Image Restoration Using Particle Filters By Improving The Scale Of Texture WithMRF

Dr. Anna Saro Vijendran *saroviji@rediffmail.com*
Director &Head ,
Department of Computer Applications,
S.N.R Sons College, Coimbatore

Bobby Lukose *bobbylukose74@gmail.com*
Assistant Professor
Department of Computer Science,
Hindusthan College of arts & Science College, Coimbatore

Abstract

Traditional techniques are based on restoring image values based on local smoothness constraints within fixed bandwidth windows where image structure is not considered. A common problem for such methods is how to choose the most appropriate bandwidth and the most suitable set of neighboring pixels to guide the reconstruction process. The present work proposes a denoising technique based on particle filtering using MRF (Markov Random Field).It isan automatic technique to capture the scale of the texture. The contribution of our method is the selection of an appropriate window in the image domain. For this we first construct a set containing all occurrences then the conditional pdf can be estimated with a histogram of all center pixel values.Our method explores multiple neighbors' sets that can be used for pixel denoising, through a particle filtering approach. This technique associates weights for each hypothesis according to its relevance and its contribution in the denoising process.

Keywords: Additive Guassian Noise, MC, MRF,Transition, Particle Filters

1. INTRODUCTION

A digital image could get corrupted easily due to various types of noise during transmission and acquisition. A noise is any unwanted signal/pixel that may be added or subtracted during transmission. These unwanted signals/pixels decrease the image quality. The sources of noise in digital images arise during image acquisition and/or transmission with unavoidable short noise of an ideal photon detector.

1.1 Additive and Multiplicative Noises

Noise is undesired information that contaminates the image. Typical images are corrupted with noise modeled with either a Gaussian, uniform, or salt and pepper distribution. Another typical noise is a speckle noise, which is multiplicative in nature.

An additive noise follows the rule$w(x, y) = s(x, y) + n(x, y)$, while the multiplicative noise satisfies$w(x, y) = s(x, y) \times n(x, y)$, where $s(x, y)$ is the original signal, *and* (x, y) denotes the noise introduced into the signal to produce the corrupted image $w(x, y)$, and (x, y) represents the pixel location.

Image noise is the random variation of brightness or color information in images produced by the sensor and circuitry of a scanner or digital camera. Image noise can also originate in film grain and in the unavoidable shot noise of an ideal photon detector. Image noise is generally regarded as an undesirable by-product of image capture. Although these unwanted fluctuations became known as "noise" by analogy with unwanted sound they are inaudible and such as dithering. The types of Noise are following :

- Amplifier noise (Gaussian noise)
- Salt-and-pepper noise
- Shot noise (Poisson noise)
- Speckle noise

Amplifier noise / Gaussian Noise

Noise is a statistical noise that has its PDF (probability density function) equal to that of the normal distribution, which is also known as the Gaussian distribution. Gaussian noise is most commonly known as additive white Gaussian noise. The term "white Gaussian noise" could be precise. Gaussian noise is evenly distributed over the signal. This means that each pixel in the noisy image is the sum of the true pixel value and a random Gaussian distributed noise value [16]. The standard model of amplifier noise is additive, Gaussian, independently at each pixel and independent of the signal intensity. Amplifier noise is a major part of the "read noise" of an image sensor, that is, of the constant noise level in dark areas of the image.

Salt-and-pepper noise

An image containing salt-and-pepper noise will have dark pixels in bright regions and bright pixels in dark regions. This type of noise can be caused by dead pixels, analog-to-digital converter errors, bit errors in transmission, etc. This can be eliminated in large part by using dark frame subtraction and by interpolating around dark/bright pixels.

Poisson noise

Poisson noise or shot noise is a type of electronic noise that occurs when the finite number of particles that carry energy, such as electrons in an electronic circuit or photons in an optical device, is small enough to give rise to detectable statistical fluctuations in a measurement.

Speckle noise

Speckle noise is a granular noise that inherently exists in and degrades the quality of the active radar and synthetic aperture radar (SAR) images. Speckle noise in conventional radar results from random fluctuations in the return signal from an object that is no bigger than a single image-processing element. It increases the mean gray level of a local area. Speckle noise in SAR is generally more serious, causing difficulties for image interpretation. It is caused by coherent processing of backscattered signals from multiple distributed targets. In SAR oceanography, for example, speckle noise is caused by signals from elementary scatters, the gravity-capillary ripples, and manifests as a pedestal image, beneath the image of the sea waves.

