

Two-Stage Image Denoising by Principal Component Analysis with Local Pixel Grouping.

(Chinese :基于主成分分析和局部像素分组的图像去噪方法研究)

Author¹: SHABAN KASSIM KILIZA

¹ SHABAN KASSIM KILIZA, Information Communication Engineering, Jiangsu University Science and Technology, Zhenjiang, China

May, 2018

摘要

本文提出了一种基于主成分分析 (PCA) 和局部像素分组 (LPG) 的高效图像去噪方案。为了更好地保留图像局部结构, 将像素及其最近的邻居建模为矢量变量, 通过使用基于块匹配的 LPG 从本地窗口中选择其训练样本。这样的 LPG 过程保证了只有内容相似的样本块在局部统计计算中用于 PCA 变换估计, 使得在 PCA 域中的系数收缩以消除噪声之后, 图像局部特征可以很好地保留。LPG-PCA 降噪过程再次迭代以进一步提高降噪性能, 并在第二阶段自适应调整噪声水平。基准测试图像的实验结果表明, 与最先进的技术相比, LPG-PCA 方法实现了非常有竞争力的去噪性能, 特别是在图像精细结构保存方面

去噪算法。

在本文中, 根据六种不同的图像滤波算法的重建噪声影响图像的能力进行比较。这些算法的目的是消除可能通过传输图像而发生的信号中的噪声。引入了一种新的算法 - 空间中值滤波器, 并将其与当前的图像平滑技术进行了比较。实验结果表明, 该算法与流行的图像平滑算法相当。另外, 引入对该算法的修改以实现比其他流行技术更精确的重构。

在本文中, 通过使用 LPG (局部像素分组) 的 PCA (主成分分析) 获得了图像中的噪声去除的有效算法。这种技术确保了图像局部结构的保存。在这里, 像素及其邻居被视为矢量变量, 其使用基于块匹配的 LPG 从本地窗口中选择训练样本。这确保只为 PCA 变换选择相似的样本, 以便仅在相当程度的降噪的情况下保留期望的局部特征。LPG-PCA 算法执行两次以提高图像的质量。第一次迭代将显著消除噪声, 第二次迭代将保留像边缘等图像特征。与 WT (小波变换) 不同, LPG-PCA 算法将自适应地调整图像的噪声水平。几个实验结果表明了该算法的有效性。

关键词: PCA (主成分分析), LPG (局部像素分组)

ABSTRACT

This paper presents an efficient image denoising scheme by using principal component analysis (PCA) with local pixel grouping (LPG). For a better preservation of image local structures, a pixel and its nearest neighbors are modeled as a vector variable, whose training samples are selected from the local window by using block matching-based LPG. Such an LPG procedure guarantees that only the sample blocks with similar contents are used in the local statistics calculation for PCA transform estimation, so that the image local features can be well preserved after coefficient shrinkage in the PCA domain to remove the noise. The LPG-PCA denoising procedure is iterated one more time to further improve the denoising performance, and the noise level is adaptively adjusted in the second stage. Experimental results on benchmark test images demonstrate that the LPG-PCA method achieves very competitive denoising performance, especially in image fine structure preservation, compared with state-of-the-art Denoising algorithms.

In this paper, six different image filtering algorithms are compared based on their ability to reconstruct noise affected images. The purpose of these algorithms is to remove noise from a signal that might occur through the transmission of an image. A new algorithm, the Spatial Median Filter, is introduced and compared with the current image smoothing techniques. Experimental results demonstrate that the proposed algorithm is comparable to popular image smoothing algorithms. In addition, a modification to this algorithm is introduced to achieve more accurate reconstructions over other popular techniques.

In this paper, an effective algorithm for noise removal in an image is obtained by using PCA (principal component analysis) with LPG (Local Pixel Grouping). This technique ensures the preservation of image local structure. Here the pixels and its neighbors are treated as vector variables whose training samples are selected from local windows using block matching based LPG. This ensures only the similar samples are selected for the PCA transformation so that the desired local characteristics are only preserved with considerable noise reduction. The LPG-PCA algorithm is performed twice to enhance the quality of an image. The first iteration would remove the noise considerably and the second iteration would preserve the image features like edges etc. The LPG-PCA algorithm will adaptively adjust the noise level of an image unlike WT (Wavelet Transformation). Several experimental results show the effectiveness of the proposed algorithm.

Keywords: *PCA (Principal Component Analysis), LPG (Local Pixel Grouping)*

CHAPTER 1 – INTRODUCTION

1.1 Background

Image analysis can be defined as inspecting images for the intention of recognizing objects and judging their importance. Image processing is a technique in which various mathematical procedures are applied to the data. It generates an enhanced image which is further useful to perform some of the analysis and detection tasks by human beings. Digital Image Processing allow the usage of computers to execute image processing algorithms on digital images to fulfil several tasks in acquisition, management, enhancement and pre-processing of images. With the fast computers and signal processors available, digital image processing is commonly in use.

Yves Meyer, Mallat and Albert Cohen [1] developed Wavelet theory, which is one of the most analytical tool used in modern areas of technical research: electronics, computers, communications, biology, medicine, astronomy and so on. In 1-D and 2-D signal processing, the main applications of wavelet theory are compression and denoising. In the first chapter the investigator presents a brief introduction to image denoising and compression, motivation for the research work and problem statement.

Image denoising is one of the important and essential components of image processing. Many scientific data sets picked by the sensors are normally contaminated by noise. It is contaminated either due to the data acquisition process, or due to naturally occurring phenomenon. There are several special cases of distortion. One of the most prevalent cases is due to the additive white gaussian noise caused by poor image acquisition or by communicating the image data through noisy channels. Other categories include impulse and speckle noises. The goal of denoising algorithm is to remove the unwanted noise while preserving the important signal features as much as possible. Noise elimination

introduce artifacts and blur in the images. So, image denoising is still a challenging task for the investigators. Several methods are being developed to perform denoising of corrupted images.

The two fundamental approaches of image denoising are the spatial filtering methods and transform domain filtering methods[2]. Spatial filters operate a low-pass filtering on a set of pixel data with an assumption that the noise reside in the higher region of the frequency spectrum.[3, 4] Spatial low-pass filters not only provide smoothing but also blur edges in signals and images. Whereas high pass filters improve the spatial resolution, and can make edges sharper, but it will also intensify the noisy background. Fourier transform domain filters in signal processing involve a trade-off between the signal-to-noise ratio (SNR) and the spatial resolution of the signal processed. Using Fast Fourier Transform (FFT), the denoising method is basically a low pass filtering procedure, in which edges of the denoised image are not as sharp as it is in the original image[5]. Due to FFT basis functions the edge information is extended across frequencies, which are not being localized in time or space. Hence low pass-filtering results in the spreading of the edges. Wavelet theory, due to the advantage of localization in time and space, results in denoising with edge preservation. The success of denoising technique is ensured by the ability of de-correlation (separation of noise and useful signal) of the different discrete wavelet transform coefficients. As the signal is contained in a small number of coefficients of such a transform, all other coefficients essentially contain noise. By filtering these coefficients, most of the noise is eliminated.

Currently there is a large proliferation of digital data. Multimedia is an evolving method of presenting many types of information[6, 7]. Multimedia combines text, sound, pictures and animation in a digital format to relate an idea. In future multimedia may be readily available as newspapers and magazines. The multimedia and other types of digital data require large memory for storage, high bandwidth for transmission and more communication time. The only means to get better on these resources is to compress the data size, so that it can be transmitted quickly and followed by decompression at the receiver. Another most significant and booming applications of the wavelet transform is image compression. More popular and efficient existing wavelet-based coding standards like JPEG2000 can easily perform better than conventional coders like Discrete Cosine Transform (DCT) and JPEG. Unlike in DCT based image compression, the effectiveness of a wavelet-based image coder depends on the choice of wavelet selection. However, different categories of images like medical images, satellite images and scanned documents do not have the same statistical properties as photographic images. The standard wavelets employed in image coders often do not match such images, resulting in lower compression and picture quality. It is significant to identify a specially adapted wavelet for non-photographic images. The goal of compression algorithm is to eliminate redundancy in the data *i.e.* the compression algorithms calculate which data is to be considered to recreate the original image along with the data to be removed.

1.2 MOTIVATION FOR THE RESEARCH WORK

After the development of continuous wavelet transform by Morlet and Grossman, many wavelet transforms (WT) have been extended their usage in image processing applications like de-noising. Wavelets are mathematical tools that decompose the data into number of different frequency components, and then studying each component with good resolution, matched to its scale. Wavelet transforms have advantages over traditional Fourier methods in analyzing the signal containing discontinuities and sharp spikes [8]. Basically, wavelet transforms are classified into continuous wavelet transform and discrete wavelet transform[9]. The digital signal processors and computes are discrete in nature, image processing algorithms use discrete wavelet transform. Wavelets perform a better-quality in image denoising, due to the sparsity and multiresolution properties. Each wavelet-based image denoising method follow three steps:

- 1) Computing a linear forward wavelet transform of the image to be DE noised,
- 2) Filtering with nonlinear thresholding in the wavelet domain.
- 3) Computing a linear inverse wavelet transform.

In signal denoising, wavelet thresholding suggested by Donoho, is a signal identification technique that make use of the properties of wavelet transform. Coefficients that are insignificant relative to some threshold can be eliminated by thresholding. The choice of a thresholding parameter determines the effectiveness of denoising algorithm. Even though the Discrete Wavelet Transform (DWT) is a powerful tool, it suffers with three limitations (shift sensitivity, poor directionality and absence of phase information), which decreased its usage in many applications. DWT is shift sensitive because it produces unpredictable changes in DWT coefficients if input signal is shifted. Next, the DWT undergo poor directionality because DWT coefficients unveil only three orientations (horizontal, vertical and diagonal). Last, absence of phase information because DWT investigation of non-stationary signals lacks the phase information.[10]proposed a redundant complex wavelet transform to avoid the above limitations in standard DWT. A

Dual-Tree Wavelet Transform (DTWT) with good directionality, approximate shift sensitivity and explicit phase information perform in excellence where redundancy is acceptable. In DTWT a pair of filter banks operate simultaneously on the input signal and furnish two wavelet decompositions. The wavelets related with filter banks form a Hilbert Transform (HT) pair and provides shift insensitivity, good directionality and explicit phase information. However, the design of DTWT filters is complex because it requires an iterative optimization over the space of ideal reconstruction filter banks. A thorough study and interest in later years showed pathway for usage of complex wavelets, and complex analytic signals particularly in signal processing and statistical applications. Further it is linked to the expansion of complex-valued discrete wavelet filters and intelligent dual filter banks. Finally, the complex wavelet transforms, directional wavelet transforms, analytic wavelets, steerable pyramids, curvelets and contourlets are intelligent and powerful redundant tools applied to signal and image analysis.

Based on the above study, it is inferred that the transform domain is better suited for image analysis.[11] A novel complex wavelet transforms(CWT) can be used for analyzing and identifying the objects in image processing applications like image denoising, compression and segmentation. Investigation results illustrate that complex wavelet transforms outperform the standard real wavelet transforms in the sense of shift-insensitivity, directionality and anti-aliasing. These features have motivated to develop Diversity Enhanced Wavelet Transform (DEDWT), Dual-Tree Complex Wavelet Transform (DTCWT), and Hyper Analytic Wavelet Transform (HWT) based image denoising methods and Huffman coding based DTCWT image compression.

1.3 PROBLEM STATEMENT

The main aim behind this thesis is estimating the recovered image from the distorted or noisy image. Though many denoising algorithms have been published, there is scope for improvement! One of the objectives of the current research work is to show that the proposed denoising algorithms based upon DWT, can be applied successfully to enhance the characteristics of noisy images by the proper selection of filtering and thresholding methods. The advantages of Complex Wavelet Transform and Hyper Analytic Wavelet Transform over real standard wavelet transforms provides more scope in the areas of image denoising and image compression[12].

