Egyptian Computer Science Journal (ISSN: 1110 – 2586)
Volume 40 – Issue 03, September 2016

An Improved Image Denoising Model by Incrementally Learning Gaussian Mixtures Parameters

Mohammad Hashem[*1], Essam Rashed[#2], Fatma Harby Mohamed[#3]
[*]Computer Science Department, Faculty of Computer & Information Sciences,
Ain Shams University, Cairo, Egypt.
[#]Department of Mathematics, Faculty of Science, Suez Canal University, Ismailia, Egypt.
Emails: [1]mhashem100@yahoo.com, [2]erashed@science.suez.edu.eg, [3]fatma_harby@yahoo.com

Abstract

Patch-based prior learning algorithm is capable of delivering state-of-the-art performance in image denoising. The major concern that affects the patch-based restoration algorithms is the accuracy of the patch priors. Our work is based on the extension of the Gaussian Mixture Model (GMM) estimation which is developed to iteratively capture and process data incrementally. In this paper, we introduce an image denoising framework that uses patch prior learned online refines Gaussian mixture model incrementally. This model decides if we can merge two given Gaussians without drifting from the real data distribution. An incrementally learning mixture model helps to reduce the complexity of the model while still giving a precise description of the observations. The results show that the proposed method generally achieves comparable performance to the conventional approach, while producing models at lower training time without any compromise with the model accuracy.

Keywords: *Gaussian mixture model, Image denoising, Expectation Maximization, Incremental learning*.

1. Introduction

Image denoising is a significant task in the digital image processing applications, which requires a robust method to accomplish this task. The goal of image denoising is to remove imperfections caused by poor image sensors or artifacts which appear during data acquisition process [1]. The challenge of most image denoising approaches is to recover images contaminated by noise while producing sharp images without losing fine details or obtaining blurring edges [2]. In the last decade, software-based denoising approaches had become more popular and widely applicable. These methods had been proposed and a considerable improvement in denoising performance was achieved. Most denoising approaches which used techniques such as the Lucy-Richardson algorithm, Wiener filtering, and least-squares are always generating unwanted artifacts which makes it hard to determine whether features learned by such models represent a true property of natural images or not [25]. Levin *et al*. [4] introduced an approach by incorporating priors derived from natural image statistics. However, learning whole image distribution is a tremendous computational challenge because images are almost of high dimensions, non-Gaussian and continuous signals.

Due to the computational difficulty, tasks such as learning priors and likelihood estimation are set to use small image patches rather than working with whole images directly. So, instead of processing each image pixel individually, the block-based (or patch-based) approaches have been introduced [5].

Recently, a new denoising approach is proposed based on image internal self-similarity prior and external patch priors [3]. Similar patches from clean images are clustered and subspaces are learned by utilizing (GMMs). Finally, the learned GMMs are used to guide the clustering of noisy patches followed by a low-rank approximation process to estimate the latent subspace for image recovery [3]. In [6], an improved image denoising framework based on spatially constraining GMM prior for the image patches is proposed. By assuming the same multivariate normal distribution for underlying similar patches in a neighborhood, the weights are averaged for the pixel estimations based on the similarity of the estimated patches to their corresponding clusters.

Image prior plays a critical role in image denoising problem. Simplicity, Gaussian mixture prior is popularly used as it is easy to be learned and to perform denoising by finding the Maximum a posterior (MAP) estimate of a corrupted patch [7]. In the proposed framework, a probabilistic approach via GMM has been learned to develop a filter where the parameters are computed using trained image patches. The goal is to find a recovered image in which every patch is likely under the modelled prior while keeping the denoised image still close to the true one.

