

# Smooth Ordering of Patches and its Application

Dhanya Pious  
Dept. of Computer Science  
GEC.Idukki,Kerala,India

Philumon Joseph  
Dept. of Computer Science  
GEC.Idukki,Kerala,India

---

**Abstract:**Here a new image processing technique is used. For a given corrupted image, we extract all patches with overlaps, refer to these as coordinates in high dimensional space, and order them such that they are chained in the “shortest possible path”. The obtained ordering applied to the corrupted image implies a permutation of the image pixels to what should be a regular signal. This technique is used for image denoising, image inpainting and image deblurring. Here we include an encryption scheme using Bakers algorithm to enhance security of image under military or medical section.

**Keywords:**Inpainting, Denoising, Clustering.

---

## 1. INTRODUCTION

Image processing using local patches has become very popular and was shown to be highly effective and useful. The main idea behind these and many other algorithms are the same. When we are given the image to be processed, all possible patches with over-laps need to be extracted. Now these patches are typically very small when we are comparing it to the original image size (a typical patch size would be 88 pixels). The processing itself proceeds by operating on these patches and interrelations between them are made use. The modified patches (or sometimes only their centre pixels) are then put back into the original image canvas to form the resulting image.

There are numerous ways in which the relations between patches can be considered. Weighted averaging of pixels with similar surrounding patches, as in the case of NL-Means algorithm, clustering the patches into disjoint sets and treating each set differently, seeking a representative dictionary for the patches and using it to sparsely represent them, gathering groups of similar patches and applying a sparsifying transform on them. A common theme to many of these methods is the expectation that every patch taken from the image may find similar ones extracted elsewhere in the image.

Here the problem addressed is of reconstructing and enhancing an image given the noisy observations gathered by a digital camera sensor. There are several methods in which we view both denoising and demosaicking as image reconstruction problems, and propose a novel image model that combines two now classical techniques into a single framework. The non-local means approach to image restoration explicitly exploits self-similarities in natural

images to average out the noise among similar patches, whereas sparse coding en-codes natural image statistics by decomposing each image patch into a linear combination of a few elements from a basis set called a dictionary. The two applications implemented here are image denoising and image inpainting. Image deblurring can be considered as another application in enhancement.

## 2. PREVIOUS WORK

Let's go through few of the previous methods that have remarkably removed noise from images and reconstructed them.

### 2.1. Non Local Means Filtering

The goal of image denoising methods is to recover the original image from a noisy image,

$$v(i)=u(i)+n(i)$$

where  $v(i)$  is the observed value,  $u(i)$  is the "true" value and  $n(i)$  is the noise perturbation at a pixel  $i$ . The best simple way to model the effect of noise on a digital image is to add a Gaussian white noise. In that case,  $n(i)$  are Gaussian values with zero mean and variance  $s^2$ . Formally we define a denoising method  $D_h$  as decomposition

$$v=D_h v+n(D_h, v)$$

where  $v$  is the noisy image and  $h$  is a filtering parameter which usually depends on the standard deviation of the noise. Ideally,  $D_h v$  is smoother than  $v$  and  $n(D_h, v)$  looks like the realization of a white noise. The de-noising methods

should not alter the original image  $u$ . Now, most denoising methods degrade or remove the fine details and textures of  $u$ . To overcome the defects non-local means approach was introduced.

## 2.2. Clustering-Based Denoising With Locally Learned Dictionaries

Here algorithm aims to erase the limitations like, the static nature of the dictionary, and the constancy of the approximation order across the image. We go about this task with a clustering-based[3] algorithm which consists of three stages the clustering step where the image is clustered using features that capture the local structure of the underlying image data (patches of pixels from the image), the dictionary selection stage where we form an optimized dictionary that adapts to the geometric structure of the image patches in each cluster; and, finally, the coefficient calculation stage where the coefficients for the linear combination of dictionary atoms are estimated, subject to the (steering) kernel weights. In the following sections, we describe and motivate each of the above stages in detail.

