

The Hong Kong Polytechnic University

Department of Computing

Thesis Title

Thesis Author

A thesis
submitted in partial fulfilment of the requirements
for the degree of

Doctor of Philosophy

March 2014

CERTIFICATE OF ORIGINALITY

I hereby declare that this thesis is my own work and that, to the best of my knowledge and belief, it reproduces no material previously published or written, nor material that has been accepted for the award of any other degree or diploma, except where due acknowledgement has been made in the text.

(Signed)

Thesis Author

(Name of student)

Abstract

The nonlocal self-similarity (NSS) prior of natural images has been extensively studied in many image restoration methods. In this thesis, we exploit the NSS property of external natural images, external guided internal NSS property, and internal NSS property for image denoising tasks.

Acknowledgement

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

Contents

1	Introduction	1
1.1	The Image Noise Formulation	1
1.2	Image Denoising	3
1.3	Evaluating Denoising Performance	4
1.4	Literature Review	6
1.5	Synthetic Grayscale Image Denoising	7
1.6	Realistic Color Image Denoising	9
1.7	Contribution	10
1.8	Thesis Structure	11
2	External Non-local Self-Similarity Prior for Additive White Gaussian Noise	15
2.1	Introduction	15
2.2	Patch Group Based Prior Modeling of Nonlocal Self-Similarity	18
2.2.1	Patch Group and Group Mean Subtraction	18
2.2.2	PG-GMM Learning	19
2.2.3	Complexity Analysis	20
2.2.4	Discussions	21
2.3	Image Denoising by Patch Group Priors	22
2.3.1	Denoising Model	22
2.3.2	Denoising Algorithm	25
2.4	Experiments	25
2.4.1	Implementation Details	26
2.4.2	Comparison Methods	27
2.4.3	Results and Discussions	27
2.5	Conclusion	28
3	External Prior Guided Internal Prior Learning for Real Noisy Image Denoising	33
3.1	Learning External Nonlocal Self-Similarity Priors	33
3.2	System Design	34
3.3	Demo System	35

3.4	Calibration	35
3.5	Conclusion	35
4	Internal Nonlocal Self-Similarity Prior for Real Color Image De-noising: A Low Rank based Method	37
4.1	Introduction	37
4.2	Related Work	38
4.3	Method	38
4.4	Experimental Results	38
4.5	Summary	39
5	Internal Nonlocal Self-Similarity Prior for Real Color Image De-noising: A Sparse Coding based method	41
5.1	Introduction	41
5.2	Related Work	42
5.3	Summary	42
6	A Large Real Noisy Image Dataset, with A Comprehensive Evaluation of State-of-the-arts	43
6.1	Introduction	43
6.2	Related Work	44
6.3	Summary	44
7	Conclusions	45
7.1	Section 1	45
7.2	Section 2	45
7.3	Future Work	46

List of Figures

2.1	Flowchart of the proposed patch group based prior learning and image denoising framework.	16
2.2	Different patch groups (PG) share similar PG variations.	19
2.3	The convergence curve of log-likelihood in PG-GMM training on the Kodak PhotoCD Dataset.	21
2.4	Eigenvectors of 3 Gaussian components from the learned PG-GMM, sorted by the values of corresponding eigenvalues.	24
3.1	Figure example: (a) example part one, (c) example part two; (c) example part three	33
3.2	Another Figure example: (a) example part one, (c) example part two; (c) example part three	34

List of Tables

2.1	PSNR(dB) results of different denoising algorithms on 20 natural images.	30
2.2	Average run time (seconds) with standard deviation of different methods on images of size 256×256 and 512×512 . BM3D uses parallelization and is implemented with compiled C++ mex-function while the other methods are implemented in Matlab.	31

Introduction

Nowadays, CCD or CMOS cameras are becoming more and more important in many aspects of human life such as photography, artificial intelligence, and security system. For each camera product, the camera imaging pipeline in the camera is of particular importance since it is the core part to transform the photons reflected by the real scene being captured in the camera sensor into the pixel values of an image, which can be displayed on a screen. During the camera imaging process, the noise is unavoidably generated due to many reasons. Two major reasons of noise generation are the discrete nature of light and the thermal agitation, which can cause the photon shot noise and the dark-current noise, respectively. Image denoising is the problem of recovering the latent clean image from the captured noise version.

Chapter abstract This chapter will introduce the image noise and its acquirezation equation, the image denoising problem, the objective measures to evaluate the image denoising performance, the proposed denoising methods. Finally, I will summarize the structural of this thesis, and the contribution I made in this thesis.

