

Rearrangement of Pixels and Patches to Remove Noise in Color Images and to Restore the Missing Pixels in it by Permuting its Pixels and Patches

M. Nithyabhavani¹ (Student), S. R. Malathi² (Associate Professor)
Department of Computer and Science and Engineering,
Sri Venkateswara College of Engineering,
Chennai. Tamilnadu, India

Abstract— An patch in an image is a combination of its picture elements. We reorder the patches in a color image which is the concatenation of the RGB components in a given corrupted image using an image processing scheme based on reordering of patches. Divide the given corrupted image into two sub-images such that they share some of their pixels with their neighboring patches and extract all patches with overlaps from each sub-image. We use the Euclidean measure to find the distance between these patches and use these distances to order the patches such that they are chained in “shortest possible path”, essentially solving traveling salesman problem. The obtained ordering applied to the corrupted image implies permutation of the image pixels to *what should be* a regular signal. To obtain the good recovery of the clean image, we apply the relatively simple one-dimensional operations such as filtering or interpolation. This method shows promising results in both image denoising and inpainting. We propose to restore the color of the given input image by considering the luminance and chrominance and transferring their values. Also, to automate the decision for denoising or inpainting by considering the image pixels. Also, we propose to apply an improved FMH (Finite Impulse Response Median Hybrid) filtering technique to the reordered set of patches. An improved FMH filtering technique is a combination of linear as well as nonlinear filtering technique that removes Gaussian noise and also preserves edges.

Keywords— Patches, patch based processing, denoising, inpainting, pixel permutation, travelling salesman problem.

I. INTRODUCTION

One of the fundamental challenges in the field of image processing and computer vision is image denoising, where the underlying goal is to estimate the original image by suppressing noise from a noise-contaminated version of an image. An image may contain various noises which are the small disturbances that make an image unclear. Noises may be produced by the sensor and circuitry of a scanner or digital camera. An Image noise is an undesirable byproduct of image capture that adds spurious and extraneous information. These noises in the image can be removed or suppressed by various methods of denoising. Image noise may be caused by different intrinsic (i.e., sensor) and extrinsic (i.e., environment) conditions which are often not possible to avoid

in practical situations. Therefore, image denoising plays an important role in a wide range of applications such as image restoration, visual tracking, image registration, image segmentation, and image classification. There are several ways that noise can be introduced into an image, depending on how the image is created.

Image inpainting is used to fill the missing data in an image given. The purpose of inpainting is to reconstruct missing regions in a visually plausible manner so that it seems reasonable to the human eye. Structural inpainting uses geometric approaches for filling in the missing information in the region, which should be inpainted. Structural inpainting algorithms focus on the consistency of the geometric structure. Texture has a repetitive pattern which means that a missing portion cannot be restored by continuing the level lines into the gap. Since the structural inpainting cannot restore the texture, the textural inpainting is used to restore the texture in an image. The patches are the combination of $n \times n$ pixels which forms the regions of the image. In recent years image processing using local patches are shown to be highly effective. The patches are generally small in size when compared to original image size (a typical patch size would be 8×8 pixels). The patches in an image are used in the application of image denoising, image segmentation, classification, decomposition, reconstruction etc. The overlapping patches are those which share some of its pixels with its neighboring patches. These patches with overlaps are extracted and process is carried on these extracted patches by exploiting the interrelations between these patches. The patches are put back in to image canvas after manipulation. The core idea of reordering of patches is adopted which seek a new way of organizing the patches. The pixels in a patch are reordered by obtaining the permutation of the image pixels in a patch. The permutation of the image pixels is drawn from the shortest possible path ordering of the image patches. The subimage for a given corrupted image (noisy, containing missing pixels etc) is divided and the patches with maximal overlaps are extracted which are then ordered according to shortest possible path essentially solving travelling salesman problem. It is assumed that the similarity between the two image patches implies the similarity between

their center pixels. The ordering of patches according to the shortest possible path is expected to be robust to distortions when the image is deteriorated (noisy, missing pixels etc).