In the initial stage, our algorithm attempts to perform clustering to identify regions of similar structure in the image. To perform clustering we need to first identify informative features from the image. While the choice of features remains an open research problem, in many cases the features are directly computed from the input image. Commonly used low level features to identify similar pixels (or patches) are pixel intensities, gradient information etc., or a combination of these. The use of such features directly from the input image is not advisable for our denoising problem due to their instability in the presence of noise. However, it has been observed that the steering weights computed in a neighbourhood are robust to the presence of significant amounts of noise. These weights are roughly representative of the under-lying local data structure. Thus, clustering is performed using feature vectors of size  $N \times 1$  for each local steering kernel computed over a  $N \times N$  window centered at pixel in the image. That is to say, every pixel of the image is mapped to a feature vector of size  $N \times 1$ . At the end of this stage we expect the image to be divided into not necessarily contiguous ( $W_k$ ) regions, each containing patches of similar structure. Hence, the entire noisy image can be thought to be composed of a union of such clusters Distance Metric: Before we proceed to perform clustering on the weights, we need to specify a metric to calculate the distance between two weight functions. Once the clusters are formed, we proceed to form a dictionary best suited to each cluster independently. For each cluster we intend to find a dictionary that best describes the structure of the under-lying data within that cluster. In other words, for each image patch in a cluster  $W_k$  we want to find an estimate

$Y_i^0$  which best approximates the input vectorized patch  $Y_i$  Coefficient Calculation Once the dictionary is formed for each cluster, we proceed to estimate the  $b_i$  parameters under a regression framework. We pose this as an optimization problem. The dictionary now is adapted to a specific class of image structure that is captured by each cluster. Furthermore, the number of principal components or dictionary atoms that will be needed to fit a prespecified percentage of data varies across the different clusters. The patches thus estimated are overlapping, so we should ideally optimally combine the overlapping regions somehow to form the final image.

## 2.3. Non-local Sparse Models for Image Restoration

Here both denoising and demosaicking as image reconstruction problems, and propose a novel image model that combines two now classical techniques into a single framework: The non-local means approach to image restoration explicitly[4],[5] exploits self-similarities in natural images to average out the noise among similar patches, whereas sparse coding encodes natural image statistics by decomposing each image patch into a linear combination of a few elements from a basis set called a dictionary. Although fixed dictionaries based on various types of wavelets have been used in this setting, sparse decompositions based on learned, possibly over complete, dictionaries adapted to specific images have been shown to provide better results in practice. We propose to extend and combine these two approaches by using simultaneous sparse coding to impose that similar patches share the same dictionary elements in their sparse decomposition. To the best of our knowledge, this is the first time that the corresponding models of image self-similarities are explicitly used in a common setting with learned dictionaries. Experiments with images corrupted by synthetic or real noise show that the proposed method outperforms the state of the art in both image denoising and image demosaicking tasks, making it possible to effectively restore raw images from digital cameras at a reasonable speed and memory cost.

## 2.4. From Patch Likelihoods to Patch Restoration

For many patch priors a[2] closed form of log likelihood, Bayesian Least Squares (BLS) and Maximum A-Posteriori (MAP) estimates can be easily calculated. Given that, we start with a simple question: Do priors that give high likelihood for natural image patches also produce good results in a restoration task such as denoising, for many

popular MRF priors, neither the log likelihood nor the MAP estimate can be calculated exactly. Here we compare several popular priors, trained over 50,000 8 X 8 patches randomly sampled from the training set. We compare the log likelihood each model gives on a set of unseen natural image patches (sampled from the test set and the performance of each model in patch denoising using MAP estimates. The models we use here are: Independent pixels with learned marginals (Ind. Pixel), Multivariate Gaussian over pixels with learned covariance (MVG), Independent PCA with learned (non-Gaussian) marginals and ICA with learned marginals. The results for each of the models can be seen in Figure As can be seen, the higher the likelihood a model gives for a set of patches, the better it is in denoising them when they are corrupted.