1.1 The Image Noise Formulation

The realistic noise are very complex in real photography images captured by the camera sensors. One major reason is that the noise is unable to be explicitly modeled by some definitive probabilistic distributions. Often the sRGB images we look at the screen have been processed via the camera imaging pipeline, during which the noise will be much more complex than its initial status.

The noise are generated mainly due to the discrete nature of light and the thermal agitation, which will cause unstable measurement in camera sensors. The major types of noise generated during the imaging pipeline are the random noise, the spatial non-uniformity noise, and quantization noise. The random noise includes photon shot noise, dark current, and readout noise. The spatial non-uniformity noise includes the fixed pattern noise (PRNU, DCNU), CCD/CMOS specific noise.

To better describe the noise quantitatively, we provide a simplified signal acquisition model [] including various noise sources (for each pixel) as follows:

$$P = f((g_{cv}(\mathbf{C} + \mathbf{D}) + \mathbf{N}_{reset})g_{out} + \mathbf{N}_{out}) + \mathbf{Q}. \quad (1.1)$$

The above equation is explained in details:

- P is the raw pixel value;
- f is the camera response function, usually a linear function before attaining a saturation threshold;
- \mathbf{C} is the number of absorbed electrons (charges) transformed from the photons via the photon-diodes in the camera sensor, which can be modeled by a Poisson distribution;
- \mathbf{D} is the number of absorbed electrons generated by dark current by thermal generation, which is often modeled by a Poisson distribution;
- \mathbf{N}_{reset} is the thermal noise generated by the readout circuitry (or reset noise related to reset voltage), which can be well modeled by a Gaussian distribution;
- \mathbf{N}_{out} is the readout noise, which is also modeled by a Gaussian distribution;
- \mathbf{Q} is the quantization error happened during rounding to integer values, usually uniformly distributed and normally negligible compared to the readout noise;
- g_{cv} is the equivalent capacitance (EC) of the photo-diode and the gain factor during charge to voltage conversion;
- g_{out} is the gain factor during voltage to pixel value conversion (readout).

After some merging and simplifying, the acquisition model of the signal can be formulated as follows:

$$\begin{aligned} \mathbf{P} &= f((g_{cv}(\mathbf{C} + \mathbf{D}) + \mathbf{N}_{reset})g_{out} + \mathbf{N}_{out}) + \mathbf{Q}, \\ &= f(g_{cv}g_{out}(\mathbf{C} + \mathbf{D}) + g_{out}\mathbf{N}_{reset} + \mathbf{N}_{out}) + \mathbf{Q}, \\ &= f(g\lambda + N_R) + \mathbf{Q}, \end{aligned} \quad (1.2)$$

where $g = g_{cv}g_{out}$ is the overall camera gain factor, $\lambda = \mathbf{C} + \mathbf{D}$ is # of electrons in pixel capacitor and also Poisson distributed, since “Sum of Independent Poisson Random Variables is also Poisson”, and $N_R = g_{out}\mathbf{N}_{reset} + \mathbf{N}_{out}$ is the overall readout noise and also Gaussian distributed, since “Sum of Independent Gaussian Random Variables is also Gaussian”. In summary, the overall noise before the camera imaging pipeline can be modeled by a mixed Poisson and Gaussian distribution []. Unfortunately, the realistic noise will become very complex after being processed during the camera imaging pipeline [**crosschannel**]. This makes the image denoising an important and challenging task.

To make the problem easier, in image denoising community, the most common tested noise is additive white Gaussian noise (AWGN) [**ksvd**]. In AWGN degraded image, each pixel is corrupted by a random value following Gaussian distribution with zero mean and a certain standard deviation (std). Note that for each pixel, the std of the Gaussian distribution is the same, and for all the pixels, the noise values are sampled independently. The AWGN noise is similar to the read-out noise in digital cameras.

In order to obtain images of good quality while still maintaining the structures and details of the captured scenes, image denoising is an essential step. In this thesis, I will present my work on image denoising, in which we focus on synthetic additive white Gaussian noise (AWGN) and the realistic noise in real-world images. The AWGN noise is described as a Gaussian distribution $\mathcal{N}(0, \sigma^2)$, which means that the noise is Gaussian distributed with 0 mean and σ standard deviation. Most of existing methods focus on processing AWGN noise since it is a perfect testing bed for evaluating the proposed methods as well as other image restoration problems such as image super-resolution, deblurring, inpainting, etc.