2. Gaussian Mixture Prior

Stochastic models such as mixture models, graphical models, Markov random fields and hidden Markov models have played a key role for the statistics of natural image patches. Recently, there has been a growing interest in comparing various models for natural images such as log-likelihood and multi-information reduction performance, and much progress has been achieved [8]. The GMM model has shown interesting results due to its simplicity and ubiquity. It uses a small number of mixture components learned from image patches to model image statistics. Hence, a mixture model consisting of a number of densities can be used to construct a satisfactory model to discriminate between different classes. The most important class of finite mixture models is Gaussian mixtures [9]. The reason for the importance and widespread use of Gaussian mixtures is its simplicity and its concise representation requiring only two parameters: mean and variance. Also, Gaussian density is symmetric, and assumes least prior knowledge in estimating an unknown probability density [10]. These characteristics of the Gaussian distribution along with its well-studied status give Gaussian mixtures the power and effectiveness over other mixture densities. By assuming that an image consists of a set of clusters, mixtures of Gaussian functions obviously are well suited to model these clusters. The GMM is a multidimensional probability density function (PDF). It is the sum of Gaussians each has its mean (location), covariance (shape), and a probabilistic assignment of every data point to the Gaussians.

Maximum-likelihood estimation has been one of the most widely used approaches for estimating the parameters of mixture models [11]. Gaussian mixture model is a parametric PDF represented as a weighted sum of Gaussian component densities, while the model parameters are usually estimated from well-trained prior information using the expectation-maximization (EM) algorithm. The EM algorithm is an efficient iterative procedure to find maximum likelihood or (MAP) estimation [10]. The log-likelihood should be evaluated before maximum likelihood estimation. The EM algorithm has number of properties that rank it as an attractive algorithm for mixture models analysis. It satisfies the probabilistic constraints without the need to set a learning rate. One major drawback of the EM algorithm is that it converges slowly, especially when mixture components are not well separated. The EM algorithm seeks to find the Maximum Likelihood Estimate (MLE) in the presence of missing or hidden data by iteratively applying the following two steps: The Expectation step (E-step), and Maximization step (M-step). In the E-step, the missing data are estimated given the observed data and current estimate of the model parameters. That creates a function for the expectation of the log-likelihood evaluated using the current estimate for the parameters. In the M-step, the likelihood function is maximized under the assumption that the missing data are known [12]. Parameters maximizing the expected log-likelihood found on the E-step are computed and then used to determine the distribution of the latent variables in the next E-step.

Several clustering methods are based on a distance or dissimilarity measures. However, clustering analysis using mixture models has become an effective and powerful approach. In the proposed method, we introduced clustering the patches via learning a finite Gaussian mixture model over the pixels of natural image patches. The GMM model is learned without any constraints on the model, learning is performed on the means, full covariance matrices and mixing weights, over all pixels [13]. In the problem of patch denoising, the model of image corruption is supposed to have the following formulation:

$$y = Ax + w, \tag{1}$$

where y represents a corrupted image, x is a true image, A is a corruption matrix and w is an additive white Gaussian noise with known standard deviation. The work presented here is based on an optimization algorithm that maximizes expected patch log-likelihood (EPLL) framework [7] using GMM prior for image restoration. The EPLL achieved superior results rather than the previous methods which have used the same prior.

3. Related Work

In many practical applications, the collection and analysis of training data is costly and time consuming. Consequently, the performance of a generative model like GMM depends heavily on the availability of an adequate amount of representative training data to estimate its parameters. In practice, data for training GMM is often limited, and may over time no longer be representative of the underlying data distribution. In static environments, where the underlying data distribution remains fixed, designing a GMM with a limited number of training observations may significantly decrease performance. This is also the case when new information emerges in dynamically changing environments, where underlying data distribution varies or drifts in time.

Since limited training data are typically employed in practice and underlying data distribution is susceptible to change, a system based on GMMs should allow for adaptation in response to new training data from the operational environment or other source. The ability to efficiently adapt GMM parameters in response to newly acquired training data, through incremental learning, is therefore an undisputed asset for sustaining a high level of performance. Indeed, refining a GMM to novelty encountered in the environment may reduce its uncertainty with respect to the underlying data distribution. With incremental learning, GMM parameters should be efficiently updated from new data without requiring access to the previously learned training data. In addition, parameters should be updated without corrupting previously acquired knowledge.