Image de-noising is a common procedure in digital image processing aiming at the suppression of different types of noises without losing much detail contained in an image. This procedure is traditionally performed in the spatial-domain or transform-domain by filtering. To reduce the noise from images, various images de-noising filters are used. Image denoising still remains a challenge for researchers because noise removal introduces artifacts, blurring of the images, and the noise remaining in the image edges.

A traditional way to remove noise from image data is spatial filters. Spatial filters are a low pass filter. It can be further classified into non-linear and linear filters.Linear filters, which consist of convolving the image with a constant matrix to obtain a linear combination of neighborhood values, have been widely used for noise elimination in the presence of additive noise. Linear filters destroy lines and other fine image details, also it produce a blurred and smoothed image with poor feature localization and incomplete noise suppression. Variety of nonlinear median type filters such as weighted median, rank conditioned rank selection has been developed to overcome this drawback.

The transform domain filtering methods can be subdivided as data adaptive and non-adaptive. Nonadaptive transforms are discussed first since they are more popular**.**The conventional Fast Fourier Transform (FFT) based image denoising method is a low pass filtering technique in which edge is not as sharp. The edge information is spread across frequencies because of the FFT basis functions, which are not being localized in time or space.

Wavelet Analysis, a new form of signal analysis is more efficient than Fourier analysis. Wavelet transforms are classified into discrete wavelet transforms (DWTs) and continuous wavelet transforms (CWTs). Both DWT and CWT are continuous-time (analog) transforms. But, the localized nature of the wavelet transforms both in time and space results in denoising with edge preservation**.**Wavelet Transform is the nonlinear coefficient thresholding based methods. It enables the separation of signal from noise. The procedure in which small coefficients are removed while others are left untouched is called Hard Thresholding. But the method generates spurious blips, better known as artifacts. To overcome the demerits of hard thresholding, wavelet transformusing soft thresholding. In wavelet based denoising methods, the noise is estimated and wavelet coefficients are threshold to separate signal and noise. Denoising of images using VisuShrink, SureShrink and Bayes Shrink, all these methods are based on the application ofwavelet transforms. Bayes shrinks wavelet denoising has been widely used for image denoising. The focus was shifted from the Spatial and Fourier domain to the Wavelet transform domain.

Edge-Preserving Smoothing Filters (Neighborhoodfilters) areBilateral filter, sigma filter,mean sigma filter. They are used tosolve the HALO (artifacts) and noise. The extensions of Adaptive range and domain filters is Bilateral Filterwhich performs weighted averaging in both range and domain [28].It smooth's noisy images while preserving edges using neighboring pixels. Bilateral filtering is a local, nonlinear, and a non iterative technique which considers both gray level and color similarities and geometric closeness of the neighboring pixels.

The NL mean filter achieves the best results in term of small detail preserving since noise contains less image information.The disadvantage is slow in terms of computation time. The NL-mean algorithm assumes fixed size with respect to the local filtering window (window centered at the origin pixel). Then based on the similarity between the center patch and the candidate patches it performs filtering [5] [19].

The sigma filter identifies impulse noise from noisy gray scale images by utilizing the standard deviation measure [20]. And the mean shift filter does not require prior knowledge of the number of clusters, and does not constrain the shape of the clustersthe main advantage of this method is computationalefficiency, but it is constrained by the amountof information present at the considered window.

In the particle filter, the posteriorprobability density is approximated as a set of particles [1]. Whenthe particles are properly placed, weighted and propagated,posteriors can be estimated sequentially over time. The densityof particles represents the probability of posterior function. In this the set of candidate pixels not fixed and changes per pixel location according to local pixel properties. The disadvantage iseven with a large number of particles, there are no particles in the vicinity of the correct state.This is called the particle deprivation problem.In this paper presented an efficient particle filtering algorithm using MRF which overcomes the above disadvantages and to remove low to high density noise for several standard images. This denoising algorithm should be able to extract the most important correlations of local structure of the entire image domain. Gaussian kernels are the most common selection of such an approach. Sequential Monte Carlo is a well known technique evolving densities to the different hypotheses.

The remainder of this document is organized as follows:In sectionII discuss about the MRF of image structure learning. Application of particle filter image denoising is presented in section 3, experimental resultsand comparisons are presented in section 4, and tables and figures are presented in section 5. Finally, we conclude in section6.