In the existing work, the maximum overlapping patches are extracted after dividing the whole into subimages. The subimages are divided such a way that the some part of the image is shared by its neighboring subimage. The extracted patches with maximum overlaps are ordering such that they chained in the shortest possible path essentially solving travelling salesman problem. The patches which are extracted are further divided into smooth and non smooth patches. The patches without edges or textures are considered as smooth whereas the patches with edges or textures are considered as non-smooth patches. The one-dimensional operations such as filtering or interpolation are applied to the reordered set of overlapping patches in order to obtain the good recovery of clean image. This method showed an promising results in both image denoising and image inpainting separately. The following contributions are done through this research work:

- An improved FMH (Finite impulse response Median Hybrid) filter is applied to the reordered set of patches. The FMH is a combination of linear as well non linear filtering technique that removes Gaussian noise as well as preserves edges.
- The decision for denoising and inpainting is automated based on the pixel values of the smooth and non smooth patches.

II. REMOVAL NOISE AND RESTORATION OF MISSING PIXELS USING VARIOUS METHODS

Image processing using local patches is widely used in the applications such as image denoising. In the work [1], a generalized tree based wavelet transform to reconstruct the clean original image from the noisy image is proposed. This new wavelet transform is applicable to functions defined on high dimensional data, weighted graphs and networks. The proposed method generalized the Haar-like transform, and also constructed the data adaptive orthonormal wavelets beyond Haar. It is defined via a hierarchical tree, which is assumed to capture the geometry and structure of the input data, and is applied to the data using a modified version of the common one-dimensional (1D) wavelet filtering and decimation scheme. The corresponding tree obtained from the ordering of the features will be different, since the feature obtained from the image is also different. The distance between the noisy patches is used to find the similarity between clear versions of middle pixels. Each level of the random tree is constructed by choosing first point at random and then continued from the first point to its neighbor and second nearest neighbor and so on. The denoising is performed by applying the proposed wavelet transform using hard thresholding on transform coefficients and computing inverse transform. Also different wavelet filters are applied to the noisy version of the image.

In the work [2], the framework for color image restoration is proposed. The assumption that natural signals, such as images, admit a sparse decomposition over a redundant dictionary leads to efficient algorithms for handling such sources of data. In particular, the design of

well adapted dictionaries for images has been a major challenge. The K-SVD (K-means Singular Value Decomposition) algorithm is used to remove the noise from the color images. This can be easily performed by concatenating the RGB values to a single vector. Also this K-SVD algorithm is used to recover the missing pixels of an image. The missing values are treated as corrupted by strong impulse noise. The problem of non-homogeneous noise is very important as non-uniform noise across color channels is very common in digital cameras. The case of removing white Gaussian noise is addressed, but with a different standard deviation for each pixel/color channel, which makes it non-homogeneous. The problem of learning dictionaries for color images is also addressed and the K-SVD-based grayscale image denoising algorithm is extended.

In this work [3], a general mathematical and experimental methodology is defined to compare and classify classical image denoising algorithms, and proposed an algorithm (Non Local Means) to address the preservation of structure in a digital image. The mathematical analysis is based on the analysis of the “method noise”, defined as the difference between a digital image and its denoised version. The NL-means algorithm is proven to be asymptotically optimal under a generic statistical image model. The denoising performance of all considered methods is compared in three ways; mathematical-asymptotic order of magnitude of the method noise under regularity assumptions; perceptual-mathematical: the algorithms artifacts and their explanation as a violation of the image model; quantitative experimental: by tables of L_2 distances of the denoised version to the original image. The most powerful evaluation method seems, however, to be the visualization of the method noise on natural images.

A redundant tree-based wavelet transform (RTBWT) is used in the work [4], which extended the redundant wavelet transform to scalar functions defined on high-dimensional data clouds, graphs and networks. This transform is obtained by modifying an implementation of the redundant wavelet transform. This implementation employs a filter-bank decomposition scheme, which is similar to the ortho-normal discrete wavelet transform. However, in each level of this scheme none of the coefficients are discarded. Level operators that reorder the approximation coefficients is added in each decomposition. These operators are data-dependent, and are obtained using tree-like structures constructed from the data points. Each reordering operator is derived by organizing the tree-node features in the corresponding level of the tree so as to shorten the path that passes through these points. The reordering operators increase the regularity of the permuted approximation coefficients signals, which cause their representation with the proposed wavelet transform to be more efficient (sparse). It is noted that the proposed transform shares some of the similarities with the easy path wavelet transform, which also employs permutations in order to enhance its sparsity. The use of the proposed transform for the recovery of labels defined on point clouds is explored, and it is shown that it outperforms the 1-nearest neighbor (1-nn) classifier.