The second objective of the present work, is extending the DWT implementation to Diversity Enhanced DWT. Denoising algorithms are implemented with various filtering methods[13]. A new version of Hyper Analytic Wavelet Transform (HWT) is implemented with a zero-order wiener filtering for image analysis. In the proposed HWT based method the following are the advantages of HWT. With these advantages, it is useful in denoising the corrupted images.

- 1) It allows usage of multi wavelets compared to Complex Wavelet Transform (DTCWT) where a set of predefined analysis and synthesis filters are used.
- 2) Good directionality
- 3) Quasi Shift invariance

The other objective of the work is to analyze how the Dual-Tree Complex Wavelet Transform can be applied with Huffman coding in image compression. The performance evaluation of different algorithms is based on metrics like Peak Signal to Noise Ratio (PSNR), Root Mean Square Error (RMSE), Compression Ratio (CR), Bits Per Pixel (BPP) etc.,

CHAPTER 2 – The BASIC Theory of IMAGE DENOISING

2.1 Introduction

There has been a significant amount of work done on image denoising techniques. Existing methods are able to produce good results in many practical scenarios. The various denoising techniques are as follows: Tables 2.1 shows the SNR (signal to noise ratio) of the input and output images for the filtering approach. It shows how SNR varies with different type of noise and filters used.

Table 2.1: SNR values for filtering approach

Method	SNR of input image	SNR of output image	Noise type and variance
Mean filter	18.88	27.43	Salt and pepper, 0.05
Mean filter	13.39	21.24	Gaussian, 0.05
LMS adaptive filter	18.88	28.01	Salt and pepper, 0.05
LMS adaptive filter	13.39	22.40	Gaussian, 0.05
Median filter	18.88	47.97	Sal and pepper, 0.05
Median filter	13.39	22.79	Gaussian, 0.05

2.2 Spatial Filtering

A traditional way to remove noise from image data is to employ spatial filters. Spatial filtering is commonly used to clean up the output of lasers, removing aberrations in the beam due to imperfect, dirty or damaged optics. Spatial filters can be further classified into non-linear and linear filters.

A spatial filter is essentially a beam converging device coupled with a filter. The filter, or pinhole is used to remove interference patterns in a laser beam caused by diffraction from dust, lint, lens imperfections, etc. that are part of any laser optical system. Diffraction interference degrades the laser beam by producing phase and amplitude variations or modulation on the otherwise Uniphase laser phase leading to Fresnel zone patterns in the beam[14]. The interference is removed from the beam in the following manner the laser output appears as a point source at infinity; however; the interference producing sources appear as Huygens generators a finite distance from the filter due to the difference in the point of origin, focusing the beam will produce an image of the “source” with all the “noise” or interference, defocused in an annulus around the focused beam at the pinhole; therefore, the focused beam will pass through the pinhole and the interference will be severely attenuated. Attenuations of 40dB or greater are readily produced by this filtering method.

The optimum pinhole diameter is a function of the laser wavelength, laser beam diameter and focal length of the microscope objective used[15]. They are related by

$$\text{Pinhole diameter} = \frac{8}{\pi} \times \frac{\text{Wavelength} \times \text{Focal length}}{\text{Beam diameter}} \quad (2.1)$$

Applying the above formula, we can match commercially available pinhole sizes and objectives for spatial filtering purposes. Common helium-neon (HeNe) lasers have a wavelength of 0.6328 μm and a beam diameter of 1mm, and using these parameters gives the following selection table 2.2:

Table 2.2: Pinhole sizes and objectives

Pinhole diameter	Objective	Focal length
50 μm	5x	25.5mm
25 μm	10x	14.8mm
15 μm	20x	8.3mm
10 μm	40x	4.3mm
5 μm	60x	2.9mm

In practice, a slightly larger pinhole size is preferable to one smaller than the calculated optimum size; this is reflected in the pinhole sizes above. In addition, the actual working distance between the objective and the pinhole is quite a bit smaller than the focal lengths listed above. The Spatial Filtering Procedure is shown as follow:

1) Before attaching the magnetic pinhole mount (PM) to the micrometer spindles, mount the appropriate microscope objective (MO) onto the spatial filter unit; then align the MO so that it is as close to the laser beam axis as possible. This will reduce aberrations, provide optimum light economy, and ease alignment of the pinhole.

2) Use the z-axis adjustment micrometer to move the MO as far away as possible from the x and y micrometer spindles.

3) Carefully remove the PM from its storage box, holding it by its integral handle. Do not touch the flat pinhole substrate under any circumstances. Attach the PM first to the vertical y-axis spindle making sure the machined in on the PM is against the side of the spindle opposite the MO then slide the PM towards the horizontal x-axis spindle. Before releasing the PM, make sure it is attached squarely onto both spindles.

4) While observing the output side of the PM, adjust the x and y-axes until a faint light spot is seen; be careful not to look straight into the output: always look from an angle. Place a white card near the output and adjust the x and y-axes for maximum output.

5) Slowly bring the MO closer in towards the PM with z-axis control; the output will probably drift a bit, so use the x and y controls to keep the output centered and symmetric. As the focal point of the MO is brought closer-to-the pinhole location the output will become brighter and more sensitive to adjustments.

6) Continue alternate z and x, y adjustments until the output is a smooth speckle pattern; it should look symmetric (round) with very faint or no ring patterns around it. shown in fig 2.1

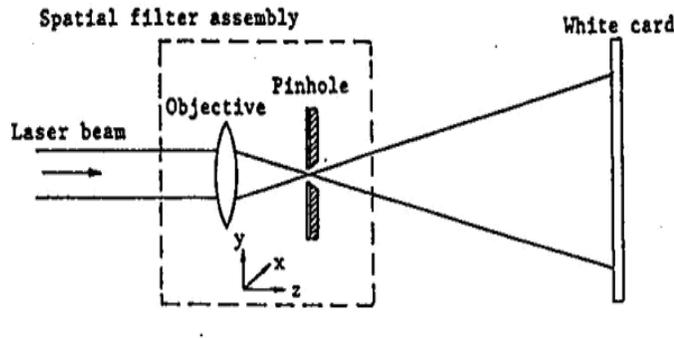


Fig 2.1: Objectives/pinhole spatial filter

2.3 Linear Filters

Linear filters process time-varying input signals to produce output signals, subject to constraint of linearity. A mean filter is the optimal linear filter for Gaussian noise in the sense of mean square error[16]. Linear filters too tend to blur sharp edges, destroy lines and other fine image details, and perform poorly in the presence of signal-dependent noise. The wiener filtering method requires the information about the spectra of the noise and the original signal and it works well only if the underlying signal is smooth. Wiener method implements spatial smoothing and its model complexity control correspond to choosing the window size.

A linear filter is an operation L which transforms a time series $X = \{X(t)$ into another time series $Y = \{Y(t)\}$,

$$Y(t) = L X(t), t \in Z. \tag{2.2}$$

Let E be the space of all stationary processes. A linear time invariant filter L is a function

$$L: \mathcal{E} \rightarrow \mathcal{E} \tag{2.3}$$

that has the following three properties:

1)Scale preservation

$$L(\alpha X) = \alpha L(X) \tag{2.4}$$

2)Superposition

$$L(X_1 + X_2) = L(X_1) + L(X_2) \tag{2.5}$$

3)Time invariance

$$L B^k X = B^k L X \quad \text{for all } k \in Z \tag{2.6}$$

where B is the backshift operator ($(BX)(t) = X(t - k)$).

$$\text{Smoothing filter: } Y(t) = \frac{1}{4}X(t + 1) + \frac{1}{2}X(t) + \frac{1}{4}X(t - 1) \tag{2.7}$$

$$\text{Differencing filter: } Y(t) = X(t) - X(t - 1) \tag{2.8}$$

2.4 Mean Filter

A mean filter acts on an image by smoothing it; that is, it reduces the intensity variation between adjacent pixels [18, 19]. The mean filter is nothing but a simple sliding window spatial filter that replaces the center value in the window with the average of all the neighboring pixel values including itself. Image corrupted with salt and pepper noise is subjected to mean filtering and it can be observed that the noise dominating is reduced. The white and dark pixel values of the noise are changed to be closer to the pixel values of the surrounding ones. Also, the brightness of the input image remains unchanged because of the use of the mask, whose coefficients sum up to the value one. The mean filter is used in applications where the noise in certain regions of the image needs to be removed. In other words, the mean filter is useful when only a part of the image needs to be processed.

2.5 LMS Adaptive Filter

Adaptive filters are capable of denoising non-stationary images, that is, images that have abrupt changes in intensity. Such filters are known for their ability in automatically tracking an unknown circumstance or when a signal is variable with little a priori knowledge about the signal to be processed. An adaptive filter does a better job of denoising images compared to the averaging filter as the Least Mean Square (LMS) adaptive filter is known for its simplicity in computation and implementation. The LMS adaptive filter works well for images corrupted with salt and pepper type noise. But this filter does a better denoising job compared to the mean filter.

The earliest work on adaptive filters may be traced back to the late 1950s, during which time a number of researchers were working independently on theories and applications of such filters. From this early work, the least-mean-square (LMS) algorithm emerged as a simple, yet effective, algorithm for the design of adaptive transversal (tapped-delay-line) filters. The LMS algorithm was devised by Widrow and Hoff in 1959 in their study of a pattern-recognition machine known as the adaptive linear element, commonly referred to as the Adaline [21, 22]. The LMS algorithm is a stochastic gradient algorithm in that it iterates each tap weight of the transversal filter in the direction of the instantaneous gradient of the squared error signal with respect to the tap weight in question. Let $\hat{w}(n)$ denote the tap-weight vector of the LMS filter, computed at iteration (time step) n . The adaptive operation of the filter is completely described by the recursive equation (assuming complex data)

$$\hat{w}(n+1) = \hat{w}(n) + \mu u(n)[d(n) - \hat{w}^H(n)u(n)]^*, \quad (2.39)$$

where $u(n)$ is the tap-input vector, $d(n)$ is the desired response, and μ is the step-size parameter. The quantity enclosed in square brackets is the error signal. The asterisk denotes complex conjugation, and the superscript H denotes Hermitian transposition (i.e., ordinary transposition combined with complex conjugation). Eq (2.36) [23] is testimony to the simplicity of the LMS filter. This simplicity, coupled with desirable properties of the LMS filter (discussed in the chapters of this book) and practical applications [5, 24], has made the LMS filter and its variants an important part of the adaptive signal processing kit of tools, not just for the past 40 years but for many years to come. Simply put, the LMS filter has withstood the test of time. Although the LMS filter is very simple in computational terms, its mathematical analysis is profoundly complicated because of its stochastic and nonlinear nature. Indeed, despite the extensive effort that has been expended in the literature to analyze the LMS filter, we still do not have a direct mathematical theory for its stability and steady-state performance, and probably we never will. Nevertheless, we do have a good understanding of its behavior in a stationary as well as a nonstationary environment, as demonstrated in the chapters of this book. The stochastic nature of the LMS filter manifests itself in the fact that in a stationary environment, and under the assumption of a small step-size parameter, the filter executes a form of Brownian motion. Specifically, the small step-size theory of the LMS filter is almost exactly described by the discrete-time version of the Langevin equation¹:

$$\begin{aligned} \Delta V_k(n) &= V_k(n+1) - V_k(n) \\ &= -\mu \lambda_k V_k(n) + \Phi_k(n), \quad k=1,2,\dots,M \end{aligned} \quad (2.40)$$

Which is naturally split into two parts: a damping force $-\mu \lambda_k V_k(n)$ and a stochastic force $\Phi_k(n)$. The terms used herein are defined as follow:

M = order (i.e., number of taps) of the transversal filter around which the LMS filter is built