One of the main challenges of the incremental learning of GMMs is the model complexity selection which is required to be dynamic due to the nature of the incremental learning framework. Apparently, there is no increase in the number of Gaussian components when a single novel point arrives while all information available at any time is the current GMM estimate. Another closely related difficulty lies in the order in which new data arrive. If successive data points are always badly correlated, then a large amount of data has to be kept in memory until accurate model order update is achieved [18]. Incremental fitting of GMMs has already been addressed in the machine learning literature. Hall *et al.* [19] merged Gaussian components in a pair-wise manner by considering volumes of the corresponding hyperellipsoids. In [20], the Gaussians are assumed to be grouped using the Chernoff bound to detect overlapping Gaussians. Then different thresholds on this bound are tested and the most likely result is kept as the simplified GMM, but this method is too slow for an on-line process. Hicks *et al.* [20] proposed to "concatenate" two GMMs firstly and then the optimal model sorted by considering models of all low complexities and the one that gives the largest penalized log-likelihood is finally chosen. Verbeek et al. [21] gradually increased the number of components in a GMM by using a greedy algorithm. A heuristic for searching for the optimal component is used to insert mixture components into the mixture one after the other.
Both Yangd [22] and Figueiredo [23] assume that a GMM that contains too many components has been fitted to the data, and the number of components is reduced by discarding "weak" ones. Both methods achieved a considerable increase in efficiency over standard EM, but they require access to past data.

An incremental updating of the density model is performed in [18] using no historical data and the consecutive data are assumed to be varied smoothly. In this method, the current GMM estimation, and a previous GMM of the same complexity after which no model updating has been done. By comparing the current GMM with the historical one, it is determined if new Gaussians are generated or some Gaussians are merged together. This algorithm fails when new data are well explained by the historical GMM, and when consecutive data infract the condition of smooth variation.

4. Expected Patch Log Likelihood Framework-EPLL

Under a predefined patch prior p, EPLL aims to minimize the following cost in order to find the reconstructed image [9]:

$$f_p(x|y) = \frac{\lambda}{2}\,\|Ax - y\|^2 - EPLL_p(X), \tag{2}$$

where λ is a tuning parameter related to the noise variance $\frac{1}{\sigma^2}$. Assuming a general image corruption model to keep the restored image *x* close to the corrupted image *y*, with the constraint that $\|Ax - y\|^2$. In image denoising problem matrix *A* is set to the identity.
The Expected Patch Log Likelihood to be maximized under prior *p* is defined as:

$$EPLL_p(x) = \sum_{i=1}^{N} log_p\,(P_i x) \quad , \tag{3}$$

where $P_i x$ is a mask which extracts the *i-th* patch from image *x* that organized as vectors and $log_p(P_i x)$ is the likelihood of the *i-th* patch under the prior *p*. The Expected Patch Log Likelihood $EPLL_p(x)$ is not the log probability of a full image, but it is the sum of log likelihood of all the *N* patches.

Optimize the cost function directly is intractable, so an alternative optimization method called Half Quadratic Split [14] has been introduced. It defines z^i as a set of per-patch auxiliary variables means that it is a set of patches, one for each overlapping patch $P_i x$ in the original image *x* and β is set to $\frac{1}{\sigma^2}$. Now the new cost function becomes:

$$C_{p,\beta}(x,\{z^i\}|y) = \frac{\lambda}{2}\|Ax - y\|^2 + \sum_i \frac{\beta}{2}\left(\left\|P_i x - z^i\right\|^2 - log\;\; p(z^i)\right) \tag{4}$$

The EPLL optimization involves two steps [7]:
(1) Solving for *x* given $\{z^i\}$ that yields :

$$\widehat{x} = \left(\lambda A^T A + \beta \sum_j P_j^T P_j\right)^{-1} \left(\lambda A^T y + \beta \sum_j P_j^T z^j\right), \tag{5}$$

which means that the solution for *x* at each optimization step is just a weighted average between the noisy image *y* and the average of pixels as they appear in the auxiliary overlapping patches.

(2) Solving for $\{z^i\}$ given x, this is dependent on the prior and involves estimation a MAP of the most likely patch under the prior , for each iteration value of β is kept constant.

$$z^i = argmin_z \frac{\beta}{2}\left(\left\|z^i - P_i x\right\|^2\right) - log\ p(z^i) \tag{6}$$

The solution for z^i is just a MAP estimate with prior p and noise level $\sqrt{(1/\beta)}$. For each β, the current estimated image is averaged with the noisy one and obtaining a new set of z^i patches, solving for them and then obtaining a new estimate for patch *x*.