The work [5], proposed an image denoising algorithm, which applied permutations to the noisy image, but

overcomes the two shortcomings of applying linear smoothing filters such as 1) The linear smoothing filters did not take advantage of the distances between the noisy image patches, which were used in the reordering process; and 2) the smoothing filters required a separate training set to be learned from. The need for learning filters is by employing the nonlocal means (NL-means) algorithm and each pixel is estimated as a weighted average of noisy pixels in union of neighborhoods obtained from different global pixel permutations, where the weights are determined by distances between the patches. Also it is shown that the proposed scheme achieves results which are close to the state-of-the-art. This neighborhood is replaced with a union of neighborhoods obtained from different global pixel permutations, and the weights are determined using the distances between the patches. The performance of the proposed image denoising scheme is demonstrated, and combined with a patch classification and a subimage averaging scheme, it outperforms both the NL-means algorithm, and achieves results which are close to those of the BM3D algorithm.

III. PROPOSED SYSTEM

Image noise results in pixels that look different from its neighbors. Hence the permutation is carried on the pixels of each patch and the one dimensional operations are done after reordering the patches. This helps us to obtain the good recovery of the clean images. The Figure 2 shows the block diagram of denoising and inpainting of images using smooth ordering of patches. The corrupted image is divided into subimages and patches. The patches are reordered after the construction of permutation matrix for smooth and non smooth patches. It also shows that the decision for denoising and inpainting is automated. The filters are applied to it and the subimages are reconstructed. Each reconstructed subimage is plugged into its original place.

A. Subimage division and patch extraction:

The given image Z is divided into two subimages Z_1 and Z_2 . The two subimages are divided such that each subimage shares some of its pixels with its neighboring subimage especially the center of the pixels are shared by each subimage. For the subimages Z_1 and Z_2 the patches with maximal overlaps are extracted. The overlapping patches are those for given image n , the patches with size $\sqrt{n} \times \sqrt{n}$ are extracted such that each patch shares some of its pixels with nearest neighboring patches in both row wise and column wise. Once these patches are extracted their spatial relationships are disregarded.

B. Classification of smooth and non smooth patches:

The divided subimages with the extracted patches are then converted into binary format so as to divide the smooth and non smooth patches effectively. For a given binary subimage Z_{1b} and Z_{2b} , the extracted patches are divided into smooth and non smooth. The smooth patches S_s are those which does not contain any edges of textures whereas the patches which contains edges of textures are called as the non smooth patches S_n . Moreover, the smooth patches are white

patches and non smooth patches are black patches. The patches are classified into smooth and non smooth such that if $\text{std}(x_i) < C\sigma$ then $x_i \in S_s$, otherwise $x_i \in S_n$. The extracted patches of subimage Z_{1b} are divided into smooth patches Z_{1bs} and non smooth patches Z_{1bn} . Similarly the subimage Z_{2b} is divided into smooth patches Z_{2bs} and non smooth patches Z_{2bn} . The pixel values of the smooth patches are assumed to have more number of zeros whereas the pixel values of the non smooth patches are assumed to have more number of ones. The permutation matrix is constructed for the smooth patches as well as for the non smooth patches. The process of permutation of image patches is carried out as shown in the figure 1.

$$Z_1 = \begin{Bmatrix} Z_{1s} \\ Z_{1n} \end{Bmatrix}, P = \begin{Bmatrix} P_s \\ P_n \end{Bmatrix}$$