λ_k = k th eigenvalue of the correlation matrix of the input vector $u(n)$, which is denoted by R

$\Phi_k(n)$ = k th component of the vector $-\mu Q^H u(n) e_o^*(n)$

Q = unitary matrix whose M columns constitute an orthogonal set of eigenvalues of the correlation matrix R

$e_o^*(n)$ = optimum error signal produced by the corresponding Wiener filter driven by the input vector $u(n)$ and the desired response $d(n)$

To illustrate the validity of Eq. (2.37) as the description of small step-size theory of the LMS filter, we present the results of a computer experiment on a classic example of adaptive equalization. The example involves an unknown linear channel whose impulse response is described by the raised cosine [25]

$$h_n = \begin{cases} \frac{1}{2}[1 + \cos(\frac{2\pi}{W}(n-2))], & n = 1, 2, 3 \\ 0, & \text{otherwise} \end{cases} \quad (2.41)$$

where the parameter W controls the amount of amplitude distortion produced by the channel, with the distortion increasing with W . Equivalently, the parameter W controls the eigenvalue spread (i.e., the ratio of the largest eigenvalue to the smallest eigenvalue) of the correlation matrix of the tap inputs of the equalizer, with the eigenvalue spread increasing with W . The equalizer has $M = 11$ taps. Figure 1 presents the learning curves of the equalizer trained using the LMS algorithm with the step-size parameter $m = 0.0075$ and varying W . Each learning curve was obtained by averaging the squared value of the error signal $e(n)$ versus the number of iterations n over an ensemble of 100 independent trials of the experiment. The Fig. 2.1 Learning curves of the LMS algorithm applied to the adaptive equalization of a communication channel whose impulse response is described by Eq. (2.38) for varying eigenvalue spreads: Theory is represented by continuous well-defined curves. Experimental results are represented by fluctuating curves.

2.6 Non-Linear Filters

An alternative approach to preserving edges while smoothing noise is to think of a filter as a statistical estimator. In particular, the goal here is to estimate the actual image value at a pixel, in the presence of noisy measurements. This view leads us to a class of filters that are hard to analyze but can be extremely useful [26].

Smoothing an image with a symmetric Gaussian kernel replaces a pixel with some weighted average of its neighbors. If an image has been corrupted with stationary additive zero-mean Gaussian noise, then this weighted average gives a reasonable estimate of the original value of the pixel. The expected noise response is zero, and the estimate has better behavior in terms of spatial frequency than a simple average.

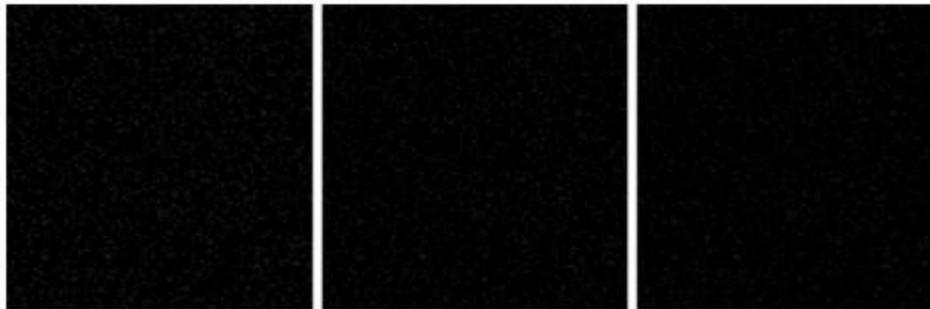


Fig 2.2 Expected Noise Response

On the left, a black background with white noise pixels distributed as a Poisson point process. These pixels are outliers, in the sense that they differ radically from their neighboring pixels. In the center image, we see the result of estimating pixels as the response of the image to a Gaussian filter with σ one pixel; we are estimating a pixel value as a weighted sum of its neighbors. Because the noise pixels are wildly different from their neighborhood, they skew this estimate substantially [27]. In the right-hand image, we see the result of using a Gaussian filter with σ two pixels; the effect remains, but is smaller, because the effective support of the filter is larger. However, if the image noise is not stationary additive Gaussian noise, difficulties arise. For example, consider a noise model where image points are set to the brightest or darkest possible value with a Poisson point process. In particular, consider a region of the image which has a constant dark value and there is a single bright pixel due to noise — smoothing with a Gaussian will leave a smooth, Gaussian-like, bright bump centered on this pixel.

The problem here is that a weighted average can be arbitrarily badly affected by very large noise values. Thus, in our example, we can make the bright bump arbitrarily bright by making the bright pixel arbitrarily bright — perhaps as result of, say, a transient error in reading a memory element. Estimators that do not have this most undesirable property are often known as robust estimates. The best-known robust estimator involves estimating the mean of a set of values using its median. For a set with $2k + 1$ elements, the median is the $k + 1$ 'th element of the sorted set of values.

For a set with $2k$ elements, the median is the average of the k and the $k + 1$ 'th element of the sorted set. It does not matter whether the set is sorted in increasing or decreasing order (exercises!).

2.7 Median filter

A median filter is specified by giving some form of neighborhood shape (which can significantly affect the behavior of the filter). This neighborhood is passed over the image as in convolution, but instead of taking a weighted sum of elements within the neighborhood, we take the media[28]

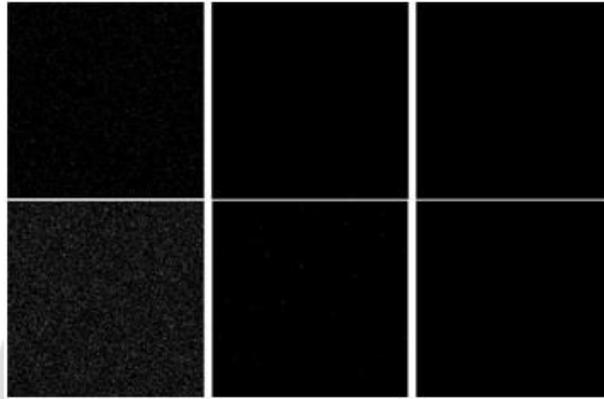


Fig 2.3: Neighborhood Shape

In fig 2.4, The columns on the left show Poisson noise processes of different intensities; on the top row, there are 2000 noise pixels and on the bottom row, 20000. The second column shows the effect of applying a filter that returns the median of a 3×3 neighborhood to these images, and the third column shows the effect of applying a filter that returns the median of a 7×7 neighborhood to these images. Notice that, if the noise is intense, then the median filter is unable to suppress it. If we write the neighborhood centered at i, j as N_{ij} , the filter can be described by:

$$y_{ij} = \text{med}(\{X_{uv} | X_{uv} \in N_{ij}\}) \quad (2.42)$$

Applying a median filter to our example of a uniform dark region with a single, arbitrarily bright, pixel will yield a dark region. In this example, up to half of the elements in the neighborhood could be noise values and the answer would still be correct. It is difficult to obtain analytic results about the behavior of median filters, but a number of general observations apply[27].

Median filters preserve straight edges but tend to behave badly at sharp corners fig2.4. This difficulty is usually dealt with by forming a multi-stage median filter; this filter responds with the median of a set of different medians, obtained in different neighborhoods:

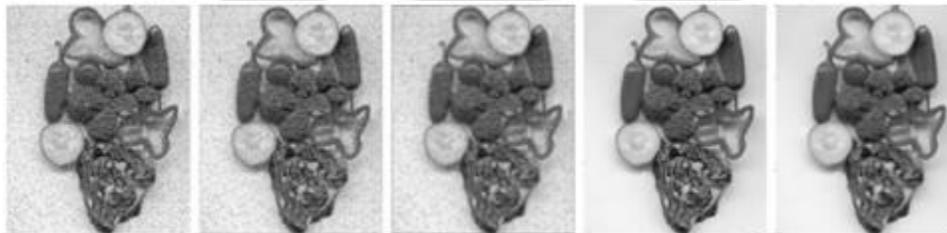


Fig 2.4 Salt and Pepper Noise Multi-Stage Median Filter

On the left, an image corrupted with salt-and-pepper noise (points are chosen by a Poisson process, and then with even probability marked either black or white; in this image, about 9% of the pixels are noise pixels). Gaussian smoothing (center left shows σ one pixel and center shows σ two pixels) works particularly poorly, as the contrast makes the dark regions left behind by averaging in dark pixels very noticeable. A median filter is much more successful

(center right shows a 3x3 median filter and right shows a 7x7 median filter). Notice how the median filter blurs boundaries.

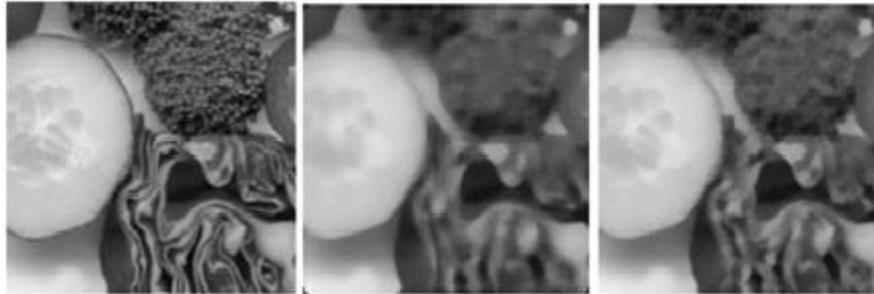


Fig 2.5 Median Filter Neighborhood Support

On the left of fig 2.6, a detail from the figure of vegetables; in the center the result of applying a median filter with a 7x7 neighborhood of support. Notice that the texture of the broccoli florets is almost completely smoothed away, and that corners around the red cabbage have been obscured. These effects could be useful in some contexts but reduce the usefulness of the filter in suppressing long-tailed noise because they represent a reduction in image detail, too. On the right, the result of applying a multistage median filter, using 7-pixel domains that are horizontal, vertical, and along the two diagonals. Significantly less detail has been lost.

While median filters tend to be better than linear filters at rejecting very large noise values so called outliers, they tend to be poorer than linear filters at handling noise that does not have outliers. In jargon, noise that can produce occasional large values is often called long-tailed noise, because the probability density for the noise values has “long tails” there is significant weight in the density far from the mean; similarly, noise that does not have this property is often called short-tailed noise. In a neighborhood, long-tailed noise will produce a small number of very large values, which tend not to affect the median much; however, short-tailed noise will produce values that are similar to the original pixel value and which will affect the median more. This difficulty can be handled either by using an α -trimmed linear filter where $\alpha/2$ percent of the largest and smallest values in a neighborhood are removed from consideration and the rest are subjected to a linear filter or by using a hybrid median filter where the output is the median of a set of linear filters over a neighborhood[29].

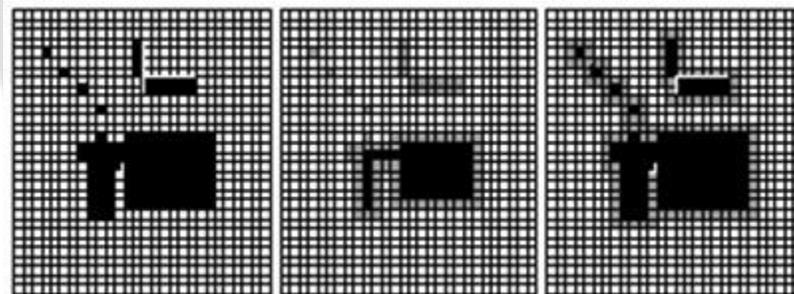


Fig 2.6 Hybrid Median Filter

On the left of fig 2.6, a binary image; a natural strategy for removing small groups of dark pixels is to lighten all pixels that do not lie at the center of a 3x3 dark neighborhood. This process is known as erosion. In the center, the relevant pixels have been greyed. Similarly, we could fill in small gaps by marking all pixels such that a 3x3 neighborhood around the pixel contacts a dark pixel, a process known as dilation. The relevant pixels have been greyed on the right.