1.2 Image Denoising

In general, image denoising aims to recover the latent clean image \mathbf{x} from the observed noisy image $\mathbf{y} = \mathbf{x} + \mathbf{n}$, where \mathbf{n} is assumed to be the additive noise. \mathbf{n} is often assumed to be synthetic additive white Gaussian noise, or the realistic noise in real-world images. Image denoising can be viewed as a regression problem, in which a *plausible* clean image can be obtained from the infinite number of possible candidates. The word *plausible* means that

the denoised image should look like the noisy image but without the noise component.

Image denoising would be very hard if we do not employ any prior information on it. The reason is that we do not know what exactly the latent clean image is without the prior information of the clean image. Hence, it is meaningful to exploit the most *plausible* image under some prior information. The most commonly used prior information in image denoising community is the Bayesian rule, which is also known as maximum A-posterior (MAP) property. Under the MAP framework, the most *plausible* latent clean image is the one which has the maximum Bayesian probability given the given noisy version. The posterior probability can be measured by some existing evaluation methods which I will introduce in the following sections. In fact, the probability or measurements can measure the closeness of the latent clean image to the given noisy image. The closeness is usually measured by the ℓ_2 norm of the difference between the two images mentioned above. There are many latent clean images with the same ℓ_2 norm distance with the given noisy image. But some images in the circle are more *plausible* than the others due to the aspects of less artifacts, better structural preservation, and less remaining noise, etc.

1.3 Evaluating Denoising Performance

In order to achieve the maximum Bayesian probability, we need calculate the measurements of goodness for the denoised images. A natural problem is, how to measure the quality of the denoised image? It is very important to find better answer to this question. In fact, the research of image quality assessment is to find good algorithms to measure the quality of images under different situations and applications including image denoising.

A initial understanding is that the answer to the above question is largely depends on the situations we face when we perform image denoising experiments. When we perform synthetic experiments on additive white Gaussian noise (AWGN), we usually already have the original clean image, the noisy image is generated by adding synthetic AWGN noise to the clean image. Then we can directly measure the quality of the denoised image by some existing image quality assessment (IQA) metrics. When we do not have the clean image provided as “ground truth”, a possible and final solution is to measure

the image quality by relying on human subjective evaluation. The IQA metrics can be roughly divided into two major directions: 1) full reference IQA; and 2) no reference IQA. Full reference IQA metrics are based on the assumption that the true underlying image is available in order to compute a measure, while no reference IQA metrics perform quality assessments without the reference image since the true underlying image is not available.

RMSE and PSNR: We mentioned previously that it is corresponding to the ℓ_2 norm or equally mean square error (MSE) which is used to measure the distance between the denoised image and the given noisy image. In fact, the MSE measure is closely related to the famous peak signal to noise ratio (PSNR) metric. PSNR is the most commonly used full reference IQA metric for many image restoration tasks including denoising. The definition of PSNR can be formulated as follows (for 8-bit image):

$$\text{PSNR} = 20\log_{10}\left(\frac{2^8}{\text{RMSE}(\mathbf{x}, \mathbf{y})}\right), \quad (1.3)$$

where $\text{RMSE}(\mathbf{x}, \mathbf{y})$ refers to the root mean square error defined as $\text{RMSE} = \sqrt{\frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N (\mathbf{x}_{ij} - \mathbf{y}_{ij})^2}$ for images $\mathbf{x}, \mathbf{y} \in \mathcal{R}^{M \times N}$. As we can see, PSNR is closely related to the ℓ_2 norm distance between two images. The unit of PSNR is decibel (dB) and higher dB value indicates better image quality and lower RMSE. Even though PSNR is very simple and intuitive, higher PSNR does not indicate higher visual structural similarity. Hence, many researchers still make effort to find alternative and better IQA metrics.

SSIM [ssim]: Some researchers attempted to exploit the visual properties of the human visual system. One of the seminar work in this direction is the famous structural similarity index metric (SSIM), which is a full reference IQA metric. In SSIM, each image patch is separated into three different components indicating three core informative parts of the original patch. The three components are luminance (mean value of the pixels in the patch), contrast (the standard deviation of the patch), and structure (the mean subtracted patch). The major advantage of the SSIM is that it takes into account the fact that the human visual system is very sensitive to the relative changes in luminance, rather than the absolute changes in luminance. The value range of the SSIM is between 0 and 1, where higher value indicate higher similarity (SSIM of 1 indicates that the two images are exactly the same). Note that it does not indicate that higher SSIM indicate better image quality, since the reference image is not the image of the best quality.