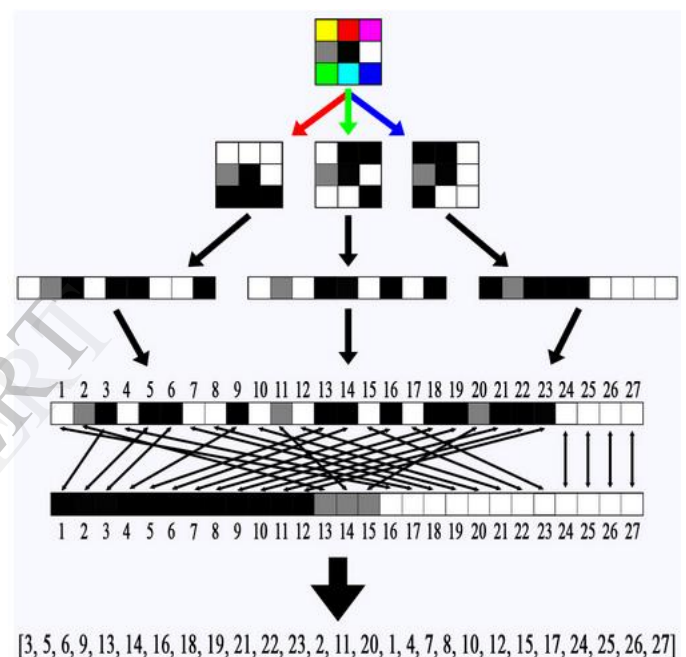


Figure 1: Permutation of Image Pixels

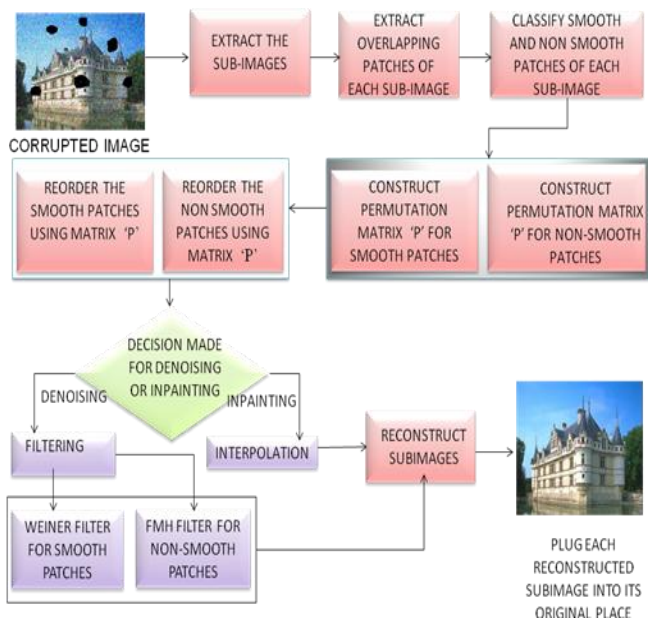


Figure 2: Block diagram of denoising and inpainting of images

C. Patch Reordering:

The key idea of patch reordering is the regularity due to ordering. The center (or any other) pixel in each patch is considered so that the new path is expected to lead to very smooth (or at least, piece-wise smooth) 1D signal. The ordering is expected to be robust to noise. The patches are reordered according to the shortest possible path considering the center pixel. The distance between the patches X_i and X_j is found out and the using the distance found the patches are organized such that the patches are arranged in the decreasing order of the distance between them. The distance between the patches are found using the squared Euclidean measure

$$\omega(X_i, X_j) = 1/n \|X_i - X_j\|^2$$

D. Application of filters:

The patches after reordering the filters are to applied to the set of reordered patches. Since the patches are reordered into smooth and non smooth separate filters has to be applied into smooth and non smooth. An improved FMH(Finite impulse response Median Hybrid Filter) is applied to smooth patches inorder obtain the deblurred image. The weiner filter is applied to the non smooth patches to obtain good recovery of the image.

E. Automation of decision making for denoising or inpainting:

The pixel values of the noisy image and the image to be inpainted is considered inorder to differentiate the image to be inpainted and denoised. It is noted that the image which contains noise which contain the pixel values which are completely different from those of the neighbouring pixels where as the image to be inpainted contains the pixels values different from those of noisy images but the pixel values of the pixels which are missing contains 0 in their respective position. Based on this criteria the decision of the denoising or inpainting is automated.