2. PRESENT WORK

Capturing the geometric structure of the image is the important process in image restoration. Such a process involvestwo steps, (i) a learning stage where the image structure ismodeled, and (ii) a reconstruction step.Our aim is to introduce a strategythat allows a best possible selection of the pixels contributingto the reconstruction process driven by the observed imagegeometry. Using this to retrievesimilar pixels. The issues are (i) the selection of the trajectory,(ii) and the evaluation of the trajectory appropriateness.Each walk is composed of a number of possible neighboringsites/pixels in the image which are determined according tothe observed image structure. To overcome the issues optimizing the selectionof candidate pixels within a walk as well as the overallperformance of the method image structure at local scale isconsidered as a learning stage. It computes aprobability density function that describes the spatial relationbetween similar image patches in a local scale.Here to improve the scale of texture by using MRF.

2.1 Automatic capturing of scale of texture using Markov Random field (MRF)

MRF models have been used in image restoration, region segmentation, and texture synthesis. In image processing, texture may be defined in terms of spatial interactions between pixel gray levels within a digital image. The aim of texture analysis is to capture the visual characteristics of a texture by mathematically modeling these spatial interactions. Markov Random Fields (MRFs) are widely used probabilistic models for regularization. The probability density function (pdf) defined by the MRF is the normalization constant. Maximum Likelihood (ML), probably the most common and popular method of probabilistic parameter estimation,require the pdf to be normalized. Sampling techniques, such as Markov chain Monte Carlo (MCMC) used in this model [11] [12]. Our aim is to preserve the local structure of the texture as much as possible. To achieve this to define a strategy to generate neighborhood candidate windows that takes into account the image content. After that to determine the most appropriate window for estimating the image intensity in a given position.

2.2 Steps for synthesizing one pixel

1. LetI be an image that is being synthesized from a texture sample image $I_{smp} \subset I_{real}$ whereI_{real} is the real infinitetexture.
2. Let$p \varepsilon I$ be a pixel and let $\omega (p) \subset I$be a square image patch of width ω centered at p.
3. Let$d (\omega_1, \omega_2)$denotes some perceptual distance between two patches.
4. Assume that all pixels in I except for pair known.
5. To synthesize the value of p we first construct an approximation to the conditional probability
6. distribution$P(p|\omega (p))$ and then sample from it.
7. Based on our MRF model we assume that p isindependent of $I \backslash \omega (p)$ given$\omega (p)$.
8. If we define a set $\Omega (p) = \{\acute{\omega} \subset Ireal : d (\acute{\omega}, \omega (p)) = 0\}$
9. containing all occurrences of $\omega (p)$ inI_{real}, then the conditional pdf of p can be estimated with a
10. histogram ofall center pixelvalues inΩ (p).
11. Then a variation of the nearest neighbor technique is usedfor finding the closest match.
12. If the closest match

$$\omega_{best}=\operatorname{argmin}_{\omega} d(\omega(p),\omega) \subset I_{smp} \quad (1)$$

is found, and all image patchesω with$D (\omega (p),\omega)<(1+ \varepsilon)d (\omega (p),\omega_{best})$is included in$\Omega'(p)$, where $\varepsilon = 0.1$.
13. The center pixel values of patchesin $\Omega'(p)$ give us a histogramfor p, which can then be sampled, either niformly or weighted by d.
14. Thento find a suitable distance d by using normalized sum of squared differences metricd_{SSD}.
15. This metric gives the same weight to any mismatched pixel, whether near the center or at theedge ofthe window.
16. To generate neighborhood candidate windows that take into account the image content.
17. Finally to determine the most appropriate window to estimate the image intensity in a givenposition.

3. PARTICLE FILTER IMAGE DENOISINGFRAMEWORK

The probability density function (pdf) that aims to find a spatial representation of different structures through the computation of the relative position of similar patches. Given this density, our aim is to determine the most appropriate set of neighbors to estimate the noise-free intensity of a given pixel. This is done through particle filtering technique. The particle filter is a special version of the Bayes filterbased on Monte Carlo sampling [7]. The particle filter algorithm consists of three steps:sampling, calculation of the importance weight and resampling. In the sampling step, samples are generated according to pdf. In the step of importance weighting, theimportance weight is computed for each particle. In the resampling step, the particles with different weights aresampled again with replacement according to their weights andthe particles with different weights are replaced by the newparticles with equal weights. The particles with largerweights are more likely to be selected than the particles withsmaller weights.