Other IQA Metrics: Besides of the PSNR and SSIM, there are many other IQA metrics for full reference and no reference IQA. Some examples include the MS-SSIM [msssim], which is a multi-scale extension of the original SSIM. Some examples in no reference IQA include BLIINDS [bliinds] and BIQI [biqi]. These IQA metrics capture the deviations from the expected statistics of the natural images. For example, BLIINDS measures the deviations from the expected histogram of certain features in DCT domain, while BIQI measures deviations from the expected distribution of wavelet coefficients in a multi-scale decomposition.

I have to mention that no IQA metric is perfect or best for image denoising task, both in full reference and no reference cases. The de facto standard metrics in image restoration community are PSNR and SSIM. In order to avoid these two metrics generate bad results, it is essential to demonstrate the image quality in the thesis for human subjective evaluation.

1.4 Literature Review

In this chapter, I will review the methods related to denoising in literature during the past decades. These methods can be divided into three categories. Firstly, I will review the denoising methods designed for additive white Gaussian noise (AWGN) since AWGN is the mostly studied noise distribution in the literature. Though these methods are proposed to deal with the AWGN noise, the idea can be applied to the other image denoising tasks such as real color image denoising. Secondly, I will review the existing methods proposed for processing real noisy images. Though the methods in this domain is not that versatile than those methods for the AWGN noise, the real noisy image denoising is the current mainstream for the image denoising community. Due to the noise is not known beforehand, noise estimation should be performed for the real noisy image denoising task. Finally, I will also review the image noise estimation methods in the literature.

However, noise in real images is more complex than simple Gaussian or mixed Gaussian and impulse distribution. Besides, noise is usually unknown for existing methods. This makes image denoising still a challenging problem.

1.5 Synthetic Grayscale Image Denoising

As a classical problem in low level vision, image denoising has been extensively studied in the past decades, yet it is still an active topic for the reason that it provides an ideal test bed for image modeling techniques. In general, image denoising aims to recover the clean image \mathbf{x} from its noisy observation $\mathbf{y} = \mathbf{x} + \mathbf{n}$, where \mathbf{n} is the additive noise which is often assumed to be additive white Gaussian noise (AWGN). Other types of noise, e.g., Poisson noise and salt-and-pepper noise, are also been studied in the literature. The Poisson noise can be transformed into the additive noise after some transformation. Besides, the salt-and-pepper noise is naturally additive noise and can be formulated into the model mentioned above.

A variety of image denoising methods have been developed in past decades, including diffusion based methods [**PeronaMalik1990**], total variation based methods [**rudin1992nonlinear**; **osher2005iterative**], filtering based methods [**Tomasi1998**], wavelet/curvelet based methods [**softthresholding**; **bayesshrink**; **curvelet**], nonlocal self-similarity based methods [**nlm**; **bm3d**], sparse representation based methods [**ksvd**; **lssc**; **ncsr**], and low rank based methods [**nmm**; **wnnm**], etc. Recently, some discriminative denoising methods have also been developed by learning discriminative priors from pairs of clean and noisy images [**mlp**; **csf**; **tndr**].

Image modeling plays a central role in image denoising. By modeling the wavelet transform coefficients as Laplacian distributions, many wavelet shrinkage based denoising methods such as the classical soft-thresholding [**softthresholding**] have been proposed. Chang et al. modeled the wavelet transform coefficients as generalized Gaussian distribution, and proposed the BayesShrink [**bayesshrink**] algorithm. By considering the correlation of wavelet coefficients across scales, Portilla et al. [**blsgsm**] proposed to use Gaussian Scale Mixtures for image modeling and achieved promising denoising performance. It is widely accepted that natural image gradients exhibit heavy-tailed distributions [**weiss**], and the total variation (TV) based methods [**rudin1992nonlinear**; **osher2005iterative**] actually assume Laplacian distributions of image gradients for denoising. The Fields of Experts (FoE) [**foe**] proposed by Roth and Black models the filtering responses with Student's t-distribution to learn filters through Markov Random Field (MRF) [**Bishop**]. Recently, Schmidt and Roth proposed the cascade of shrinkage fields (CSF) to perform denoising efficiently [**csf**].