F. Restoration of color of the given image

Traditional colored images are a composition of three-dimensional (RGB) pixel values. For the task of “colorizing” a grayscale image, the three-dimensional (RGB) pixel values are assigned to an image that varied only along one dimension. To achieve this task, the color space YCbCr, are utilized where Y represents the luminance components and Cb and Cr represent the chromaticity components. The beauty of this color space is that it allows us to “color” the image by keeping one value constant while transferring the other dimensions. This idea is the basis of the whole “colorization” technique. Once the three-dimensional (RGB) pixel values were transformed into two-dimensional (YCbCr) values, we proceeded to do a comparison between the source color image and the destination grayscale image. With both images in the YCbCr format, we could choose to keep either the Y (luminance) or the CbCr (chromaticity) constant and transfer the information of the source color image onto the destination grayscale image. Regardless, with the YCbCr coefficients of the colored grayscale image generated by the gray2rgb.m file, we needed to carry out the following functions for conversion.

$$\begin{aligned} R' &= 1.164(Y - 16) + 1.596(Cr - 128) \\ G' &= 1.164(Y - 16) - 0.813(Cr - 128) - 0.392(Cb - 128) \\ B' &= 1.164(Y - 16) + 2.017(Cb - 128) \end{aligned}$$

IV. APPLICATIONS AND IMPLEMENTATION RESULTS

In our proposed system the FMH filter is used, which removes impulse noise as well as it reduces jitter and preserves edges. Also the decision making for the denoising or inpainting is also automated.

A. Input image:

Image which contains which noise is taken as the input image. Various noises are contaminated in the image which makes the images completely unclear. The input noisy image is then divided into two subimages.



244	220	188	187	188	230	241	189	242	246	229	189	229	229	216	197	182	179
226	246	246	230	230	244	249	244	240	244	249	246	230	230	222	208	222	243
202	220	248	242	221	189	245	247	230	219	231	241	244	242	219	225	246	239
189	250	250	242	210	242	250	232	231	226	245	226	244	219	231	218	247	233
211	221	248	250	245	226	209	220	220	231	226	236	244	221	247	250	247	234
193	232	252	248	246	227	246	216	243	248	246	245	243	220	232	235	250	248
190	233	233	233	220	249	233	244	232	248	252	228	232	221	222	248	227	232
228	248	224	220	244	246	249	233	244	248	238	246	243	248	228	221	245	228
196	228	249	248	234	251	244	228	193	220	220	249	249	238	224	222	248	233
193	212	248	236	223	223	228	194	232	193	228	246	248	235	234	230	192	192
251	253	248	223	223	236	193	194	233	247	234	234	223	246	252	230	236	233
248	255	237	237	189	247	236	226	236	236	247	223	249	247	250	213	233	212
220	240	236	236	231	255	251	231	237	236	248	229	231	253	255	255	248	255
214	246	255	237	255	248	248	255	237	237	220	197	231	236	255	251	255	231
251	248	255	253	195	253	254	253	253	237	236	251	226	240	248	239	240	253
227	254	253	214	228	251	237	237	237	226	237	253	237	237	248	254	253	227
195	239	251	251	233	239	196	251	238	238	239	255	233	254	253	246	237	248
201	252	250	238	241	250	253	255	255	250	255	255	253	238	252	255	255	254
239	229	229	255	239	255	244	242	252	253	255	255	255	228	253	255	255	239
197	255	229	255	250	242	251	250	239	250	250	228	239	255	253	255	255	253
197	242	255	240	253	250	255	250	255	240	235	211	255	243	253	252	255	253
243	241	246	255	255	255	255	243	245	239	200	255	251	255	243	255	255	251
254	254	255	237	200	237	241	251	242	231	254	255	255	255	255	251	255	251
254	255	255	237	255	233	230	255	237	255	255	241	254	246	255	255	241	255

