Rastislav Lukac, Xiaolin Wu and David Zhang, "PCA-Based Spatially Adaptive Denoising of CFA Images for Single-Sensor Digital Cameras" IEEE transactions on image processing, vol. 18, no. 4, April 2009.

[2] L. Zhang and X. Wu, "Color demosaicking via directional linear minimum mean square-errorestimation," IEEE Trans. Image Process., vol.14, no. 12, pp. 2167–2178, Dec. 2005.

[3] Xiaolin Wu, Dahua Gao, Guangming Shi, and Danhua Liu, "Color demosaicking with sparse representations" Proceedings of 2010 IEEE 17th International Conference on Image Processing September 26-29, 2010, Hong Kong.

[4] K. Hirakawa and T. W. Parks, "Adaptive homogeneity-directed demosaicing algorithm," IEEE Trans. Image Process.ing, vol. 14, no. 3, pp. 360–369, Mar. 2005.

[5] R. Lukac and K. N. Plataniotis, "Color filter arrays: design and performance analysis," IEEE Trans.Consum. Electron., vol. 51, no. 11, pp. 1260–1267, Nov. 2005.

[6] Liujicheng ,zhangyi "Image Denoising based on Independent Component Analysis" International Conference on Intelligent Human-Machine Systems and Cybernetics 2009 .

[7] Megha.P.Arakeri1, G.Ram Mohana Reddy , “A Comparative Performance Evaluation of Independent Component Analysis inMedical Image Denoising” IEEE-International Conference on Recent Trends in Information Technology, ICRTIT 2011978-1-4577-0590-8/11/$26.00 ©2011 IEEE MIT, Anna University, Chennai. June 3-5, 2011.

[8] P. Gruber , F.J. Theis ,A. M. Tomie ,E.W. Lang “Automatic denoising using local independent component analysis” EIS 2004 Island of Madeira, Portugal.

[9] Hong-yan Li Guang-long Ren Bao-jin Xiao “Image Denoising Algorithm Based on Independent Component Analysis” World Congress on Software Engineering，2009.

[10] Radha Chitta, Rong Jin, Timothy C. Havens, Anil K. Jain Approximate Kernel k-means: Solution to Large Scale Kernel Clustering” KDD’11, August 21–24, 2011, San Diego, California, USA.

[11] T. Hitendra Sarma and P. Viswanath,B. Eswara Reddy A Fast Approximate Kernel k-means ClusteringMethod For Large Data sets”. 978-1-4244-9477-4/©2011 IEEE.