Instead of modeling the image statistics in some transformed domain (e.g., gradient domain, wavelet domain or filtering response domain), another popular approach is to model the image priors on patches. One representative is the sparse representation based scheme which encodes an image patch as a linear combination of a few atoms selected from a dictionary [olshausen1996emergence; olshausen1997sparse; ksvd]. The dictionary can be chosen from the off-the-shelf dictionaries (e.g., wavelets and curvelets), or it can be learned from natural image patches. The seminal work of K-SVD [ksvdtsp; ksvd] has demonstrated promising denoising performance by dictionary learning, which has yet been extended and successfully used in various image processing and computer vision applications [srcolor; srcvpr; lcksvd]. By viewing image patches as samples of a multivariate variable vector and considering that natural images are non-Gaussian, Zoran and Weiss [epll; gmmnips] and Yu et al. [ple] used Gaussian Mixture Model (GMM) to model image patches, and achieved state-of-the-art denoising and image restoration results, respectively.

When the input is a noisy RGB color image, there are mainly three strategies for color image denoising. (1) The first strategy is to apply the grayscale image denoising algorithm to each channel. However, such a straightforward solution will not exploit the spectral correlation among RGB channels, and the denoising performance may not be very satisfying. (2) The second strategy is to transform the RGB image into a less correlated color space, such as YCbCr, and perform denoising in each channel of the transformed space [foe; cbm3d]. One representative work along this line is the CBM3D algorithm [cbm3d]. However, the color transform will complicate the noise distribution, and the correlation among color channels is not fully exploited. (3) The third strategy is to perform joint denoising on the RGB channels simultaneously for better use of the spectral correlation. For example, the patches from RGB channels are concatenated as a long vector for processing [mairal2008sparse; Zhu_2016_CVPR].

Though joint denoising of RGB channels is a more promising way for color image denoising, it is not a trivial extension from single channel (grayscale image) to multiple channels (color image). The noise in standard RGB (sRGB) space can be approximately modeled as AWGN, but it has different variances for different channels [Liu2008; Leungtip; crosschannel2016] due to the sensor characteristics and on-board processing steps in digital camera pipelines [crosschannel2016; karaime_brown_ECCV_2016]. This makes

the real color image denoising problem much more complex. If the three channels are treated equally in the joint denoising process, false colors or artifacts can be generated [**mairal2008sparse**]. How to account for the different noise characteristics in color channels, and how to effectively exploit the within and cross channel correlation are the key issues for designing a good color image denoising method.

1.6 Realistic Color Image Denoising

During the last decade, a few methods have been proposed for real color image denoising. To the best of our knowledge, the study of real color image denoising can be traced back to the BLS-GSM model [**blsgsm**]. In [**blsgsm**], Portilla et al. proposed to use scale mixture of Gaussian in overcomplete oriented pyramids to estimate the latent clean images. In [**fullyblind**], Portilla proposed to use a correlated Gaussian model for noise estimation of each wavelet subband. Based on the robust statistics theory [**huber2011robust**], Rabie modeled the noisy pixels as outliers, which could be removed via Lorentzian robust estimator [**rabie2005robust**]. The CBM3D method [**cbm3d**] is a representative color image denoising method, which first transforms the RGB image into a luminance-chrominance space (e.g., YCbCr) and then applies the benchmark BM3D method [**bm3d**] to each channel separately. The non-local similar patches are grouped by the luminance channel. In [**Liu2008**], Liu et al. proposed the “Noise Level Function” to estimate the noise for each channel in natural images, and then use Gaussian conditional random field to obtain the latent clean image [**Liu2008**]. However, processing each channel separately would often achieve inferior performance to processing the color channels jointly [**mairal2008sparse**]. Later, Lebrun el al. proposed a multiscale denoising algorithm called ‘Noise Clinic’ [**noiseclinic**] for blind image denoising task. This method generalizes the NL-Bayes [**nlbayes**] to deal with signal and frequency dependent noise. Therefore, the methods [**noiseclinic**; **ncwebsite**; **Zhu_2016_CVPR**] perform real color image denoising by concatenating the patches of RGB channels into a long vector. However, the concatenation treats each channel equally and ignores the different noise statistics among these channels. The method in [**crosschannel2016**] models the cross-channel noise in real noisy images as multivariate Gaussian and the noise is removed by the Bayesian non-local means filter [**kervrann2007bayesian**]. The commercial software Neat Image [**neatimage**] estimates the noise parameters from a flat region

of the given noisy image and filters the noise accordingly. The methods in [crosschannel2016; neatimage] ignore the non-local self-similarity of natural images [bm3d; wnnm].