Figure 3: Noisy Input image and its pixel values

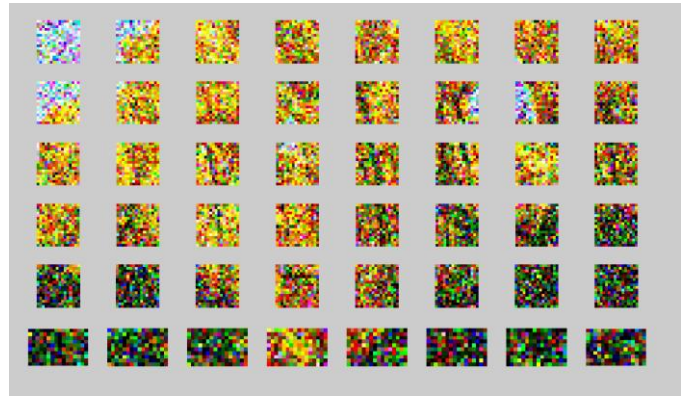


Figure 6: Patches of sub image 2

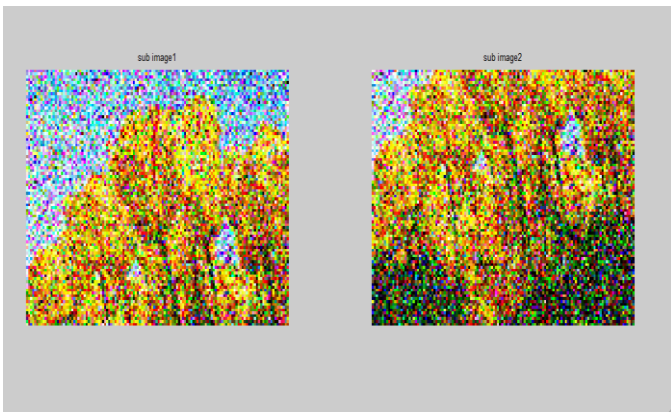


Figure 4: Subimages

B. Overlapping patches:

The patches are extracted such that the patches of the images in the rows are overlapping with the patches of the images in the columns and they are ordered such that they are chained in the shortest possible path.

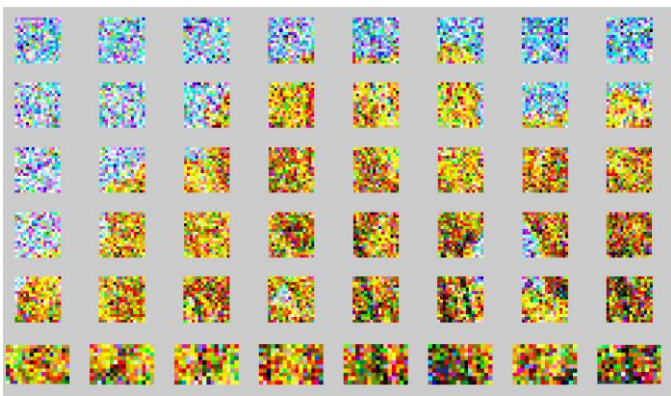


Figure 5: Patches of the subimage 1

C. Classification of smooth and non smooth patches:

The patches are classified into smooth and non smooth patches by converting the image into binary. The smooth patches are considered as the patches with more number of 0s where as the non smooth patches contains edges and textures.

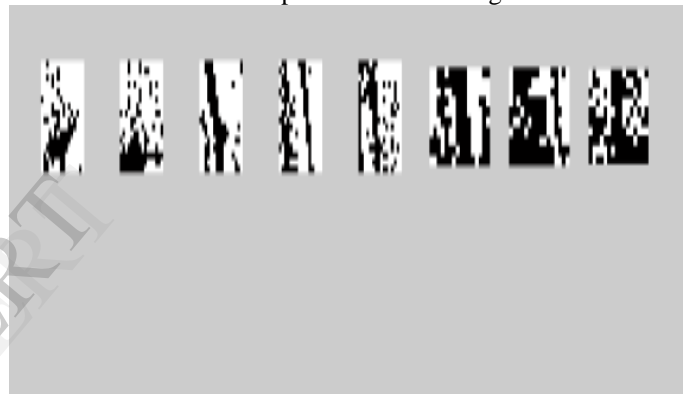


Figure 7: Non smooth patches of subimage 1

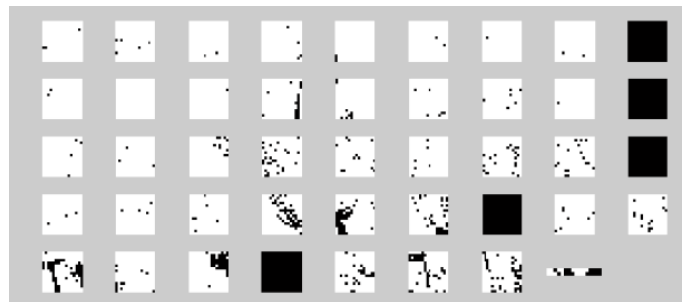


Figure 8: Smooth patches of subimage 1

F. Image inpainting:

The input image will be the image with maximum number of pixels missing. It implies that the values of the missing pixels will be 0.