3.1 Bayesian Tracking

Filtering is to determine an estimation of the state vector. In the Bayesian framework to compute the pdf of a system, based on observations [18]. The recursive computation of the priori and the posteriori pdfleads to the exact computation of the posterior density. Particle filters, which are sequential Monte-Carlo techniques, estimate the Bayesian posterior probability density function (pdf) with a set of samples.The posterior pdf is computed using the equation

$$P (x_k | z_{1:k}) = \frac{p (z_k | x_k) p(x_k | z_{1:k-1})}{P(z_k|z_{1:k-1})} \quad (2)$$

Where$x_k$is the state vector and$(z_{1:k})$ is set of all the available observations. Similarly, the prior pdf is also computed usingthe equation

$$p(x_k | z_{1:k}) = \int p(x_k | x_{1:k-1})\, p(x_{k-1} | z_{1:k-1}) dx_{k-1} \quad (3)$$

3.2 Sampling Importance Resampling Filter

The sequential Importance Sampling (SIS) algorithm is a Monte Carlo (MC) method that forms the basis for most sequential MC filters developed over the past decades. The key idea is to represent the required posterior density function by a set of random samples with associated weights and to compute estimates based on these samples and weights.The posterior pdf can be approximated by a discrete weighted sum

$$P(x_k \mid z_{1:k}) \approx \sum_{m=1}^{Np} \omega^m_k \delta(x_k - x^m_k) \text{ and } \sum_{m=1}^{Np} \omega^m_k = 1 \quad (4)$$

whereδ is the kronector delta function and Np is the random state samples. The samples are generated through the principles of Importance Sampling. Then compute the particle weights iteratively according to

$$\omega^m_k \ \alpha \omega^m_{k-1}\, p(z_k|x^m_k) \quad (5)$$

A common problem is samples degeneration, where many particles carry less information and the variance in the weight increases. So that many particles have their weight close to zero. To overcome this problem a resampling step is necessary. The resampling scheme used to eliminate particle with smaller weights.

3.3 Transition Model

The transition according to the probability density function means that a maximum number of particles explore sites that are similar to the original pixel [26].

3.4 Likelihood measure

Measuring similarities between image patches is an important thing. In the case of denoising, filtering approach should consider pixels with the exact same value.Parallel to that, each particle corresponds to a random walk where a certain number of pixels have been selected and contribute a new element to the denoising process. The random walk is the set of the sites contributing to the reconstruction of a given pixel is determined [2]. This measure evaluates the new candidate pixel position using the formula

$$D_{sk} = \frac{\sum_{V\varepsilon[-w,w]^2} |I(x_0+v)-I(x_k+v)|^2}{(2w+1)^2} \quad (6)$$

whereD_{sk} is the similarity measure, x_0 is the original pixel and x_k is the current particle position and w is the bandwidth. The observation of the walk variance with respect to the origin value

$$D_{vk}=(1/k)\sum_{p=0}^{k}(I(x_p)-I(x_0))^2 \quad (7)$$

The patch around the original pixel is defined as an exponential function of the two metrics.

$$\omega_k = e^{-(D_{sk}/2\sigma^2_g + D_{vk}/2\sigma^2_v)} \quad (8)$$

whereσ_g and σ_v are constants that determine the bandwidth of the weight computation function.

3.5 Intensity Reconstruction

For each pixel of the image, generate particles by applying perturbations starting from the initial position according to the transition law. Repeating the process for each particle number of times. To reconstruct the original pixel according to the walk of the particle using the formula

k

$$\hat{U}^{m}_{k}(x) = \left(\frac{1}{k}\right)\sum_{P=0} I(x^{m}_{p}) \quad (9)$$

where m is the particle. And the reconstructed value is a weighted average of the mean intensity of each walk which is defined as

$$\hat{U}_{k}(x) = \sum_{m=0}^{N_p} \omega^{m}_{k}\hat{U}^{m}_{k}(x) \quad (10)$$

The whole process such as transition, weight computation and resampling is repeated a number of times.

3.6 Denoising Algorithm

1. Read the Gaussian noisy image
2. Split the image into different candidate windows
3. Calculate pdf for each windows
4. calculate the mean value for each window
5. choose minimum mean value window using the equation (1)
6. apply particle filter to this window
7. For each pixel repeat the steps 8 to 14
8. Generate particle according to the pdf
9. Compute intensity for each walk using equation (9)
10. Compute the weight of each particle using equation (8)
11. Normalize the weight of the particle
12. Compute the estimated intensity
13. Perform resampling
14. Select the best pixel position
15. Final pixel intensity estimation = weighted mean of all filtered values of different random walks using equation (10).