Median filters can be extremely slow. One strategy is to pretend that a median filter is separable and apply separate x and y median filters. The best-known order-statistics filter is the median filter, which, as its name implies, replaces the value of a pixel by the median of the gray levels in the neighborhood of that pixel:

$$\hat{f}(x, y) = \text{median}_{(s,t) \in S_{xy}} \{g(s, t)\} \quad (2.43)$$

The original value of the pixel is included in the computation of the median. Median filters are quite popular because, for certain types of random noise, they provide excellent noise-reduction capabilities, with considerably less blurring than linear smoothing filters of similar size. Median filters are particularly effective in the presence of both bipolar and unipolar impulse noise. In fact, as Example above shows, the median filter yields excellent results for images corrupted by this type of noise.

Although the median filter is by far the order-statistics filter most used in image processing, it is by no means the only one. The median represents the 50th percentile of a ranked set of numbers, but the reader will recall from basic statistics that ranking lends itself to many other possibilities. For example, using the 100th percentile results in the so-called max filter given by:

$$\hat{f}(x, y) = \max_{(s,t) \in S_{xy}} \{g(s, t)\} \quad (2.44)$$

This filter is useful for finding the brightest points in an image. Also, because pepper noise has very low values, it is reduced by this filter as a result of the max selection process in the sub image area S. The 0th percentile filter is the Min filter.

$$\hat{f}(x, y) = \min_{(s,t) \in S_{xy}} \{g(s, t)\} \quad (2.45)$$

2.8 A Modified Spatial Median Filter

The Spatial Median Filter is similar to the Vector Median Filter in that in both filters, the vectors are ranked by some criteria and the top-ranking point is used to replace the center point. No consideration is made to determine if that center point is original data or not. The unfortunate drawback to using these filters is the smoothing that occurs uniformly across the image[30]. Across areas where there is no noise, original image data is removed unnecessarily in the Modified Spatial Median Filter, after the spatial depths between each point within the mask are computed, an attempt is made to use this information to first decide if the mask's center point is an uncorrupted point. If the determination is made that a point is not corrupted, then the point will not be changed. We first calculate the spatial depth of every point within the mask and then sort these spatial depths in descending order. The point with the largest spatial depth represents the Spatial Median of the set. In cases where noise is determined to exist, this representative point then replaces the point currently located under the center of the mask. The point with the smallest spatial depth will be considered the least similar point of the set[31].

By ranking these spatial depths in the set in descending order, a spatial order statistic of depth levels is created. The largest depth measures, which represent the collection of uncorrupted points, are pushed to the front of the ordered set. The smallest depth measures, representing points with the largest spatial difference among others in the mask (and possibly the most corrupted points), are pushed to the end of the list. We can prevent some of the smoothing by looking for the position of the center point in the spatial order statistic list. Let us consider a parameter T (where $1 \leq T \leq \text{mask size}$)[32], which represents the estimated number of original points under a mask of points. As stated earlier, points with high spatial depths (and supposedly uncorrupted points) are at the beginning of the list. Pixels with low spatial depths appear at the end. If the position of the center mask point appears within the first T bins of the spatial order statistic list, then we can argue that while the center point is not the best representative point of the mask, it is still original data and should not be replaced.

Two things should be noted about the use of T in this approach. When T is 1, this is the equivalent to the unmodified Spatial Median Filter. When T is equal to the size of the mask, the center point will always fall within the first T bins of the spatial order statistic and every point is determined to be original. This is the equivalent of performing no filtering at all since all of the points are left unchanged[33]. The algorithm to detect the least noisy point depends on a number of conditions. First, the uncorrupted points should outnumber, or be more similar, to the corrupted points. If two or more similar corrupted points happen in close proximity, then the algorithm will interpret the occurrence as original data and maintain the corrupted portions. While T is an estimation of the average number of uncorrupted points under a mask of points, the experimental testing made no attempt to measure the impulse noise composition of an image prior to executing the filter.

2.9 Wavelet Transform

The wavelet transform is similar to the Fourier transform (or much more to the windowed Fourier transform) with a completely different merit function[34]. The main difference is this: Fourier transform decomposes the signal into sines and cosines, i.e. the functions localized in Fourier space; in contrary, the wavelet transform uses functions that

are localized in both the real and Fourier space. Generally, the wavelet transform can be expressed by the following equation

$$F(a, b) = \int_{-\infty}^{\infty} f(x) \psi_{(a,b)}^*(x) dx \quad (2.46)$$

where the * is the complex conjugate symbol and function ψ is some function. This function can be chosen arbitrarily provided that it obeys certain rules.

As it is seen, the Wavelet transform is in fact an infinite set of various transforms, depending on the merit function used for its computation. This is the main reason, why we can hear the term “wavelet transform” in very different situations and applications. There are also many ways how to sort the types of the wavelet transforms. Here we show only the division based on the wavelet orthogonality. We can use *orthogonal wavelets* for discrete wavelet transform development and *non-orthogonal wavelets* for continuous wavelet transform development. These two transforms have the following properties:

The discrete wavelet transform returns a data vector of the same length as the input is. Usually, even in this vector many data are almost zero. This corresponds to the fact that it decomposes into a set of wavelets (functions) that are orthogonal to its translations and scaling. Therefore, we decompose such a signal to a same or lower number of the wavelet coefficient spectrum as is the number of signal data points. Such a wavelet spectrum is very good for signal processing and compression, for example, as we get no redundant information here.

The continuous wavelet transform in contrary returns an array one dimension larger than the input data. For a 1D data we obtain an image of the time-frequency plane. We can easily see the signal frequencies evolution during the duration of the signal and compare the spectrum with other signals spectra. As here is used the non-orthogonal set of wavelets, data are highly correlated, so big redundancy is seen here. This helps to see the results in a more humane form.

The discrete wavelet transforms (DWT) are an implementation of the wavelet transform using a discrete set of the wavelet scales and translations obeying some defined rules. In other words, this transform decomposes the signal into mutually orthogonal set of wavelets, which is the main difference from the continuous wavelet transform (CWT), or its implementation for the discrete time series sometimes called discrete-time continuous wavelet transform (DT-CWT)[5].

The wavelet can be constructed from a scaling function which describes its scaling properties. The restriction that the scaling functions must be orthogonal to its discrete translations implies some mathematical conditions on them which are mentioned everywhere, e.g. the dilation equation

$$\phi(x) = \sum_{k=-\infty}^{\infty} a_k \phi(Sx - k) \quad (2.47)$$

where S is a scaling factor (usually chosen as 2). Moreover, the area between the function must be normalized and scaling function must be orthogonal to its integer translations, i.e.

$$\int_{-\infty}^{\infty} \phi(x) \phi(x + l) dx = \delta_{0,l} \quad (2.48)$$

After introducing some more conditions (as the restrictions above does not produce a unique solution) we can obtain results of all these equations, i.e. the finite set of coefficients a_k that define the scaling function and also the wavelet. The wavelet is obtained from the scaling function as N where N is an even integer. The set of wavelets then forms an orthonormal basis which we use to decompose the signal. Note that usually only few of the coefficients a_k are nonzero, which simplifies the calculations. There are several types of implementations of the DWT algorithm. The oldest and most known one is the Mallat (pyramidal) algorithm. In this algorithm two filters – smoothing and non-smoothing one – are constructed from the wavelet coefficients and those filters are recurrently used to obtain data for all the scales. If the total number of data $D = 2^N$ is used and the signal length is L , first $D/2$ data at scale $L/2^{N-1}$ are computed, then $(D/2)/2$ data at scale $L/2^{N-2}$, ... up to finally obtaining 2 data at scale $L/2$. The result of this algorithm is an array of the same length as the input one, where the data are usually sorted from the largest scales to the smallest ones. Within Gwyddion the pyramidal algorithm is used for computing the discrete wavelet transform. Discrete wavelet transform in 2D can be accessed using DWT module. Discrete wavelet transform can be used for easy and fast denoising of a noisy signal[35]. If we take only a limited number of highest coefficients of the discrete wavelet transform spectrum, and we perform an inverse transform (with the same wavelet basis) we can obtain more or less denoised signal. There are several ways how to choose the coefficients that will be kept. Within Gwyddion, the

universal thresholding, scale adaptive thresholding and scale and space adaptive thresholding is implemented. For threshold determination within these methods, we first determine the noise variance guess given by

$$\hat{\sigma} = \frac{\text{Median}|Y_{i,j}|}{0.6745} \tag{2.49}$$

where Y_{ij} corresponds to all the coefficients of the highest scale sub band of the decomposition (where most of the noise is assumed to be present). Alternatively, the noise variance can be obtained in an independent way, for example from the AFM signal variance while not scanning. For the highest frequency sub band (universal thresholding) or for each sub band (for scale adaptive thresholding) or for each pixel neighborhood within sub band (for scale and space adaptive thresholding) the variance is computed as

$$\hat{\sigma}_Y^2 = \frac{1}{n^2} \sum_{i,j=1}^n Y_{i,j}^2$$

Threshold value is finally computed as

$$T(\hat{\sigma}_X) = \hat{\sigma}^2 / \hat{\sigma}_X \tag{2.50}$$

where

$$\hat{\sigma}_X = \sqrt{\max(\hat{\sigma}_Y^2 - \hat{\sigma}^2, 0)}$$

When threshold for given, scale is known, we can remove all the coefficients smaller than threshold value (hard thresholding) or we can lower the absolute value of these coefficients by threshold value (soft thresholding).

CHAPTER 3 – Principal Component Analysis Image Denoising with Local Pixel Grouping

3.1 Principal Component Analysis (PCA)

Let $X = [x_1, x_2, x_3, \dots, x_m]^T$ denote an m component vector in its transpose form. This is denoted as

$$X = \begin{bmatrix} x_1^1 & x_1^2 & \dots & x_1^n \\ x_2^1 & x_2^2 & \dots & x_2^n \\ \vdots & \vdots & \dots & \vdots \\ x_m^1 & x_m^2 & \dots & x_m^n \end{bmatrix} \tag{3.1}$$

The sample matrix of X where X_{ij} where $j=1,2,\dots, n$ represent the discrete sample variables of the sample vector X_i where $i=1,2,3,\dots, m$. The mean value of sample vector is calculated as

$$\mu = \frac{1}{n} \sum_{j=1}^n X_t(j) \tag{3.2}$$

Thus, the average value of the pixels is computed using the above equation. The sample vector is modified and centralized as follows

$$X_t = X_t - \mu = \langle X_t^1 | X_t^2 | \dots | X_t^n \rangle$$

Finally, we calculate the covariance matrix using the formula

$$\Omega = \frac{1}{n} X X^T \tag{3.3}$$

The goal of PCA is to generate an orthogonal transformation matrix P to de-correlate the matrix such that $Y = PX$, such that the covariance matrix of Y is diagonal[24]. Since the covariance matrix is symmetrical it can be written as

$$\Omega = \theta\Delta\theta^T \tag{3.4}$$

Where Ω is $m \times m$ eigenvector matrix and Δ is a diagonal eigenvector matrix. Based on the eigen values the orthogonal transformation matrix P is given as

$$P = \theta^T \tag{3.5}$$

Thus, the matrix X can be de-correlated using the orthogonal transformation matrix P as $Y=PX$