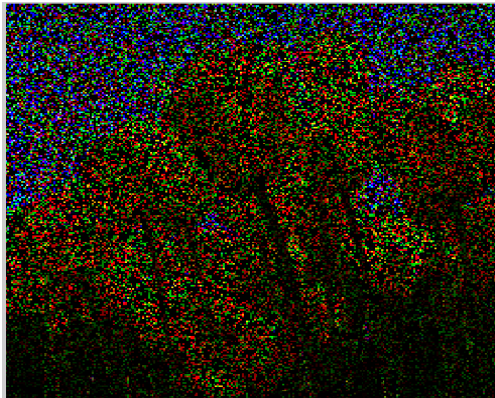


Figure 14: Input image with missing pixel

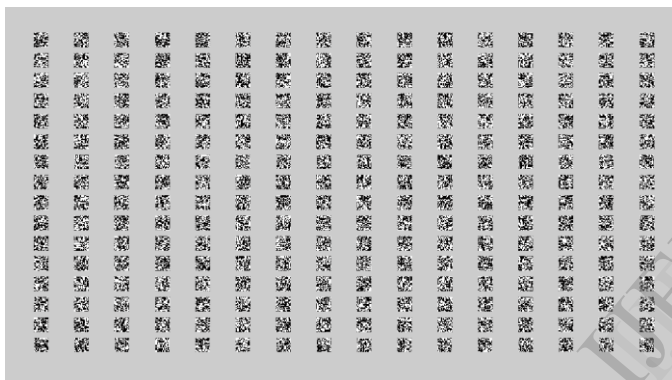


Figure 15: Patches of the input image

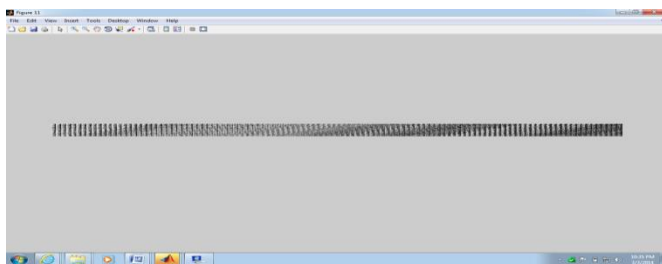


Figure 16: Reordering of patches

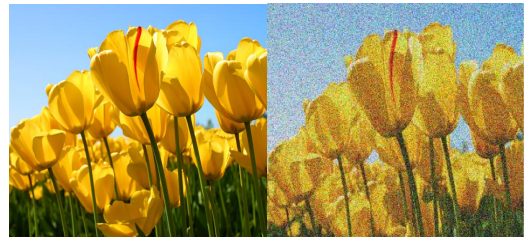


Figure 17: Output image with maximum number of pixels

V. ANALYSIS WITH VARIOUS IMAGES

A. Removal of noise from color images

Image Sample 1: Tulips



PSNR = 15.45dB

PSNR = 25.45dB

Image Sample 2: Koala

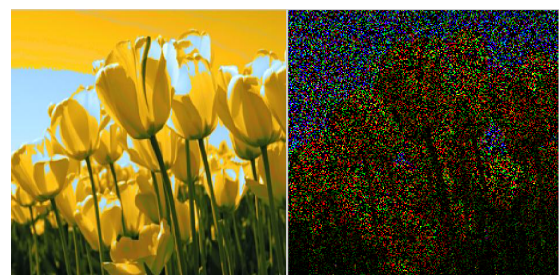


PSNR = 13.67dB

PSNR = 20.78dB

B. Restoration of missing pixels for color images

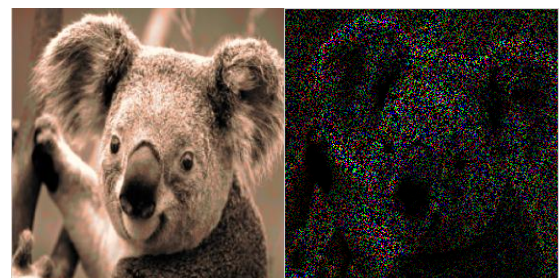
Image Sample 1: Tulips



PSNR = 10.34dB

PSNR = 14.78dB

Image Sample 2: Koala



PSNR = 12.45dB

PSNR = 15.56dB

VI. CONCLUSION AND FUTURE WORK

In this work the noise in an image is removed effectively using the method of ordering of patches and the missing pixels in an image is also recovered effectively for color images and about 70% of the color of the image was restored. The FMH (Finite impulse Median Hybrid) filters are applied in the areas of the non smooth patches in order to prevent jitter, remove impulse noise as well as to preserve edges. The decision making process for denoising or inpainting is automated to detect the image to be denoised and to be inpainted automatically and to process accordingly to remove noise by application of filters or to recover the maximum number of missing pixels. The image quality can still be improved by considering the distance between the patches in an image in the reconstruction of the subimages.

BIOGRAPHIES

Nithyabhavani. M is pursuing her Post Graduate Programme in the Department of Computer Science and Engineering in Sri Venkateswara College of Engineering.

Malathi. S.R received the Post Graduate degree in the Department of Electronics and Communication Engineering from Anna University, Chennai. She is currently working as an Associate Professor in Sri Venkateswara College of Engineering. She is currently pursuing her Ph.d degree.

REFERENCES