5. Learn Patch Priors with GMM

The GMM is learned over small patches, the GMM prior outperforms other priors in both patch and whole image restoration. The GMM is based on the Bayesian and it is considered a flexible and powerful statistical modeling tool for multivariate data. The main advantage of the GMM that its simplicity to implement and that it requires a small number of parameters. GMM describes natural image patches with a mixture of Gaussian distributions which is a combination of pdf's. The probability density function is defined as:

$$p(x|\theta) = \sum_{k=1}^{K} \pi_k\ N\ (x|\mu_k, \Sigma_k)\ , \tag{7}$$

where $\theta = \{\pi_k, \mu_k, \Sigma_k\}_{k=1}^{K}$ is the parameter set of GMM, here π_k are mixing weights for each mixture satisfy $0 \leq \pi_k \leq 1$, μ_k is the mean matrix , Σ_k is the covariance matrix of mixture component and *k* is the number of mixture components. Each Gaussian distribution is called a component of the mixture which has its mean and covariance [15]. The Gaussian distribution of the component *k* is defined as follows:

$$N(x|\mu_k, \Sigma_k) = \frac{exp\left(-\frac{1}{2}(x-\mu_k)^T \Sigma_k{}^{-1}(x-\mu_k)\right)}{(2\pi)^{n/2}|\Sigma_k|^{1/2}} \tag{8}$$

The estimation of the component density parameters is carried out by EM algorithm which is employed to maximize the log-likelihood of the given data set. For a given patch, the log-likelihood function of the standard GMM is given by:

$$\log\ p(x) = log(\ \sum_{k=1}^{K} \pi_k\ N(x|\mu_k, \Sigma_k)) \tag{9}$$

Patch de-noising is performed using the approximate MAP procedure described as follows:

1.For each noisy patch *x*, calculate log PDF of Gaussian with zero mean for each component, we call it PYZ_k .

2. Choose the component with the highest log probability $k_{max} = max_k\ (PYZ_k)$.

3. Approximate MAP estimate $\hat{y}$ solution for the k_{max} –*th* component as follows: $\hat{y} = \left(\Sigma_{k_{max}} + \sigma^2 I\right)^{-1}\left(\Sigma_{k_{max}} x\right)$

6. Experimental Results

In the presented work, we have used GMM with few prior assumptions, as the number of components in the mixture and the size of learned patches. In all experiments, λ is set to $N\ /\ \sigma$^2 where *N* is the number of pixels in each patch. We evaluated the GMM by varying the number of mixture components and the size of the image patch. Learning patches is performed using an incremental EM algorithm [17]. For the proposed GMM model, the values for β were set to $M/\sqrt{\sigma}$.[1, 4, 8, 16, 32, 64], where *M* is the patch size. Each model trained on 6x6 pixels of overlapped patches and 8x8 pixels of non-overlapped patches. These patches are randomly sampled from chest screening CT from JAMIT medical image database [16] and DC component of all patches is removed.

6.1. Analysis of Results

We compare the denoising performance of the EPLL-GMM model trained with different number of components and different patch size. Our goal is to evaluate both the number of components and size of the patch used in training in dance performance. An independent white Gaussian noise with known standard deviation is added to each test image. The performance of our approach is experimentally verified on a variety of images and noise levels. The main goal of our work is to demonstrate that a GMM with a relatively small number of components is able to give better results than the original model with lower training time. Here we evaluate the GMM by varying the number of components and the size of the image patch. First, the proposed GMM model is trained with 50 mixture components from a set of $2x10^5$ patches sampled from CT image dataset [16], then we increase the number of components to be 100 mixture. We trained GMMs with unconstrained means and full covariance matrices using EM. Training is performed by incrementally learning a Gaussian mixture model using the means of a merge and/or split operations that combine similar components of GMM. In this method Deleclerq and Piater [17] assumed that each data is a Gaussian with a predefined covariance and a subset of the data can be replaced by a single component to reduce the complexity of the model while still giving a precise description of the observations. An independent white Gaussian noise with known standard deviation is added to each of the test images. We calculate the MAP estimate $\hat{y}$ of each model given the noisy patches. For those models where the MAP estimate does not have a closed form. We have used a numerical approximation as in section 5.