4. EXPERIMENTAL RESULTS

The proposed algorithm is tested using 256 X 256 8-bits/pixel standardgray scaleimages. There are 20 imagestaken from the Berkely Segmentation Dataset & Benchmark database. The performance of the proposed algorithm is tested with different noise levels.Each time the test image is corrupted by different additive white Gaussian noise standard deviation ranging from 10 to 50 with an increment of 10. These noisy images are denoised by two algorithms and the performancedifference between the particle filter and the proposedapproach measured by the parameters PSNR and MSE. All the filters are implemented inMatlab 10.

A quantitative measure of comparison likePeak signal to noise ratio (PSNR), mean square error (MSE) is used in this work.

$$PSNR=10\log_{10}\frac{255^{2}}{MSE} \quad (12)$$

$$MSE=\frac{1}{\|\Omega\|}\sum_{x \in \Omega}(U(x)-\hat{U}(x))^{2} \quad (13)$$

4.1 Algorithm for Peak Signal to Noise ratio (PSNR)

Step1: Difference of noisy images and noiseless image iscalculated.

Step2: Size of the matrix obtains in step 1 is calculated.
Step3: Each of the pixels in the matrix obtained in step2 issquared.
Step4: Sum of all the pixels in the matrix obtained instep3 is calculated.
Step5: (MSE) is obtained by taking the ratio of valueobtained in step 4 to the value obtained in the Step2.
Step6: (RMSE) is calculated by taking the square rootof thevalue obtained in Step5.
Step7: Dividing 255 with RMSE, taking log base 10 andmultiplying by 20 gives the value of PSNR.

4.2 Algorithm for Root Mean Square Error (RMSE)

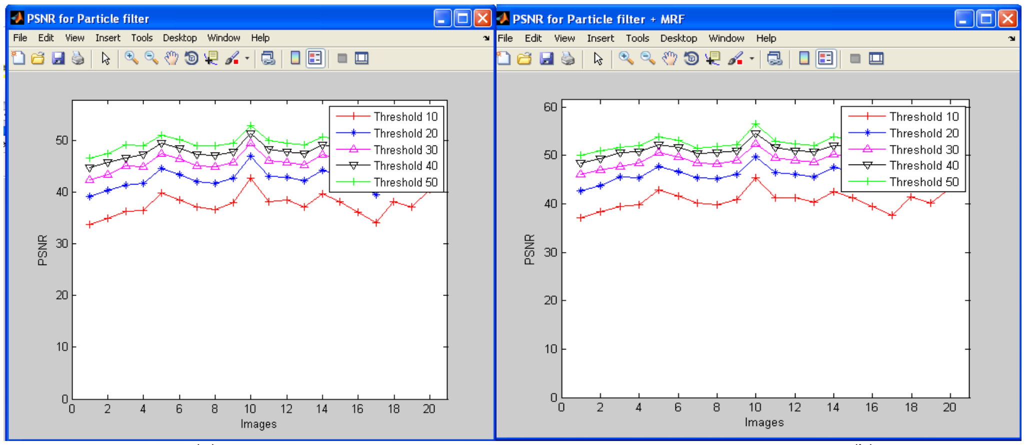
Step1: Difference of noisy images and noiseless image is calculated.
Step2: Size of the matrix obtains in Step1 is calculated.
Step3: Each of the pixels in the matrix obtained in step 2 is squared.
Step4: Sum of all the pixels in the matrix obtained in step 3 is *calculated.*
Step5: (MSE) is obtained by taking the ratio of value obtained in Step4 to the value obtained in the step 2.
Step6: (RMSE) is calculated by taking the square rootof the *value* obtained in Step5.

In Table 1, provide a PSNR value of restored images for the particle filter and the MRF particle filter. As seen the results of Table I the MRF particle Filter method produces very good results. The PSNR values for particle filter and the MRF particle filter for 20 images at different Gaussian levels are displayed in Fig. 1.(a) and (b) respectively. We can make several observations from these plots.In Fig. 2. (a) and (b) the PSNR values for particle filter and MRF particle filter for 20 images at Gaussian noise σ=20 and MSE values for particle filter and MRF particle filter for 20 images at Gaussian noise σ=20 are displayed respectively.The visual quality results are presented in Fig. 3. Noise free image, Gaussian noise image, restored image using particle filter, restored image using MRF particle filters for 20 images at Gaussian noise σ=20 as shown in Fig. 3.(a), (b), (c) and (d) respectively.In all graphs the x-axis values are represented as 1,2,3 etc. which denotes 1 for baboon, 2 for Barbara respectively that are presented in Table 1.The visual quality and quantitative results clearly show the MRF particle filter perform much better thana particle filter in terms of PSNR and MSE.