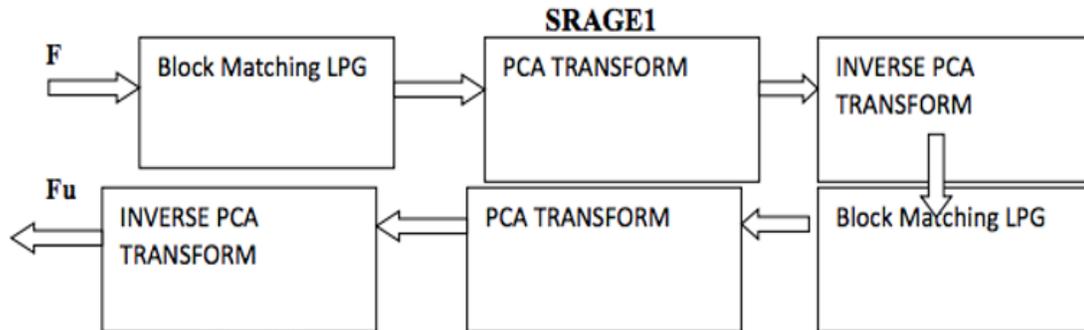


Fig 3.1 Stage implementation of LPG-PCA Algorithm

3.2 PCA algorithm with Local Pixel Grouping (LPG)

Here we assume that the noise (u) in the image is additive, with zero mean and standard deviation σ . Let this noise be added to the original image say F . Therefore, the new image value is determined as $F_u = F + u$. The goal of our project is to find an image F^1 which is approximately equal to the original image F . Pixels are identified based on the spatial coordinates and their grey scale value (intensity value) whereas of different intensity values. Here we assume the pixels in local structure as vectors and improve the edge preservation process. The image F and noise u are uncorrelated[24]. For removing noise from an underlying pixel, according to the fig, a $K \times K$ matrix centered on the pixel and denote by $X = [x_1, x_2, \dots, x_m]^T$ with total no of elements $m = k^2$. The window is centered on the image X . Since the image is prone to noise u we represent the new image vector as $X_u = X + u$. The noisy image where $U = [u_1, u_2, \dots, u_m]^T$. The statistical PCA is used on these vectors. To remove the noise from an image the covariance matrix X_u and PCA transformation matrix are to be calculated. Therefore, we use a LL training block centered on X_u , such that $L \times L$ is greater than $K \times K$. From the training block, we need to estimate the required pixels for the PCA. This selection of different pixels from training blocks is a complex process and may sometimes leads to inaccurate results.

Here selecting the training samples similar to the $K \times K$ central block from the given $L \times L$ training block is achieved using block matching based technique. The total number of samples available are $(L - K + 1)^2$ training samples of X_u in the $L \times L$ training window. In the $L \times L$ training block, let x_0 denote the vector containing sample pixels in the center $K \times K$ block and x_i represent the pixels of $(L - K + 1)^2 - 1$ [38].

In the $m \times n$ dataset matrix \bar{X}_u , each component x_k^u $k = 1, 2, \dots, m$, of the vector variable x_u has n samples. Denote by x_k^u the row vector containing the n samples of x_k^u . The n the dataset x_u can be represented as $X_u = [(X_1^u)^T \dots (X_m^u)^T]^T$. Similarly, we have $X = [X_1^T \dots X_m^T]^T$, where X_k is the row vector containing the n samples of x_k and $X_u = X + V$, [39]. Where $V = [V_1^T \dots V_m^T]^T$. Similarly, is the dataset noise variable u and V_k is the sample vector u_k

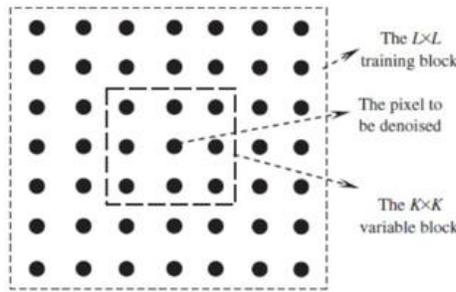


Fig 3.2 Illustration of the modeling of LPG-PCA based denoising

Next, we centralized a set X_u . The mean value of X_k^u is $u_k = \sum_{i=1}^n X_k^u(i)$, and then X_k^u is centralized by $\bar{X}_k^u = X_k^u - u_k$. Since the noise u_k is Zero-mean, X_k can also be centralized by $\bar{X}_k = X_u - u_k$. Then centralized of X_u and X are obtained as $\bar{X}_u = [(\bar{X}_1^u)^T \dots (\bar{X}_m^u)^T]^T$ and $\bar{X}_k = [\bar{X}_{k_1}^T \dots \bar{X}_{k_m}^T]^T$, and we have $\bar{X}_u = \bar{X} + V$. By computing the covariance matrix of \bar{X} denoted by $\Omega_{\bar{x}}$, then PCA transformation matrix $P_{\bar{x}}$ can be obtained. $\Omega_{\bar{x}_u} = \frac{1}{n} \bar{X}_u \bar{X}_u^T \approx \frac{1}{n} (\bar{X} \bar{X}^T + \bar{X} V^T + V \bar{X}^T + V V^T)$

Since \bar{X} and V are uncorrelated, items $\bar{X} V^T$ and $V \bar{X}^T$ will be nearly zero matrices and thus

$$\Omega_{\bar{x}_u} \approx \frac{1}{n} (\bar{X} \bar{X}^T + V V^T) = \Omega_{\bar{x}} + \Omega_u \tag{3.6}$$

Where $\Omega_{\bar{x}} = \left(\frac{1}{n}\right) \bar{X} \bar{X}^T$ and $\Omega_u = \left(\frac{1}{n}\right) V V^T$, $\Omega_u(i, j)$ is the correlation between u_i and u_j . since u_i and u_j are uncorrelated for $i \neq j$, we know that Ω_u is a $m \times m$ diagonal matrix with all the diagonal components being σ^2 . In other words, Ω_u can be written as $\sigma^2 I$, where I is identity matrix. Then it can be readily proved that the PCA transformation matrix $P_{\bar{x}}$ associated with $\Omega_{\bar{x}}$ is same as PCA transformation matrix associated with $\Omega_{\bar{x}_u}$. Since $\Omega_{\bar{x}}$ is written as

$$\Omega_{\bar{x}} = \phi_{\bar{x}} \Lambda_{\bar{x}} \phi_{\bar{x}}^T \tag{3.7}$$

Where $\phi_{\bar{x}}$ is the $m \times m$ $\Lambda_{\bar{x}}$ is the diagonal eigen matrix. Since $\phi_{\bar{x}}$ is an orthonormal matrix, we can write

$$\Omega_u = (\sigma^2 I) \phi_{\bar{x}} \phi_{\bar{x}}^T = \phi_{\bar{x}} (\sigma^2 I) \phi_{\bar{x}}^T = \phi_{\bar{x}} \Omega_u \phi_{\bar{x}}^T$$

Thus, we have

$$\begin{aligned} \Omega_{\bar{x}_u} &= \Omega_{\bar{x}} \Omega_u = \phi_{\bar{x}} \Lambda_{\bar{x}} \phi_{\bar{x}}^T + (\sigma^2 I) \phi_{\bar{x}} \phi_{\bar{x}}^T \\ &= \phi_{\bar{x}} (\Lambda_{\bar{x}} + \sigma^2 I) \phi_{\bar{x}}^T = \phi_{\bar{x}} (\Lambda_{\bar{x}_u}) \phi_{\bar{x}}^T \end{aligned} \tag{3.8}$$

Where $\bar{x}_u = \Lambda_{\bar{x}} + \sigma^2 I$. above equation implies that $\Omega_{\bar{x}_u}$ and $\Omega_{\bar{x}}$ have the same eigen vector matrix $\phi_{\bar{x}}$. Thus, impractical implementation we can directly compute $\phi_{\bar{x}}$ by decomposing $\Omega_{\bar{x}_u}$, instead of $\phi_{\bar{x}}$, and the orthonormal PCA transformation matrix for \bar{X} is set as

$$P_{\bar{x}} = \phi_{\bar{x}}^T \tag{3.9}$$

Applying $P_{\bar{x}}$ to dataset X_u we have

$$\bar{Y}_u = P_{\bar{x}} \bar{X}_u + P_{\bar{x}} V = \bar{Y} + V_y \tag{3.10}$$

Where $\bar{Y} = P_{\bar{x}} \bar{X}$ is the de-correlated dataset for \bar{X} and $V_y = P_{\bar{x}} V$ is the transformed noise dataset for V . since $\bar{Y} + V_y$ noise is uncorrelated, we can easily derive that the covariance matrix of \bar{Y}_u is

$$\Omega_{\bar{y}_u} = \frac{1}{n} \bar{Y}_u \bar{Y}_u^T = \Omega_{\bar{y}} + \Omega_{u_y} \tag{3.11}$$

Where $\Omega_{\bar{y}} = \Lambda_{\bar{x}}$ is the covariance matrix of de-correlated dataset \bar{Y} and $\Omega_{u,y} = P_{\bar{x}} \Omega_u P_{\bar{x}}^T$ is the covariance matrix of noise dataset V_y . In the PCA transformed domain \bar{Y}_u , most energy of noiseless dataset \bar{Y} concentrate on the most important components, while the energy of noise V_y distributes much more evenly. The noise in \bar{Y}_u can be suppressed by using linear minimum mean square-error estimation (LMMSE) technique. Since \bar{Y}_u is centralized, the LMMSE of \bar{Y}_k , i.e the kth row of \bar{Y} , is obtained as $\hat{\bar{Y}}_k = w_k \cdot \bar{Y}_u^{k-k}$ Where the shrinkage coefficient $w_k = \frac{\Omega_{\bar{y}}(k,k)}{\Omega_{\bar{y}}(k,k) + \Omega_{u,y}(k,k)}$. And \bar{Y}_u^{k-k} is the Kth row of \bar{Y}_u . In flat Zone, $\Omega_{\bar{y}}(k,k)$ is much smaller than $\Omega_{u,y}(k,k)$ so that w_k is close to 0. Hence most of the noise will be exactly 0 and all the noise \bar{Y}_k by LMMSE operator $\hat{\bar{Y}}_k = w_k \cdot \bar{Y}_u^{k-k}$. In implementation, we first calculate $\Omega_{\bar{y}u}$ from the variable noisy dataset \bar{Y}_u and then estimate $\Omega_{\bar{y}}(k,k)$ by $\Omega_{\bar{y}}(k,k) = \Omega_{\bar{y}u}(k,k) - \Omega_{u,y}(k,k)$. In flat zones, it is often that $\Omega_{\bar{y}u}(k,k) - \Omega_{u,y}(k,k) \leq 0$, and then we set $\Omega_{\bar{y}}(k,k) = 0$. In this case w_k will be exactly 0 and all the noise in \bar{Y}_u^{k-k} is removed. Denoted by $\hat{\bar{Y}}$ the matrix of all $\hat{\bar{Y}}_k$. By transforming $\hat{\bar{Y}}$ back to the time domain, we obtain the de-noise result of \bar{X}_u as $\hat{\bar{X}} = \hat{\bar{Y}} \cdot P_{\bar{x}}$. We used the fact that $P_{\bar{x}}^{-1} = P_{\bar{x}}^T$. Adding the mean values μ_k back to $\hat{\bar{X}}$ gives the de-noise dataset \hat{X} . The estimation of the central block \hat{x}_u , denote \hat{x}_u , can then be extracted from \hat{X} and finally the denoised result of the underlying central block \hat{x}_u . Applying the above procedure to each pixel leads to the full de-noised image of I_u .

Table 3.1: PSNR Images

IMAGE	Psnr1	Psnr2	Ssim1	Ssim2	Stage (3) Psnr	Stage (3) ssim
Image1(House)	32.224	33.082	0.809	0.867	33.1	0.87
Image2(Lena)	30.77	32.46	0.7887	0.8777	32.47	0.89
Image3(yellow)	31.661	33.576	0.7737	0.888	33.59	0.888



Fig 3.9 (a)original image (b)Noisy image (c)Stage 1 LPG-PCA output (d)Stage 2 LPG-PCA output

Most of the noise will be removed by using the denoising procedures described in Sections 3.1–3.3. However, there is still much visually unpleasant noise residual in the denoised image. Table 3.1 shows an example. fig 3.8a is the original image Cameraman; fig 3.8b is the noisy version of it (s=20, PSNR= 22.1 dB); fig3.9a is the denoised

image (PSNR=29.8 dB) by using the proposed LPG-PCA method[40] . Although the PSNR is much improved, we can still see much noise residual in the denoising output.