6.1.1. Performance Metrics

The performance of each model was measured using Peak Signal to Noise Ratio (PSNR):

$$PSNR = 20\ log_{10}\left(\frac{1}{\|x-y\|^2}\right), \tag{10}$$

where x is the original image and y is the recovered image. Training the proposed model used a set of 5000 patches (with 50 mixture components) takes around one hour on Intel (R) core™ i5, CPU @ 2.5 GHz, while taking approximately 6h for training by the original EM algorithm.

Table 1 shows the results obtained with the proposed EPLL-GMM compared with results obtained by training 10000 patches (with 100 mixture components) using original EPLL-GMM. The results in Table 1 are obtained by denoising chest CT from [16]. Under three noise levels, the PSNR values in Table I show that the proposed method leads to an improvement in average with different cases. Table 1 demonstrates the differences in PSNR values when small patch size trained, results show that trained 6x6 patch also achieved good results compared with the original EPLL-GMM.

Visually, the difference is not noticeable, so another image quality metric should be used to qualify an image. Contrast to noise ratio (CNR) is used as a measure instead of image contrast in order to have a better measure of the image quality. A higher CNR value is necessary to distinguish among different tissue types, and in particular between a healthy tissue and a pathological tissue. The CNR value evaluates the ratio of the contrast of a target structure in the image and the standard deviation of statistical noise. The CNR between two tissues is defined in terms of their noise respective signal noise-to-ratios of the two tissues:

$$CNR_{AB} = |S_A - S_B|/\sigma_N, \tag{11}$$

where CNR_{AB} is the relationship between pixels intensity differences S_A and S_B of tissues A and B , σ_N is the standard deviation of the noise.

With 100 EM iterations and 5000 patches of size 8×8, the training time with the incremental GMM takes about 204.203 seconds, while using 6×6 patches takes about 187.381 seconds and done with 85 EM iterations only. The log likelihood of the GMM prior learned from 6×6 patches and 8×8 patches using the incremental GMM converges faster using 6×6 patches, although GMM prior gives higher likelihood using 8×8 patches.

Table 1. Values of psnr (db) of denoising based original EPLL-GMM compared with the proposed method for different patients at various noise variance.

Image		Original EPLL-GMM (10000 patch with 100 mixture)	Proposed EPLL-GMM (5000 patch with 50 mixture)	Proposed EPLL-GMM (10000 patch with 50 mixture)
Patient (1) Pulmonary Tuberculosis	**σ** =15	32.89	32.65	32.55
	σ =25	30.32	30.69	30.38
	σ =50	26.60	27.99	27.41
Patient (2) Hypersensitivity Inflammation	**σ** =15	29.21	29.77	29.69
	σ =25	27.25	27.67	27.56
	σ =50	24.77	25.14	25.00
Patient (3) Cancer (adenocarcinoma)	**σ** =15	30.20	30.55	30.58
	σ =25	27.96	28.35	28.28
	σ =50	25.00	25.40	25.52

(a) Training using 8×8 patches

Image		Original EPLL-GMM (10000 patch with 100 mixture)	Proposed EPLL-GMM (5000 patch with 50 mixture)	Proposed EPLL-GMM (10000 patch with 50 mixture)
Patient (4) Calcification	**σ** =15	30.52	30.71	30.57
	σ =25	27.92	28.23	28.35
	σ =50	24.69	25.05	25.06
Patient (5) Chronic Inflammation	**σ** =15	31.78	31.43	31.39
	σ =25	29.57	29.45	29.28
	σ =50	26.41	26.71	26.59
Patient (6) Inflammation	**σ** =15	29.14	29.74	29.74
	σ =25	27.18	27.52	27.60
	σ =50	24.33	24.62	24.70

(b) Training using 6×6 patches

The denoising results demonstrate that proposed method is exceeding the original EPLL, both visually and quantitatively. From Fig 1 we can note how details are much better preserved in our method when compared to original EPLL-GMM. In Fig 1 the chest CT image is taken as an input image and Gaussian noise with **σ** =25 is added then we calculate PSNR and CNR value for the denoised images. From that, results clearly indicate that proposed method trained using large mixture components having highest PSNR and CNR values. In Fig 2, the red square region shows how the proposed method provides a higher resolution image with clearer image details than the original EPLL-GMM method.