4.3Tables And Figures

Images	Particle filter					Particle filter using MRF				
	σ =10	σ =20	σ =30	σ =40	σ =50	σ =10	σ =20	σ =30	σ =40	σ =50
Baboon	33.72	39.16	42.44	44.79	46.58	37.17	42.76	46.06	48.42	49.98
Barbara	34.99	40.34	43.43	45.76	47.47	38.38	43.82	47.05	49.25	51.00
Boathouse	36.23	41.39	45.10	46.65	49.05	39.52	45.55	47.70	50.50	51.67
Bridge	36.49	41.75	44.97	47.19	48.97	39.82	45.31	48.39	50.76	52.09
Building	39.90	44.56	47.49	49.49	50.99	42.86	47.79	50.54	52.29	53.80
Cameraman	38.46	43.36	46.38	48.53	50.11	41.57	46.72	49.64	51.69	53.18
Capsicum	37.08	41.99	45.12	47.20	49.00	40.17	45.45	48.40	50.43	51.54
Flyover	36.64	41.71	44.87	47.11	48.88	39.84	45.19	48.33	50.59	51.85
Girl	37.99	42.71	45.74	47.80	49.47	40.95	46.07	48.92	50.88	52.14
Helicopter	42.77	46.87	49.48	51.29	52.84	45.35	49.73	52.36	54.50	56.46
Hills	38.17	43.05	46.08	48.25	49.95	41.24	46.38	49.45	51.61	52.92
Lena	38.49	42.90	45.74	47.84	49.40	41.24	46.02	48.95	50.91	52.33
Mechanic	37.14	42.16	45.30	47.43	49.20	40.32	45.63	48.61	50.86	52.06
Monkey	39.71	44.28	47.21	49.14	50.63	42.58	47.50	50.17	52.08	53.82
Nature	38.17	43.15	46.30	48.56	50.36	41.28	46.62	49.78	52.19	53.37
OldWomen	36.13	41.38	44.59	46.85	48.52	39.47	44.90	48.00	50.22	51.52
Owl	34.08	39.57	42.88	45.24	47.01	37.57	43.22	46.44	48.80	50.07
Pelican	38.11	43.27	46.44	48.72	50.46	41.37	46.75	49.93	52.24	53.52
Starfish	37.12	41.96	45.00	47.20	48.85	40.16	45.30	48.36	50.42	51.67
Water plant	40.51	44.75	47.47	49.46	50.89	43.18	47.74	50.46	52.34	53.85

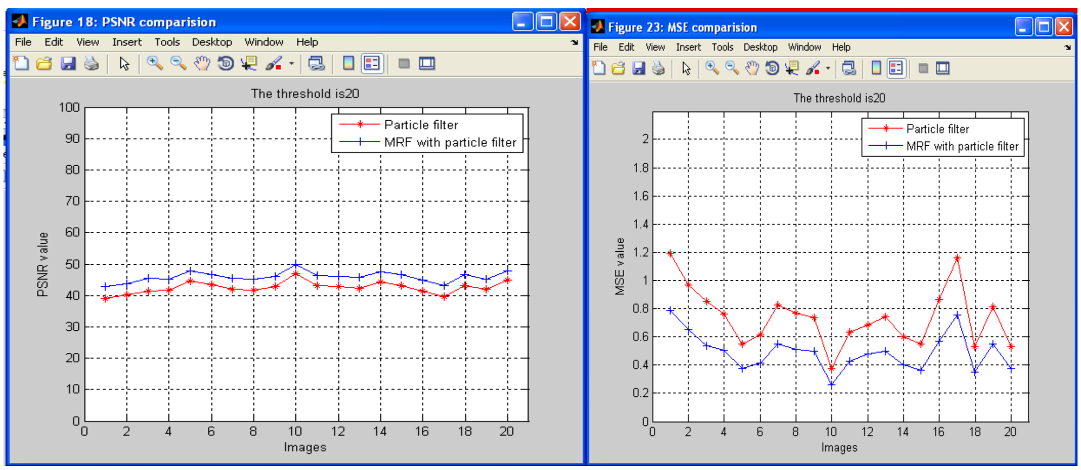
TABLE 1:PSNR Values For The Denoised Images At Different Guassian Noise Levels



(a) (b)

FIGURE 1:(a) PSNR of particle filter for 20 images at different Gaussian noise levels