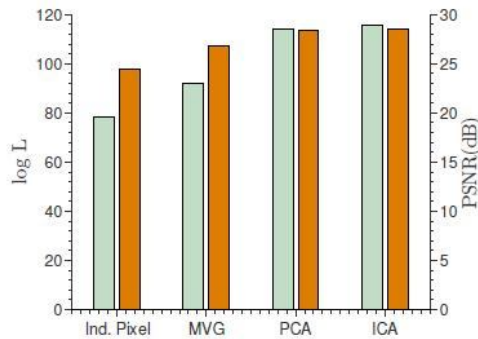


Figure 1. Likelihood of Models

### 3. IMPLEMENTATION

The smooth ordering of patches is done and this can be used for many applications. The applications are like the image denoising, image inpainting image de-blurring etc. The method for the smooth ordering of the image is reordering the patches of the image. Thus reconstructing the original image. For implementation, we have an image with us. Now we are adding some disturbances into it for making it a corrupted image. Now this is the image on which we have to apply the smooth reordering of patches and perform image denoising and image in-painting. Y is the original image and Z the image after adding impurities. Z could be having noise or it could have missing pixels. The corrupted image then satisfies

$$z = My + v$$

For reconstructing Y from z, a permutation matrix P is used. P is a N X N matrix, where N = N1 X N2. Now Y is of size N1 X N2. We assume that when P is applied to the target signal y, it produces a smooth signal

$$yp = Py$$

We will explain how such a matrix may be obtained using the image patches in Section II-B. We start by applying P to z and obtain

$$Zp = Pz$$

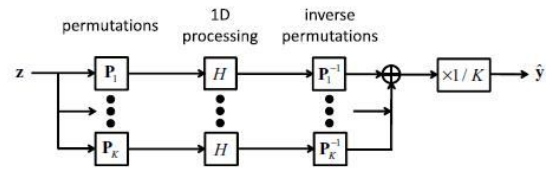


Figure 2. The basic image processing scheme

Next, we take advantage of our prior knowledge that yp should be smooth. and apply a "simple" 1D smoothing Operator H on Zp, such as 1D interpolation or filtering. Finally, we apply P1 to the result, and obtain the reconstructed image.

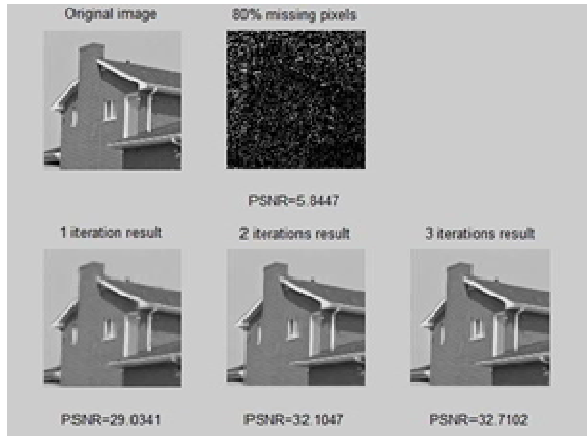
#### 3.1. Permutation matrix

To design a matrix P, that would produce a smooth signal when it is applied to the target image y is as follows.. When the image Y is known, the solution is to reorder it as a vector, and then apply a simple sort operation on the obtained vector.

#### 3.2. Image Inpainting

The problem of image inpainting is of the recovery of missing pixels in the given image. Here v = 0, and M is a diagonal matrix of size N X N which contains ones and zeroes in its main diagonal corresponding to existing and missing pixels, correspondingly. Each patch may contain missing pixels, and we denote by Si the set of indices of non-missing pixels in the patch xi. We choose the distance measure between patches xi and xj to be the average of squared differences between existing pixels that share the same location in both patches. First the matrix P is calculated. when a patch does not share pixels with any of the unvisited patches, the next patch in the path is chosen to be its nearest spatial neighbour. An operator H is used, which recovers the missing values using cubic spline interpolation. We apply the matrix P 1 on the resulting vectors and obtain the estimated subimages y j. The final estimate is obtained from these subimages. We improve our results by applying two additional iterations of a modified version of this inpainting scheme, where the only difference

is that we rebuild P using reconstructed (and thus full) patches.



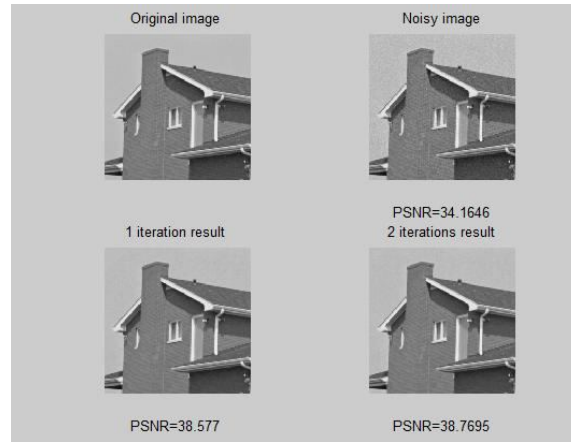
**Figure 3. The Image Inpainting**

The figure shows the original image, the corrupted one in the first row. Each iteration results of the permutation matrix are shown in the bottom row.