As we carry out a comprehensive empirical evaluation of the performance of these algorithms in terms of accuracy and running times. The results reveal that denoising using smaller patch size reduces the running time. The experiments are evaluated with Matlab implementation, running time of denoising based 8×8 patch size takes about 101.652s while using 6×6 patch size takes about 51.259s, while not affecting the denoising quality. The performance of the proposed image denoising approach also evaluated quantitatively based on the original and the denoised scene images

One of the widely used methods to compute the correlation between two images is similarity measurements such as Normalized Cross Correlation (*NCC*) [24] which is calculated by:

$$NCC = \frac{\sum_{m=0}^{M-1}\sum_{n=0}^{N-1} x(m,n)\, y(m,n)}{\sum_{m=0}^{M-1}\sum_{n=0}^{N-1} x(m,n)^2} \quad , \qquad (12)$$

where $x(m, n)$ is the original image and $y(m, n)$ is the restored image. If the normalized cross correlation tends to 1, then the image quality is seemed to be better.

Normalized absolute error (*NAE*) [24] is a criterion to evaluate the ability of preserving the information of the original image where larger value of NAE means a poor quality of the image. It is defined as follows:

$$NAE = \frac{\sum_{m=0}^{M-1}\sum_{n=0}^{N-1} |x(m,n) - y(m,n)|}{\sum_{m=0}^{M-1}\sum_{n=0}^{N-1} |x(m,n)|} \qquad (13)$$

In order to verify the reliability of our proposed approach, all test samples were degraded with Gaussian white noise of different levels 15, 25 and 50 respectively. The experimental results of denoising with the proposed and the original method are assessed and computed using NAE and NCC. Table 2 exhibits the experiment results which prove that the proposed method results achieved less NAE values compared with the original method. The experiments in Table 2 are performed using 5000 patches with 100 mixture components. Also, Table 2 shows that the proposed method achieves higher NCC values than the original EPLL-GMM even if the 8×8 or 6×6 patch size is used.

Table 2. Comparison of NCC and NAE results of proposed method and original method at different noise levels

Image		**Original EPLL-GMM**		**Proposed EPLL-GMM**	
		NCC	**NAE**	**NCC**	**NAE**
Patient (1) Pulmonary Tuberculosis	σ =15	0.9975	0.0437	0.9980	0.0399
	σ =25	0.9958	0.0537	0.9965	0.0508
	σ =50	0.9938	0.0737	0.9913	0.0727
Patient (2) Hypersensitivity Inflammation	σ =15	0.9973	0.0426	0.9981	0.0397
	σ =25	0.9962	0.0532	0.9967	0.0510
	σ =50	0.9935	0.0723	0.9919	0.0717
Patient (3) Cancer (adenocarcinoma)	σ =15	0.9978	0.0449	0.9980	0.0410
	σ =25	0.9962	0.0558	0.9665	0.0533
	σ =50	0.9923	0.0770	0.9943	0.0759

(a) Training using 8×8 patches

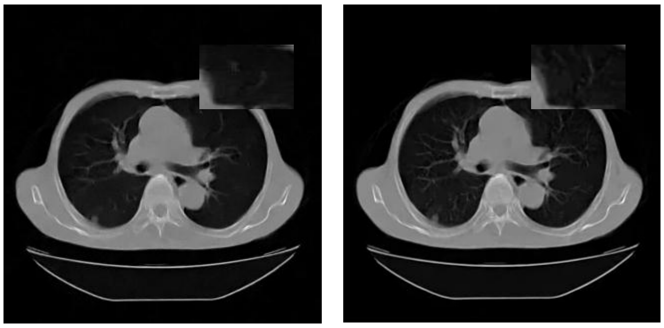
Image		**Original EPLL-GMM**		**Proposed EPLL-GMM**	
		NCC	**NAE**	**NCC**	**NAE**
Patient (4) Calcification	σ =15	0.9987	0.0339	0.9982	0.0323
	σ =25	0.9971	0.0421	0.9971	0.0411
	σ =50	0.9947	0.0587	0.9953	0.0602
Patient (5) Chronic Inflammation	σ =15	0.9983	0.0349	0.9943	.0622
	σ =25	0.9968	0.0439	0.9975	0.0426
	σ =50	0.9940	0.0621	0.9442	0.0622
Patient (6) Inflammation	σ =15	0.9984	0.0351	0.9986	0.0327
	σ =25	0.9970	0.0435	0.9970	0.0426
	σ =50	0.9940	0.0615	0.9943	0.0623