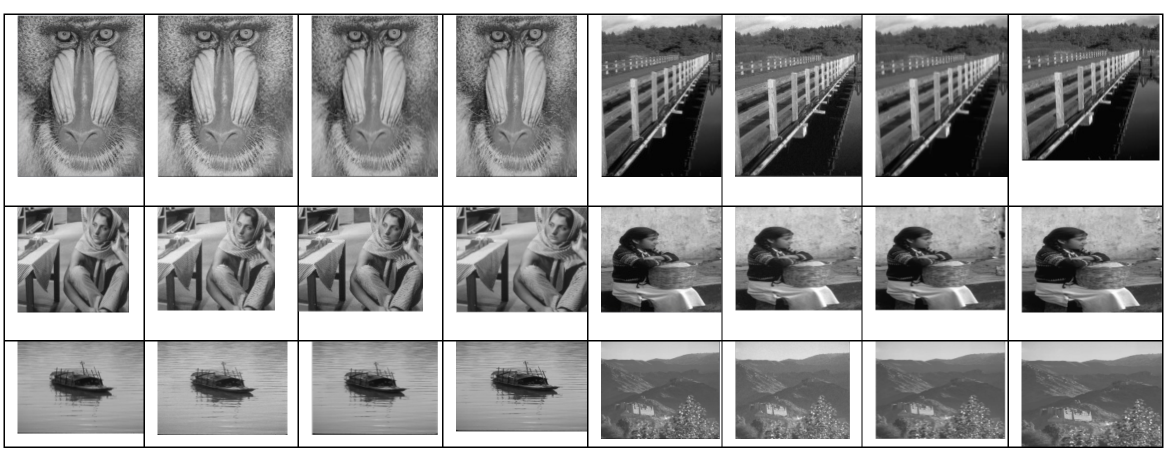
(b) PSNRof MRF particle filter for 20 images at different Gaussian noise levels



(a) (b)

FIGURE 2. (a) MSE of particle filter & MRF particle filter for 20 images at Gaussian noise level σ=20

(b) PSNR of particle filter & MRF particle filter for 20 images at Gaussian noise level σ=20



(a) (b) © (d) (a) (b) © (d)

FIGURE 3: (a) Noise Free Image **(b)**Image Corrupted with Gaussian noise σ = 20

(c) Restored Image using Particle Filter**(d)**Restored Image usingMRF particle filter

5. CONCLUSION

In this paper, an efficient algorithm is proposed for removing noise from corrupted image. This was achieved by particle filtering using MRF which isan automatic technique to capture the scale of the texture. The contribution of our method is the selection of an appropriate window in the image domain. For this we first construct a set containing all occurrences then the conditional pdf can be estimated with a histogram of all center pixel values. Particle evolution is controlled by the image structure leading to a filtering window adapted to the image content. To demonstrate the superior performance of the proposed method, extensive experiments have been conducted on several standard test images. The proposed MRF particle filter performs better than particle filter both in PSNR and visually. Promising experimentalresults demonstrate the potentials of our approach.

The limitation of this work is Computational complexity. And the performance of the system is reducedsince each particle is iterated separately.

Future work is First,improving the learning stage of the image structural modeland guiding the particles to the most appropriate directionscould be a step toward increasing the efficiency of particletransitions. Next, thelikelihood measure could be also modified to be more specificto each noise distribution and more robust. Last, the ability to perform the process in parallel for all pixelsand benefit from the reconstructed values might improve theperformance of the method.

6. REFERENCES

[1]M. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for on-line non-linear/non-Gaussian Bayesiantracking," *IEEE Trans. Signal Process.*, vol. 50, pp. 174–188, 2002.

[2] N. Azzabou, N. Paragios, and F. Guichard, "Random walks, constrained multiple hypothesis testing and image enhancement," in*Proc.Eur. Conf. ComputerVision*, 2006, pp. 379–390.

[3] N. Azzabou, N. Paragios, F. Guichard, and F. Cao, "Variable bandwidth image denoising using image-based noise models," in *Proc. IEEE Int.Conf. Computer Vision and Pattern Recognition*, 2007, pp. 1–7.

[4] J. S. D. Bonet. Multiresolution sampling procedure for analysis and synthesis of texture images. In *SIGGRAPH '97*, pages 361–368,1997.

[5] A. Buades, B. Coll, and J.-M.Morel, "A non-local algorithm for image denoising," in *Proc. IEEE Int. Conf. Computer Vision and Pattern Recognition*, 2005, pp. 60–65.