There are mainly two reasons for the noise residual. First, because of the strong noise in the original dataset X_t , the covariance matrix X_{xt} is much noise corrupted, which leads to estimation bias of the PCA transformation matrix and hence deteriorates the denoising performance; second, the strong noise in the original dataset will also lead to LPG errors, which consequently results in estimation bias of the covariance matrix X_x (or X_{xt}). Therefore, it is necessary to further process the denoising output for a better noise reduction. Since the noise has been much removed in the first round of LPG-PCA denoising, the LPG accuracy and the estimation of X_x (or X_{xt}) can be much improved with the denoised image[41]. Thus, we can implement the LPG-PCA denoising procedure for the second round to enhance the denoising results.

As shown in fig 3.3a the noise should be updated in the second stage of LPG-PCA denoising Algorithm. Denote by \hat{l} the denoised version of noisy image l_v in the first stage[2]. We write as $\hat{l}=l+v_s$ where v_s is the residual in the denoised image. We need to estimate the level of v_s , denoted by $\sigma^2=\sqrt{E[v_s^2]}$ and input it to the second stage LPG-PCA denoising. Here we estimate σ^2 based on the difference between \hat{l} and l_v let

$$\hat{l} = l_v - \hat{l} = v - v_s$$

we have:

$$E[l^2]=E[v^2]+E[v_s^2]-2E[v-v_s]=\sigma^2+\sigma_s^2-2E[v-v_s]$$

we approximately view v_s as smoothed version of noise v , and it contains mainly the low frequency component v . Let $\bar{v} = v - v_s$ be their difference and \bar{v} contains mainly the high frequency component of v . there is $E[v - v_s] = E[\bar{v} - v_s] + E[v_s^2]$.

In general, $E[\bar{v} - v_s]$ is much smaller compared with $E[v_s^2]$. For example, after the first stage denoising of noisy image cameraman ($\sigma = 20$). we have $E[v_s^2]=72$ and $E[\bar{v} - v_s] = 17$. For the convenience of development, we remove $E[\bar{v} - v_s]$ from $E[v - v_s]$ and let

$$E[v - v_s] = E[\bar{v} - v_s] + E[v_s^2] \approx E[v_s^2]=\sigma_s^2$$

Thus, from the above equation we have

$$\sigma_s^2 \approx \sigma^2 - E$$

In practical v_s will include not only the noise residual but also the estimation error of noiseless image l . therefore, in implementation we let $\sigma_s = c_s \cdot \sqrt{\sigma^2 - E[l^2]}$

Where $c_s < 1$ is a constant, we experimentally found that setting c_s around 0.35 lead to satisfying denoising results for most of the testing images, shows fig 3.8d shows the denoising results (PSNR = 30.1dB) of cameraman after the second stage. Although the PSNR is not improved too much on this image, we can clearly see that the visual quality is much improved by effectively removing the noise residual in the first stage.

3.3 Color images denoising based on LPG-PCA

There are two approaches to extending the proposed LPG-PCA algorithm to color images. The first approach is to apply separately LPG-PCA to each of the red, green and blue channels. This approach is simple to implement but it ignores the spectral correlation in the color image. The second approach is to form a $K \times 3$ color variable cube with each $K \times K$ variable block corresponding to the red, green or blue channel. Like in the denoising of grey level image, the color variable cube is stretched to a color variable vector of dimension $3K^2$. Then the training samples of the color variable vector are selected in the local $L \times 3$ window using the LPG procedure. All the other steps are the same as those in the LPG-PCA denoising of grey level images.

Compared with the first approach, the second approach can exploit both the spatial correlation and the spectral correlation in denoising color images. However, there are two main problems. First, the dimensionality of the color variable vector is three times that of the gray level image, and this will increase significantly the computational cost in the PCA denoising process. Second, the high dimensionality of the color variable vector requires much more training samples to be found in the LPG processing. Nonetheless, we may not be able to find enough training samples

in the local neighborhood so that the covariance matrix of the color variable vector may not be accurately estimated, and hence the denoising performance can be reduced. With the above consideration, in this paper we choose the first approach for LPG-PCA based color image denoising due to its simplicity and robustness.

Chapter 4: Experimental Results and Discussions

4.1 Introduction

In the proposed LPG-PCA denoising algorithm, most of the computational cost spends on LPG grouping and PCA transformation, and thus the complexity mainly depends on two parameters: the size K of the variable block and the size L of training block. In LPG grouping, it requires $(2K^2 - 1)(L - K + 1)^2$ additions, $K^2(L - K + 1)^2$ multiplications and $(L - K + 1)^2$ “less than” logic operations. Suppose in average S training samples are selected, i.e.



Fig 4. 1: The test images Lena, Cameraman, Barbara, Peppers, House, Blood cell, Paint, Monarch, Tower (color) and Parrot (Color)

4.2 Experimental Results

Table 4.1: The PSNR (dB) and SSIM results of the denoised images in the two stages by the proposed LPG-PCA method.

Images	Lena	Cameraman	House	Paint	Monarch
First Stage					
S=10	33.60(0.9218)	33.9(0.9261)	35.4(0.9003)	33.5(0.9280)	34.0(0.9522)
s=20	29.5(0.8346)	29.8(0.8320)	31.8(0.8084)	29.3(0.8440)	29.6(0.8859)
s=30	27.1(0.7441)	27.3(0.7395)	29.3(0.7225)	26.8(0.7467)	27.0(0.8071)
s=40	25.4(0.6597)	25.5(0.6393)	27.3(0.6243)	25.0(0.6590)	25.2(0.7267)
Second Stage					
s=10	33.7(0.9243)	34.1(0.9356)	35.6(0.9012)	33.6(0.9311)	34.2(0.9594)
s=20	29.7(0.8605)	30.1(0.8902)	32.5(0.8471)	29.5(0.8683)	30.0(0.9202)
s=30	27.6(0.8066)	27.8(0.8558)	30.4(0.8185)	27.2(0.8088)	27.4(0.8769)
s=40	26.0(0.7578)	26.2(0.8211)	28.9(0.7902)	25.6(0.7569)	25.9(0.8378)

Table 4.2: The PSNR (dB) and SSIM results of the denoised images in the two stages by the proposed LPG-PCA method.

Images	Lena	Cameraman	House	Paint	Monarch
First Stage					
s=10	32.5(0.9357)	33.4(0.8909)	34.6(0.9137)	34.7(0.9047)	34.5(0.9198)
s=20	28.3(0.8530)	29.9(0.8177)	31.3(0.8587)	30.6(0.7922)	30.6(0.8337)
s=30	26.0(0.7663)	27.5(0.7332)	28.6(0.7864)	28.3(0.6772)	28.2(0.7434)
s=40	24.2(0.6741)	25.9(0.6447)	26.7(0.7076)	26.6(0.5718)	26.3(0.6564)
Second Stage					
s=10	32.5(0.9378)	33.3(0.8943)	34.8(0.9173)	34.8(0.9123)	34.6(0.9255)
s=20	28.5(0.8716)	30.1(0.8413)	32.0(0.8836)	31.1(0.8522)	31.1(0.8776)
s=30	26.2(0.8028)	27.9(0.7973)	29.6(0.8538)	29.1(0.8069)	29.0(0.8415)
s=40	24.5(0.7378)	26.7(0.7648)	28.0(0.8239)	27.8(0.7695)	27.5(0.8097)

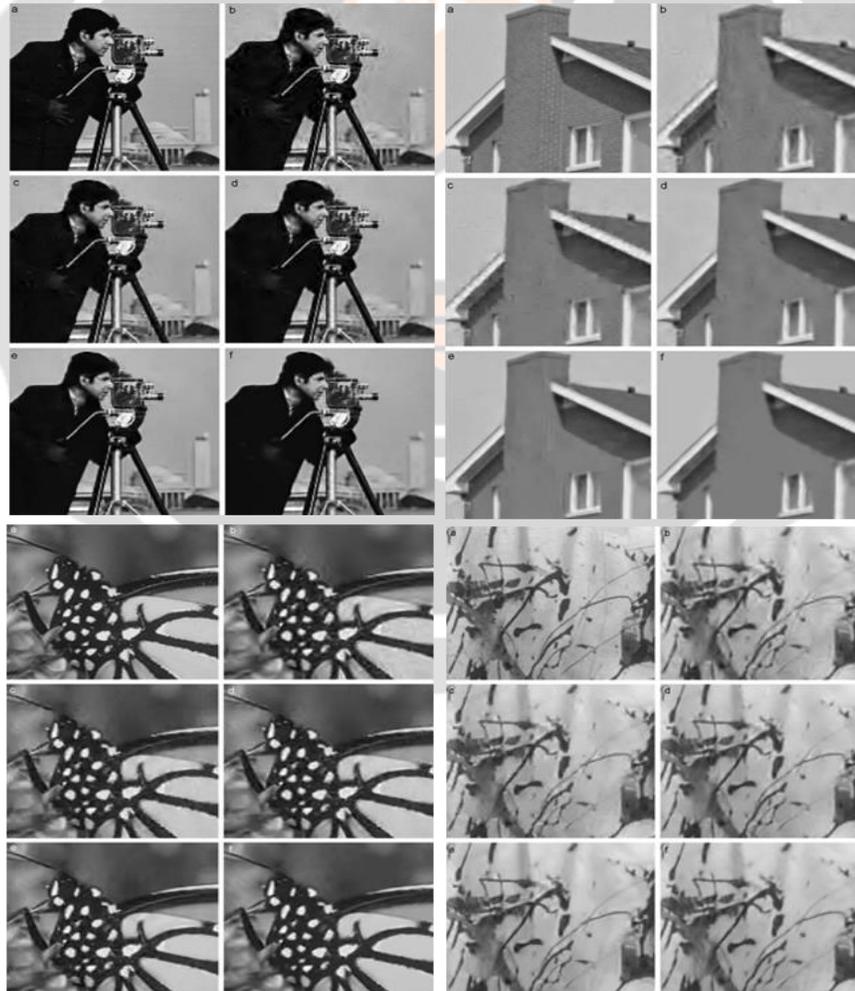
The value in the parenthesis is the SSIM measure the dataset X_t is of dimension $K^2 \times S$. Then in the PCA transformation, it requires $K^2 \times S + (S^2 - 1) \times K^4 + (K^2 - 1) \times K^2 \times S$ additions, $K^4 \times (S + S^2)$ multiplications, and an SVD decomposition of a $K^2 \times K^2$ definite covariance matrix. In this paper, we set $K = 5$ and $L = 41$ in all the experiments to test the denoising performance. The threshold T in the LPG grouping is set to 25. In the implementation of LPG-PCA denoising, actually the complete $K \times K$ block centered on the given pixel will be denoised. Therefore, the finally restored value at a pixel can be set as the average of all the estimates obtained by all windows containing the pixel. This strategy was also used in [42]. By our experiments, this can increase about 0.3 dB the noise reduction for most of the test images.

The proposed LPG-PCA algorithm can be viewed as a completion and extension of the PCA-based denoising algorithm in [42]. We compare LPG-PCA with four representative and state-of-the-art denoising algorithms: the wavelet-based denoising methods [40, 43], the sparse representation based K-SVD denoising method [44] and the recently developed BM3D denoising method [24, 45]. The BM3D method is one of the best denoising methods and it has been viewed as a benchmark for denoising algorithm evaluation. The ten test images (size: 256 × 256) used in the experiments, including eight grey level images and two-color images, are shown in fig 4.1. We added Gaussian white noise of different levels ($s = 10, 20, 30$ and 40 , respectively,) to the original image and use the five denoising algorithms for noise removal. Due to the limitation of space, in this paper we can only show partial denoising results.