- [1] I. Ram, M. Elad, and I. Cohen, "Generalized tree-based wavelet transform," *IEEE Trans. Signal Process.*, vol. 59, no. 9, pp. 4199–4209, Sep. 2011.
- [2] J. Mairal, M. Elad, and G. Sapiro, "Sparse representation for color image restoration," *IEEE Trans. Image Process.*, vol. 17, no. 1, pp. 53–69, Jan. 2008.
- [3] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman, "Non-local sparse models for image restoration," in *Proc. IEEE 12th Int. Conf. Comput. Vis.*, Sep.–Oct. 2009, pp. 2272–2279.
- [4] X. Li, "Patch-based image interpolation: Algorithms and applications," in *Proc. Int. Workshop Local Non-Local Approx. Image Process.*, 2008, pp. 1–6.
- [5] I. Ram, M. Elad, and I. Cohen, "Redundant wavelets on graphs and high dimensional data clouds," *IEEE Signal Processing Letters*, vol. 19, no. 5, pp. 291–294, May 2012.
- [6] A. Buades, B. Coll, and J. M. Morel, "A review of image denoising algorithms, with a new one," *Multiscale Model. Simul.*, vol. 4, no. 2, pp. 490–530, 2006.
- [7] Idan Ram, Michael Elad, Fellow, IEEE, and Israel Cohen, Senior Member, IEEE, "Image Processing Using Smooth Ordering of its Patches", *IEEE Trans. Image Process.*, vol. 22, no. 7, Jul 2013.
- [8] P. Chatterjee and P. Milanfar, "Clustering-based denoising with locally learned dictionaries," *IEEE Trans. Image Process.*, vol. 18, no. 7, pp. 1438–1451, Jul. 2009.
- [9] G. Yu, G. Sapiro, and S. Mallat, "Image modeling and enhancement via structured sparse model selection," in *Proc. 17th IEEE Int. Conf. Image Process.*, Sep. 2010, pp. 1641–1644.
- [10] G. Yu, G. Sapiro, and S. Mallat, "Solving inverse problems with piecewise linear estimators: From Gaussian mixture models to structured sparsity," *IEEE Trans. Image Process.*, vol. 21, no. 5, pp. 2481–2499, May 2012.
- [11] W. Dong, X. Li, L. Zhang, and G. Shi, "Sparsity-based image denoising via dictionary learning and structural clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2011, pp. 457–464.
- [12] W. Dong, L. Zhang, G. Shi, and X. Wu, "Image deblurring and superresolution by adaptive sparse domain selection and adaptive regularization," *IEEE Trans. Image Process.*, vol. 20, no. 7, pp. 1838–1857, Jul. 2011.