(b) Training using 6×6 patches

(a) Original

(b) Noisy image

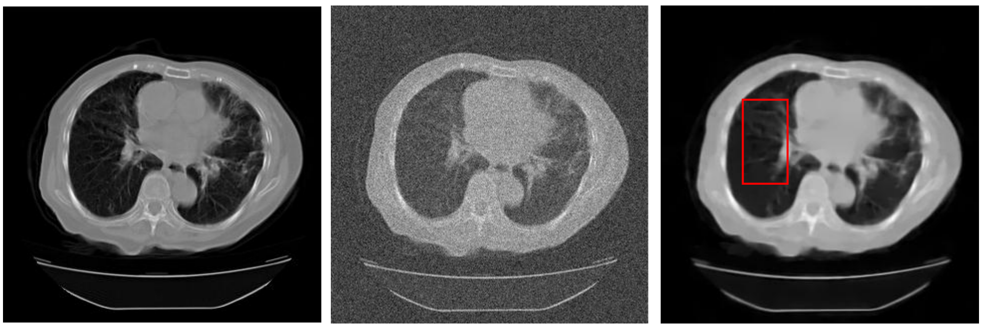
(C) Denoising using original EPLL- GMM with PSNR=24.24, CNR =15.5



(d) Denoising using proposed method trained using 5000 patches with 50 mixture components, PSNR=30.68, CNR=20.49

(e) Denoising using proposed method trained using 5000 patches with 100 mixture components, PSNR= 33.33, CNR=26.49

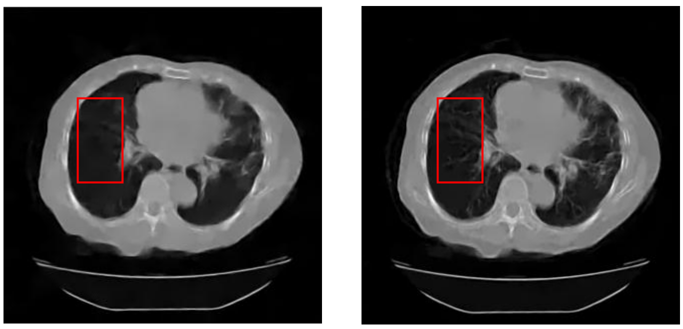
Figure 1. Denoising results using original EPLL-GMM compared with the proposed method using PSNR and CNR as image quality measurements.



(a) Original image

(b) Noisy image

(C) Denoising using the original EPLL with PSNR=24.63, CNR =15.58



(d) Denoising using proposed method trained using 5000 patches with 50 mixture components, PSNR=26.96, CNR=20.28

(e) Denoising using proposed method trained using 5000 patches with 100 mixture components, PSNR= 29.24, CNR=24.32

Figure 2. Visual quality comparison for denoising an image corrupted by Gaussian noise, $\boldsymbol{\sigma}$ =50.

7. Conclusion

In this paper, we proposed a new framework for the whole image denoising based patch models using incrementally learning a Gaussian mixture model based on a new criterion for splitting and merging mixture components. We also optimized the initialization parameters to enhance the performance under different noise levels. Our model performed better denoising results and it achieved an improvement in image quality using fewer training time than conventional. Future work will thus be devoted to improvements by incorporating some artifacts and different noise type.