We evaluate and compare the different methods by using two measures: PSNR and SSIM. Although PSNR can measure the intensity difference between two images, it is well-known that it may fail to describe the visual perception quality of the image. On the other hand, how to evaluate the visual quality of an image is a very difficult problem and it is currently an active research topic. The SSIM index proposed in, is one of the most commonly used measures for image visual quality assessment. Compared with PSNR, SSIM can better reflect the structure similarity between the target image and the reference image. We first verify the improvement of the noise removal in the second stage of the PLG-PCA method. Table 4.1 lists the PSNR and SSIM measures of the first stage and second stage denoising outputs

on the test image set. We can see that the second stage can improve 0.1–1.5 dB the PSNR values for different images under different noise level (s is from 10 to 40). Although for some images the second stage will not improve much the PSNR measures, the SSIM measures, which can better reflect the image visual quality, can be much improved. For instance, for image Lena with noise level $s=30$, the SSIM measure is much increased from 0.7441 to 0.8066 after the second stage denoising, while the PSNR is increased by only 0.5 dB.

We then compare the different methods on denoising. Table 4.2 list the PSNR and SSIM results by different methods on the 10 test images. Let's first see the PSNR measures by different methods. From Table 4.2 we see that the algorithm BM3D has the highest PSNR measures. This is because it sufficiently exploits the non-local redundancies in the image. The K-SVD algorithm uses a pre-trained over-complete dictionary in the denoising process and it achieves almost the same PSNR results as those by the proposed LPG-PCA algorithm. The PSNR result of LPG-PCA is higher than the wavelet-based methods [40, 43], and the wavelet-based method [40] has the lowest PSNR value.



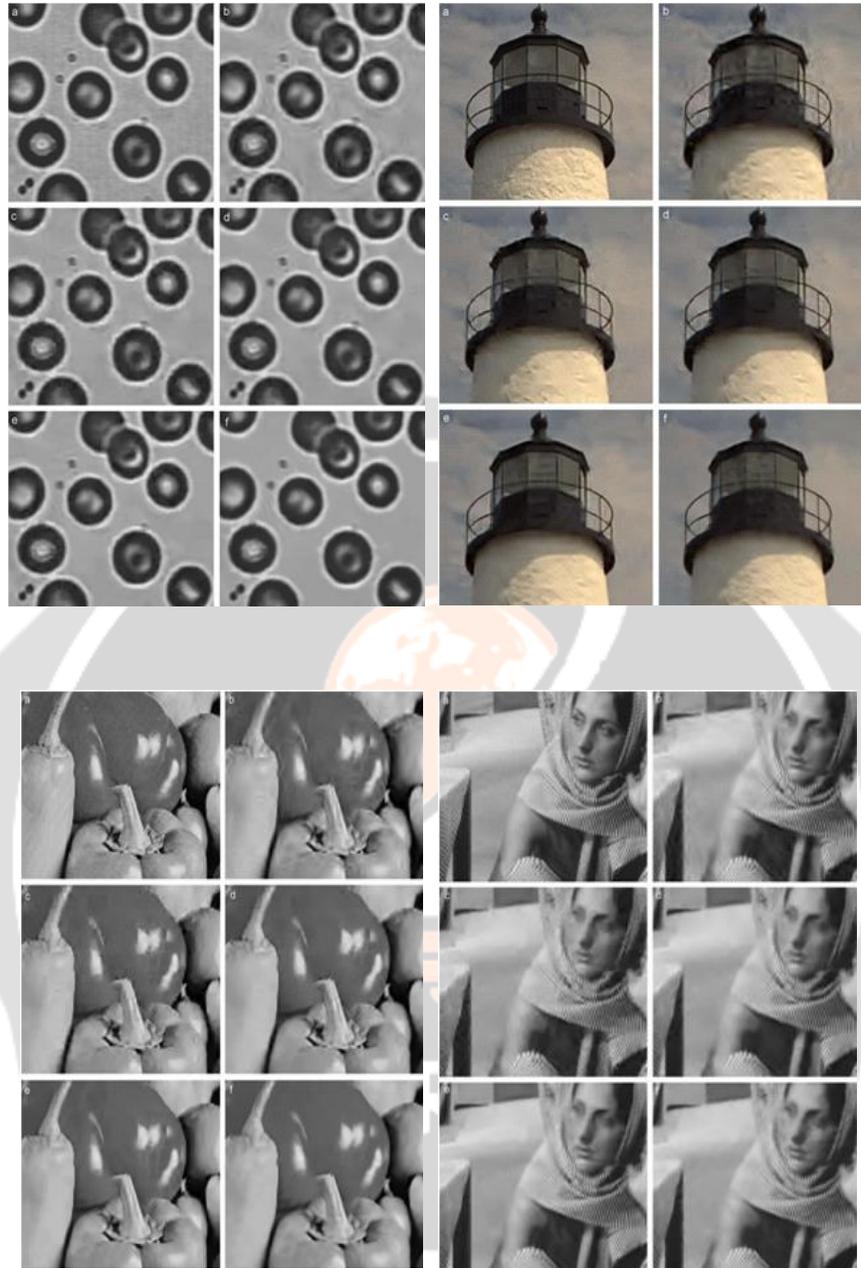


Fig 4. 2: The denoising results of different methods

Table 4.3: The PSNR (dB) and SSIM results of the denoised images at different noise levels and by different schemes

Methods	[40]	[43]	[44]	[45]	Proposed
Lena					
s= 10	33.1(0.9154)	33.2(0.9160)	33.5(0.9203)	33.9(0.9272)	33.7(0.9243)
s= 20	29.2(0.8455)	29.4(0.8514)	29.7(0.8571)	30.2(0.8699)	29.7(0.8605)
s= 30	27.2(0.7878)	27.5(0.7964)	27.8(0.8055)	28.3(0.8231)	27.6(0.8066)
s= 40	25.7(0.7315)	26.0(0.7466)	26.2(0.7504)	27.3(0.7727)	26.0(0.7578)
Cameraman					
s= 10	33.2(0.9170)	33.7(0.9307)	33.9(0.9334)	34.4(0.9399)	34.1(0.9356)

s= 20	29.1(0.8449)	29.6(0.8744)	29.9(0.8810)	30.6(0.8962)	30.1(0.8902)
s= 30	26.8(0.7945)	27.5(0.8307)	27.9(0.8426)	28.5(0.8655)	27.8(0.8558)
s= 40	25.3(0.7310)	26.0(0.7806)	26.5(0.8048)	27.1(0.8303)	26.2(0.8211)
House					
s= 10	34.4(0.8791)	34.8(0.8809)	35.5(0.8960)	36.2(0.9143)	35.6(0.9012)
s= 20	31.3(0.8199)	32.1(0.8374)	32.7(0.8458)	33.3(0.8553)	32.5(0.8471)
s= 30	29.4(0.7829)	30.2(0.8066)	30.7(0.8137)	31.6(0.8319)	30.4(0.8185)
s= 40	28.1(0.7409)	28.9(0.7708)	29.1(0.7771)	30.7(0.8065)	28.9(0.7891)
Paint					
s= 10	33.0(0.9227)	33.5(0.9319)	33.5(0.9293)	33.7(0.9329)	33.6(0.9311)
s= 20	29.0(0.8513)	29.6(0.8687)	29.6(0.8655)	29.9(0.8731)	29.5(0.8683)
s= 30	26.9(0.7897)	27.5(0.8110)	27.5(0.8091)	27.7(0.8196)	27.2(0.8088)
s= 40	25.6(0.7408)	26.0(0.7616)	26.0(0.7599)	26.6(0.7711)	25.6(0.7569)
Monarch					
s= 10	33.1(0.9442)	33.6(0.9527)	33.5(0.9501)	33.9(0.9577)	34.2(0.9594)
s= 20	28.8(0.8912)	29.5(0.9076)	29.6(0.9077)	30.1(0.9222)	30.0(0.9202)
s= 30	26.5(0.8370)	27.1(0.8583)	27.4(0.8663)	28.0(0.8850)	27.4(0.8769)
s= 40	25.0(0.7916)	25.7(0.8179)	25.9(0.8260)	26.6(0.8462)	25.9(0.8378)
Barbara					
s= 10	31.6(0.9241)	31.6(0.9246)	32.3(0.9349)	32.7(0.9420)	32.5(0.9378)
s= 20	27.4(0.8314)	27.2(0.8316)	28.4(0.8646)	28.9(0.8819)	28.5(0.8716)
s= 30	25.1(0.7472)	25.0(0.7475)	26.3(0.7919)	26.8(0.8165)	26.2(0.8028)
s= 40	23.5(0.6696)	23.5(0.6718)	24.7(0.7262)	25.0(0.7444)	24.5(0.7378)
Peppers					
s= 10	33.1(0.8853)	33.3(0.8901)	33.4(0.8920)	33.6(0.8939)	33.3(0.8909)
s= 20	29.8(0.8272)	30.1(0.8381)	30.3(0.8400)	30.6(0.8496)	30.1(0.8413)
s= 30	27.8(0.7781)	28.3(0.7968)	28.4(0.7983)	28.8(0.8108)	27.9(0.7973)
s= 40	26.4(0.7339)	26.9(0.7552)	27.1(0.7657)	27.2(0.7729)	26.7(0.7648)
Blood cell					
s= 10	34.6(0.9125)	34.5(0.9136)	35.0(0.9183)	35.0(0.9190)	34.8(0.9137)
s= 20	31.5(0.8706)	31.5(0.8790)	32.3(0.8859)	32.3(0.8874)	32.0(0.8836)
s= 30	29.2(0.8338)	29.4(0.8473)	29.9(0.8525)	30.2(0.8622)	29.6(0.8538)
s= 30	29.2(0.8338)	29.4(0.8473)	29.9(0.8525)	30.2(0.8622)	29.6(0.8538)
s= 40	27.4(0.7899)	27.8(0.8129)	28.4(0.8227)	28.0(0.8264)	28.0(0.8239)
Tower (color)					
s= 10	34.2(0.9017)	34.6(0.9099)	34.7(0.9115)	35.0(0.9144)	34.8(0.9123)
s= 20	30.5(0.8270)	31.1(0.8444)	31.4(0.8533)	31.6(0.8576)	31.1(0.8522)
s= 30	28.5(0.7711)	29.2(0.7919)	29.3(0.8018)	29.7(0.8135)	29.1(0.8069)
s= 40	27.3(0.7277)	27.9(0.7505)	27.9(0.7583)	28.3(0.7760)	27.8(0.7695)
Parrot (color)					
s= 10	34.0(0.9158)	34.1(0.9190)	34.3(0.9215)	34.6(0.9274)	34.6(0.9255)
s= 20	30.3(0.8523)	30.6(0.8665)	30.8(0.8684)	31.2(0.8832)	31.1(0.8776)
s= 30	28.2(0.8048)	28.6(0.8269)	28.8(0.8308)	29.3(0.8505)	29.0(0.8415)
s= 40	26.7(0.7642)	27.2(0.7925)	27.4(0.7994)	27.5(0.8179)	27.5(0.8097)

Let's then focus on the SSIM measure and the visual quality evaluation of these denoising algorithms. From Table 2 we see that BM3D again achieves the highest SSIM measures. Although the proposed LPG-PCA has almost the same PSNR results as K-SVD, it has higher SSIM measures than K-SVD. Again, the two wavelet-based denoising methods have the lowest SSIM measures.