[6] J. Carpenter, P. Clifford, and P. Fearnhead, "Improved particle filter for nonlinear problems," *Proc. Inst. Elect. Eng., Radar, Sonar, Navig.*, 1999.

[7] T. Clapp and S. Godsill, "Improvement strategies for Monte Carlo particle filters," in *Sequential Monte Carlo Methods in Practice*, A. Doucet, J. F. G.deFreitas, and N. J. Gordon, Eds. New York: Springer-Verlag, 2001.

[8]P. Del Moral, "Non-linear filtering: Interacting particle solution," *Markov Processes Related Fields*, vol. 2, no. 4, pp. 555–580.

[9] A. Doucet, J. de Freitas, and N. Gordon*, Sequential Monte Carlo Methods in Practice.*New York: Springer-Verlag, 2001.

[10]A. Doucet, N. Gordon, and V. Krishnamurthy, "Particle filters for state estimation of jump Markov linear systems," *IEEE Trans.Signal Processing*, vol. 49, pp. 613–624, Mar. 2001.

[11]A. Efros and T. Leung, "Texture synthesis by non-parametric sampling," in *Proc. Int. Conf. Computer Vision*, 1999, pp. 1033–1038.

[12]D. Geman, "Random fields and inverse problems in imaging," in Lecture Notes in Mathematics, vol. 1427, pp. 113–193. Springer–Verlag, 1991.

[13]S. Geman and C. Graffigne, "Markov random field image models and their applications to computer vision," Proceedings of theInternational Congress of Mathematicians, pp. 1496–1517, 1986.

[14] R. M. Haralick, K. Shanmugam, and I. Dinstein, "Textural features for image classification," *IEEE Trans. Syst., Man, Cybern.*, vol.SMC-6, pp. 610–621, 1973.

[15] R. E. Helmick, D. Blair, and S. A. Hoffman, "Fixed-interval smoothing for Markovian switching systems," *IEEE Trans. Inform.Theory*, vol. 41, pp.1845–1855, Nov. 1995.

[16]J. Huang and D. Mumford, "Statistics of natural images and models," in *Proc. IEEE Int. Conf. Computer Vision and PatternRecognition*, 1999, pp. 541–547.

[17]John Moussouris, "Gibbs and Markov random systems with constraints," Journal of Statistical

Physics, vol. 10, no. 1, pp. 11–33, 1974.

[18]G. Kitagawa, “Monte carlo filter and smoother for non-Gaussian nonlinear state space models,” *J. Comput.Graph.Statist.*, vol. 5, pp.1–25, 1996.

[19]A. Lee, K. Pedersen, and D. Mumford, “The nonlinear statistics of high-contrast patches in natural images,” *Int. J. Comput.Vis.*, pp.83–103, 2003.

[20]S. Lee, “Digital image smoothing and the sigma filter,” *CVGIP*, vol. 24, no. 2, pp. 255–269, Nov. 1983.

[21] J. S. Liu and R. Chen, “Sequential Monte Carlo methods for dynamical systems,” *J. Amer. Statist.Assoc.*, vol. 93, pp. 1032–1044,1998.

[22]M. Mahmoudi and G. Sapiro, “Fast image and video denoising via nonlocal means of similar neighborhoods,” *IEEE SignalProcess.Lett.*, vol. 12, pp. 839–842, 2005.

[23]S.Mallat, “A theory for multiresolution signal decomposition: The wavelet representation,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 11, pp. 674–693,1989.

[24]J. Polzehl and V. Spokoiny, “Adaptive weights smoothing with applications to image restoration,” *J. Roy.Statist.Soc.B*, vol. 62, pp.335–354, 2000.

[25] Richard C. Dubes and Anil K. Jain, “Random field models in image analysis,” Journal of Applied Statis-tics, vol. 16, no. 2, pp. 131–164, 1989.

[26] B. Smolka and K. Wojciechowski, “Random walk approach to image enhancement,” *Signal Process.*, vol. 81, pp. 465–482, 2001.

[27] R. L. Streit and R. F. Barrett, “Frequency line tracking using hidden Markov models,” *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 38, pp. 586–598, Apr. 1990. 188 IEEE TRANSACTIONS ON SIGNAL PROCESSING, VOL. 50, NO. 2,FEBRUARY 2002.

[28]C. Tomasi and R. Manduchi, “Bilateral filtering for gray and color images,” in *Proc. Int. Conf. Computer Vision*, 1998, pp. 839–846.