References

1. S.A. Ali, S. Vathsal and K.L. Kishore. "An Efficient Denoising Technique for CT Images using Window based Multi-Wavelet Transformation and Thresholding". Eur. J. Sci. Res., vol. 48, pp. 315-325, 2010.
2. P. Chatterjee and P. Milanfar. "Patch-based Near-Optimal Image Denoising", IEEE Transactions on Image Processing, vol. 21, pp. 1635-1649, 2012.
3. Chen, Fei, Lei Zhang, and Huimin Yu. "External Patch Prior Guided Internal Clustering for Image Denoising", In Proceedings of the IEEE International Conference on Computer Vision, pp. 603-611, 2015.
4. A Levin, R. Fergus, F. Durand and W.T. Freeman. "Image and depth from a conventional camera with a coded aperture", ACM Transactions on Graphics, vol. 26, pp. 70:1-70:9, 2007.
5. Y. Hel-Or and D. Shaked. "A discriminative approach for wavelet denoising", IEEE Transactions on Image Processing, vol. 17, pp. 443-457,2008.
6. M. Niknejad, H. Rabbani and M. Babaie-Zadeh. "Image restoration using Gaussian mixture models with spatially constrained patch clustering", IEEE Trans. Image Process., vol. 24, no. 11, pp. 3624-3636, 2015.
7. D. Zoran and Y. Weiss. "From learning models of natural image patches to whole image restoration", In Proceedings of IEEE International Conference on Computer Vision, pp. 479-486, 2011.
8. A. Srivastava, A.B. Lee, E.P. Simoncelli and S.-C. Zhu. "On advances in statistical modelling of natural images", Journal of Mathematical Imaging and Vision, vol. 18, pp. 17-33, 2003.
9. J. Z. Feng, L. Song, X. K. Yang and W. J. Zhang. "Image restoration via efficient Gaussian mixture model learning". In Proceedings of IEEE International Conference on Image Processing, pp. 1056-1060, 2013.
10. V. Hasselblad. "Estimation of parameters for a mixture of normal distributions". Technometrics, vol. 8, pp. 431-444, 1966.
11. A.P. Dempster, N.M. Laird and D.B. Rubin. "Maximum Likelihood from Incomplete Data via the EM Algorithm". Journal of the Royal Statistical Society. Series B (Methodological), vol. 39, pp.1-38, 1977.
12. G. Celeux, S. Chr`etien, F. Forbes and A. Mkhadri. "A component wise EM algorithm for mixtures", Journal of Computational and Graphical Statistics, vol. 10, pp. 697-712, 2001.
13. F. -X. Jollois and M. Nadif. "Speed-up for the expectation-maximization algorithm for clustering categorical data", Journal of Global Optimization, vol. 37, pp. 513-525, 2007.
14. D. Geman and C. Yang. "Nonlinear image recovery with half-quadratic regularization", IEEE Transactions on Image Processing, vol. 4, pp. 932- 946, 2002.
15. D.M. Titterington, A.F.M. Smith and U.E. Makov."Statistical Analysis of Finite Mixture Distributions", Hoboken, NJ: Wiley, 1985.
16. Chest Screeing CT dataset, JAMIT medical image database the Japanese Society of Medical Imaging Technology (http://www.jamit.jp/cadcommitte/caddbinfo).
17. Declercq and J.H. Piater. "Online learning of gaussian mixture models: a two-level approach", In Intl. Conf. Comp. Vis., Imaging and Comp. Graph. Theory and Applications, pp. 605-611, 2008.

18. O. Arandjelovic and R. Cipolla. “Incremental learning of temporally-coherent gaussian mixture model”, In Proc. British Machine Vision Conference (BMVC), 2005.
19. P. Hall, D. Marshall, and R. Martin. “Merging and splitting eigenspace models”. PAMI, 2000.
20. Y. A. Hicks, P. M. Hall, and A. D. Marshall. “A method to add Hidden Markov Models with application to learning articulated motion”, BMVC, 2003.
21. J. Verbeek, N. Vlassis, and B. Krose. “Efficient greedy learning of Gaussian mixtures”, Pattern Recognition, 2002.
22. Z. R. Yang and M. Zwolinski. “Mutual information theory for adaptive mixture models”, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 23, no. 4, pp. 396-403, 2001.
23. M. Figueiredo and A. K. Jain. “Unsupervised learning of finite mixture models”, IEEE Transaction on Pattern Analysis and Machine Intelligence, vol. 24, no. 3, pp. 381-396, March 2002.
24. Baviskar A., Ashtekar S., and Chinta war A. “Performance evaluation of high quality image compression techniques." In Advances in Computing, Communications and Informatics (ICACCI, 2014 International Conference on, pp. 1986-1990. IEEE, 2014.
25. N. Joshi, C.L. Zitnick, and R. Szeliski. “Image deblurring and denoising using color priors”, In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 1550-1557, 2009.