Fig. 4.2 show the cropped and zoom-in denoising results of the ten noisy images (with noise level $s = 20$) by different methods. The sub-figure (a) is the original image; sub-figures (b–f) are the denoised images by the methods in [43] [8,10,14,20] and the proposed LPG-PCA methods, respectively. We see that although BM3D has higher SSIM measures than LPG-PCA, their denoised images are very similar in real visual perception, and they have much better visual quality than all the other methods. The K-SVD method generates many visual disturbing artifacts in the

denoised image. The two wavelet-based denoising methods [8,10] have the worst visual quality. This is because in WT, the same wavelet basis function (with dilation and translation) is used to de-correlate the many different image structures. Often this is not efficient enough to represent the image content so that many denoising errors appear. The proposed LPG-PCA denoising algorithm uses PCA to adaptively compute the local image decomposition transform so that it can better represent the image local structure. In addition, the LPG operation is employed to ensure that only the right samples are involved in PCA training. The denoised images by BM3D and LPG-PCA are very similar in terms of visual perception. Both of them can well preserve the image edges and remove the noise without introducing too many artifacts. Although the PSNR and SSIM measures of LPG-PCA are lower than that of BM3D, LPG-PCA has competitive results in edge preservation. BM3D works better in preserving large-grain edges and denoising smoothing areas (e.g. the image House), where there is a rich amount of non-local redundancies that could be exploited, while LPG-PCA works better in preserving image fine structures (e.g. the eye area of image Lena and the camera boundary in image Cameraman), where BM3D may generate some artifacts because there are not so many non-local redundancies around those structures.

In summary, as a non-local collaborative denoising technique, BM3D can effectively exploit the non-local redundancy in the image to suppress noise. Therefore, it could have very high PSNR and SSIM measures. The large-grain structures and smooth areas could be well reconstructed. However, for fine-grain structures, incorrect non-local information may be introduced by BM3D for image restoration so that some visible artifacts can be generated in those areas. The proposed LPG-PCA method can be viewed as a semi-non-local scheme. It uses a local window to adaptively train the local transform. The vector variable for denoising is defined on a small local block so that LPG-PCA works well in fine-grain edge preservation.

Chapter 5: Conclusion

This paper proposed a spatially adaptive image denoising scheme by using principal component analysis (PCA). To preserve the local image structures when denoising, we modeled a pixel and its nearest neighbors as a vector variable, and the denoising of the pixel was converted into the estimation of the variable from its noisy observations. The PCA technique was used for such estimation and the PCA transformation matrix was adaptively trained from the local window of the image. However, in a local window there can have very different structures from the underlying one; therefore, a training sample selection procedure is necessary. The block matching based local pixel grouping (LPG) was used for such a purpose and it guarantees that only the similar sample blocks to the given one are used in the PCA transform matrix estimation. The PCA transformation coefficients were then shrunk to remove noise. The above LPG-PCA denoising procedure was iterated one more time to improve the denoising performance. Our experimental results demonstrated that LPG-PCA can effectively preserve the image fine structures while smoothing noise. It presents a competitive denoising solution compared with state-of-the-art denoising algorithms, such as BM3D.

In this paper, I have introduced two new filters for removing impulse noise from images and shown how they compare to four other well-known techniques for noise removal. First, four common noise filtering algorithms were discussed. Next, a Spatial Median Filter was proposed based on a combination of work on the Vector Median Filter and the Spatial Median quantile order statistic. Seeing that the order statistic could be utilized in order to make a judgment as to whether a point in the signal is considered noise or not, a Modified Spatial Median Statistic is proposed. The Modified Spatial Median Filter requires two parameters: A window size and a threshold T of the estimated non-noisy pixels under a mask.

In the results, we find the best threshold T to use in the Modified Spatial Median Filter and determined that the best threshold is 4 when using a 3×3 window mask size. Using these as parameters, this filter was included in a comparison of the Mean, Median, Component Median, Vector Median, and Spatial Median Noise Filters. In the broad comparison of noise removal filters, it was concluded that for images containing $p \leq 0.15$ noise composition, the Modified Spatial Median Filter performed the best and that the Component Median Filter performed the best overall noise models tested.

Reference

- [1]. Polikar, R., The story of wavelets. Physics and modern topics in mechanical and electrical engineering, 1999: p. 192-197.
- [2]. Chen, G. and B. Kégl, Image denoising with complex ridgelets. Pattern Recognition, 2007. **40**(2): p. 578-585.
- [3]. Zhang, L., P. Bao, and X. Wu, Hybrid inter-and intra-wavelet scale image restoration. Pattern Recognition, 2003. **36**(8): p. 1737-1746.
- [4]. Hou, Z., Adaptive singular value decomposition in wavelet domain for image denoising. Pattern Recognition, 2003. **36**(8): p. 1747-1763.
- [5]. Chang, S.G., B. Yu, and M. Vetterli, Adaptive wavelet thresholding for image denoising and compression. IEEE transactions on image processing, 2000. **9**(9): p. 1532-1546.
- [6]. Foi, A., V. Katkovnik, and K. Egiazarian, Pointwise shape-adaptive DCT for high-quality denoising and deblocking of grayscale and color images. IEEE Transactions on Image Processing, 2007. **16**(5): p. 1395-1411.
- [7]. Hayat, M.N., et al. Multimedia sensor networks: Recent trends, research challenges and future directions. in Communication, Computing and Digital Systems (C-CODE), International Conference on. 2017. IEEE.
- [8]. Coifman, R.R. and D.L. Donoho, Translation-invariant de-noising, in Wavelets and statistics. 1995, Springer. p. 125-150.
- [9]. Addison, P.S., The illustrated wavelet transform handbook: introductory theory and applications in science, engineering, medicine and finance. 2017: CRC press.
- [10]. Sarawale, M.R. and M.S. Chougule, Image Denoising using Dual-Tree Complex DWT and Double-Density Dual-Tree Complex DWT. International Journal of Advanced Research in Computer Engineering & Technology (IJARCET), 2013. **2**(6).
- [11]. Sharma, M., et al., Dual-Tree Complex Wavelet Transform-Based Features for Automated Alcoholism Identification. International Journal of Fuzzy Systems, 2018. **20**(4): p. 1297-1308.
- [12]. Yaroslavsky, L.P., Digital picture processing: an introduction. Vol. 9. 2012: Springer Science & Business Media.
- [13]. Garg, M.A. and M.S. Goyal, Vector Sparse Representation of Color Image Using Quaternion Matrix Analysis based on Genetic Algorithm. Imperial Journal of Interdisciplinary Research, 2017. **3**(7).

- [14]. Jin, D., et al., Dynamic spatial filtering using a digital micromirror device for high-speed optical diffraction tomography. *Optics express*, 2018. **26**(1): p. 428-437.
- [15]. Xiong, H., Application of a three-lens slit spatial filter in high power lasers. *Optics Communications*, 2018. **418**: p. 37-40.
- [16]. Negi, S.S. and B. Gupta. Survey of various image enhancement techniques in spatial domain using MatLab. in *International Conference on Advances in Computer Engineering and applications-20*. 2014.
- [17]. Jernigen, M. and G. Mclean, Lateral inhibition and image processing. *Nonlinear vision*, edited by RB Pinter and B. Nabet (CRC Press, Boca Raton, FL, 1992), 2018.
- [18]. Shrivastava, M.A., et al., Image Denoising Using Different Filters. *International Journal of Emerging Trends in Science and Technology*, 2015. **2**(04).
- [19]. Gonzalez, R.C., *Digital image processing*. 2016, Prentice hall.
- [20]. Liu, L., et al., A new weighted mean filter with a two-phase detector for removing impulse noise. *Information Sciences*, 2015. **315**: p. 1-16.
- [21]. Bovik, A., *Digital image processing course notes*. Dept. of electrical engineering, U. of Texas at Austin, 1995.
- [22]. Ehsaeyan, E., A Novel NeighShrink Correction Algorithm in Image Denoising. *Iranian Journal of Electrical and Electronic Engineering*, 2017. **13**(3): p. 246-256.
- [23]. Russ, J.C., *The image processing handbook*. 2016: CRC press.
- [24]. Yadav, N., *Two Phase Image Denoising By Principal Component Analysis and Local Pixel Grouping*. 2017.
- [25]. Umbaugh, S.E., *Digital Image Processing and Analysis with MATLAB and CVIptools*. 2017: CRC Press.
- [26]. Lagendijk, R.L., A. Katsaggelos, and J. Biemond. Iterative identification and restoration of images. in *Acoustics, Speech, and Signal Processing, 1988. ICASSP-88., 1988 International Conference on*. 1988. IEEE.
- [27]. Gupta, B. and S.S. Negi, Image denoising with linear and non-linear filters: a review. *International Journal of Computer Science Issues (IJCSI)*, 2013. **10**(6): p. 149.
- [28]. Umbaugh, S.E., *Computer Vision and Image Processing: A Practical Approach Using Cviptools with Cdrom*. 1997: Prentice Hall PTR.
- [29]. Bastiaans, M., Wigner distribution function and its application to first-order optics. *JOSA*, 1979. **69**(12): p. 1710-1716.
- [30]. Tukey, J., Nonlinear (nonsuperposable) methods for smoothing data. *Proc. Cong. Rec. EASCOM'74*, 1974: p. 673-681.
- [31]. Astola, J., P. Haavisto, and Y. Neuvo, Vector median filters. *Proceedings of the IEEE*, 1990. **78**(4): p. 678-689.
- [32]. Reddy, P.B., K.K. Reddy, and P.A. Reddy, Associate-Image Filtering Method with Enhanced Denoising Feature for Road Detection in Disaster Management. *Transactions on Machine Learning and Artificial Intelligence*, 2017. **4**(6): p. 50.
- [33]. Sit, T. and Z. Ying, Survival analysis: a quantile perspective, in *Handbook of Quantile Regression*. 2017, Chapman and Hall/CRC. p. 89-108.
- [34]. Prasad, A. and T. Kumar, A pair of linear canonical Hankel transformations and associated pseudo-differential operators. *Applicable Analysis*, 2017: p. 1-16.
- [35]. Neole, B.A., B.V. Lad, and K. Bhurchandi, Denoising of digital images using consolidation of edges and separable wavelet transform. *International Journal of Computer Applications*, 2017. **168**(10).
- [36]. Chen, J., et al., The design of a novel mother wavelet that is tailor-made for continuous wavelet transform in extracting defect-related features from reflected guided wave signals. *Measurement*, 2017. **110**: p. 176-191.
- [37]. Lan, Y., et al. Time-frequency decomposition of sunspot number fluctuations. in *Control Conference (CCC), 2017 36th Chinese*. 2017. IEEE.

- [38]. Muresan, D.D. and T.W. Parks. Adaptive principal components and image denoising. in Image Processing, 2003. ICIP 2003. Proceedings. 2003 International Conference on. 2003. IEEE.
- [39]. Bow, S.-T., Pattern recognition and image preprocessing. 2002: Marcel Dekker New York.
- [40]. Pizurica, A. and W. Philips, Estimating the probability of the presence of a signal of interest in multiresolution single-and multiband image denoising. IEEE Transactions on image processing, 2006. **15**(3): p. 654-665.
- [41]. Xu, J., L. Yang, and D. Wu, Ripplet: A new transform for image processing. Journal of Visual Communication and Image Representation, 2010. **21**(7): p. 627-639.
- [42]. Donoho, D.L., De-noising by soft-thresholding. IEEE transactions on information theory, 1995. **41**(3): p. 613-627.
- [43]. Portilla, J., et al., Image denoising using scale mixtures of Gaussians in the wavelet domain. IEEE Transactions on Image processing, 2003. **12**(11): p. 1338-1351.
- [44]. Aharon, M., M. Elad, and A. Bruckstein, ℓ_1 -SVD: An algorithm for designing overcomplete dictionaries for sparse representation. IEEE Transactions on signal processing, 2006. **54**(11): p. 4311-4322.
- [45]. Dabov, K., et al. Image restoration by sparse 3D transform-domain collaborative filtering. in Image Processing: Algorithms and Systems VI. 2008. International Society for Optics and Photonics.