### 3.3 Image denoising

In image denoising, the recovery of an image from its noisy version is carried out. In that case  $M = I$  and the corrupted image satisfies  $z = y + v$ . The patches  $x_i$  may contain noise, and we choose the distance measure between  $x_i$  and  $x_j$  to be the squared Euclidean distance divided by  $n$ . A 1D linear shift invariant filter, is used for this purpose. There are two filters to switch between based on the patch content. The smooth areas in the image are treated differently than areas with edges or texture. First patches are partitioned into those smooth  $S_s$  and those with edges and texture  $S_e$ .

Next divide the sub images also into two signals. A vector of length  $|S_s|$  that contain the smooth patches and a vector of length  $|S_e|$ . Now make use of the nearest neighbour search method and extract the sub images from both divisions. Now find the filters  $h_s, h_e$  each of length  $NH$ . Now define a filter  $h$  of length  $2Nh$ . The vector  $h$  stores the filter taps to be designed. We substitute and obtain the reconstructed image.



**Figure 4. The Image Denoising**

The figure shows the image denoising scheme. The top row shows the image initially, and after adding the noise from left. In second row we get to see the results of first and the second iterations of the denoising algorithm described above. We have also included an encryption scheme in this in order to enhance the security of the image being used in the scheme.

### 3.4 Encryption

An Encryption Scheme known as Bakers Algorithm is implemented on the image after denoising and inpainting. This encryption makes these images free to be used for military purposes where the secrecy of recovered image is necessary. Encryption is implemented as follows.

1. The  $M \times M$  square matrix is divided into rectangles of width  $v_r$  and number of elements  $M$ .
2. The elements in each rectangle are rearranged to a row in the permuted rectangle. Rectangles are taken from right to left beginning with upper rectangles, and then lower ones.
3. Inside each rectangle, the scan begins from the bottom left corner towards upper elements.

## 4. CONCLUSIONS

We have proposed a new image processing scheme which is based on smooth 1D ordering of the pixels in the given image. We have shown that using a carefully designed permutation matrices and simple and intuitive 1D operations such as linear filtering and interpolation, the proposed scheme can be used for image denoising and inpainting, where it achieves high quality results. The Bakers algorithm used for encryption enhances the security of images in military, medical and many other fields where in security of image reconstructed from patches is of high importance.

There are several research directions to extend this work that we are currently considering. The first is to make use of the distances between the patches not only to find the ordering

matrices, but also in the re-construction process of the subimages. These distances carry additional information which might improve the obtained results. Improvements can also be made to the patch ordering scheme itself. We have seen in that this scheme performs poorly near the end of the found path, when only a small number of unvisited patches remain. A possible solution could be to develop a scheme which allows patches to be revisited more than once. A different direction is to develop new image processing algorithms which involve optimization problems in which the 1D image reordering act as regularizers. These may both improve the image denoising and inpainting results, and allow to tackle other applications such as image deblurring.

The Research highlights of this paper are

- New method for image denoising
- New fast method for image inpainting.
- The permutation matrix formed.
- Encryption provided for securing the image details

## 5. ACKNOWLEDGEMENTS

I thank the authors of all the previous implementation methodologies, for the fruitful discussions and advices, which helped in developing the presented work. I also thank the anonymous reviewers for their helpful comments.

## 6. REFERENCES

- [1] 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.
- [2] D. Zoran and Y. Weiss, "From learning models of natural image patches to whole image restoration," in Proc. IEEE Int. Conf. Comput. Vis., Nov. 2011, pp. 479-486.
- [3] 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.
- [4] M. Elad and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," IEEE Trans. Image Process., vol. 15, no. 12, pp. 3736-3745, Dec. 2006.
- [5] R. Zeyde, M. Elad, and M. Protter, "On single image scale-up using sparse-representations," in Proc. 7th Int. Conf. Curves Surf., 2012, pp. 711-730