Despite the success of these methods, they have many limitations. On one hand, as suggested in [Liu2008; noiseclinic], Gaussian noise, assumed by [fullyblind; rabie2005robust; Liu2008], may be inflexible for more complex noise in real images. Hence, better approximation to the noise could bring better image denoising performance [Liu2008; noiseclinic]. Based on these observations, it is still needed to design an robust and effective model for blind image denoising. Few assumption and no parameter tuning would bring extra points.

1.7 Contribution

This thesis is mainly consisted by the several work I have done during my PhD study, during which I focus on designing new and better image denoising algorithms.

Firstly, I propose a method for denoising synthetic AWGN noise, from which we can study the performance of the nonlocal self-similarity priors of natural images. In fact, we propose to learn the external NSS priors and apply the learned model on denoising AWGN noise. The proposed method achieves state-of-the-art performance on AWGN denoising on both effectiveness and efficiency.

Basing on the success on the synthetic noise removal, I propose to exploit the power of the NSS priors in natural images to deal with the complex realistic noise in real-world noisy images. Specifically, we propose three methods exploiting the NSS priors of natural images for real noisy image denoising, which can be introduced as follows.

In the first method, I propose to learn the NSS prior from the external natural images, and then apply the learned external prior to guide the learning of the internal NSS prior of the input real noisy image. The experiments on two commonly used datasets and a new one we constructed to implement the shortage of existing datasets, demonstrate that the proposed method can achieve better performance than existing color image denoising methods such as CBM3D [cbm3d], the state-of-the-art Gaussian noise removal methods

[bm3d; mlp; csr], and the real noisy image denoising methods [] including a commercial software Neat Image [**neatimage**], which is embeded into the famous PhotoShop CS for image processing tasks.

In the second method, I propose to employ the low rank model describe fully the the internal NSS prior, basing on the observed fact that the similar image patches can be contanated as a matrix of low rank. Different from the previous work, I extend the WNNM model and apply it to multi-channel version to make it feasible for color image denoising.

In the third method, is to use the sparse coding based method with additional weighting scheme to regard the local noise in real noisy images as a Gaussian and the prior is used to deal with the real noisy image.

Finally, to make my thesis more comprehensive, I construct a large benchmark of real noisy images captured by different types of famous commercial cameras, on which I also evaluate the image denoising methods mentioned above and the proposed methods in this thesis.

The structure of this thesis is organized as follows: in the chapter 2, we review the literatures in the image denoising area; in the chapter 3, we introduce the fully external method; in the chapter 4, we introduce the external prior guided internal method; in the chapter 5, we introduce the internal method based on low ran model; in the chapter 6, we introduce the internal method based on sparse coding model; in the chapter 7, we introduce the real noisy image dataset we construct, and finaly evaluate the proposed methods with the compared competing methods, both for synthetic AWGN or Poisson noise and real noise, including the commercial software designed especially for real noise.

1.8 Thesis Structure

Chapter 2: External Nonlocal Self-Similarity Prior Learning for Synthetic Gaussian Noise Removal

In this chapter, I will introduce our work on external nonlocal self-similarity (NSS) prior learning for synthetic Gaussian noise removal. As far as we know, this work is the first to learn the NSS priors of natural clean images, while

previous work only utilize the NSS priors of input noisy image for online denoising. The advantages of this offline learning is that it can preserve the details of natural images while being much faster then most online denoising methods.

Chapter 3: External Prior Guided Internal Prior Learning for Real Noisy Image Denoising

In this chapter, I will introduce our work on external prior guided internal prior learning method for real noisy image denoising. This work can maintain the advantages of both sides: from the external perspective, the method can preserve the structures of natural images better than the internal methods, while from the perspective of internal method, the proposed method can recover the details of the input noisy image better than the external methods.

Chapter 4: Multi-channel Weighted Nuclear Norm Minimization for Real Color Image Denoising

In this chapter, we introduce a multi-channel weighted nuclear norm minimization (MC-WNNM) method. This method regards different channels in RGB images differently to adaptively process the real color noisy images. Besides, this work also propose a new strategy for color image denoising. Experiments demonstrate that the proposed method can achieve better performance on real color image denoising than existing state-of-the-art methods, including some commercial software.

Chapter 5: A Triple Weighted Sparse Coding Scheme for Realistic Noisy Image Denoising

In this chapter, I introduce a novel sparse coding based method for real color image denoising. In this method, I regard the noise in each of the local region in the real noisy image as a Gaussian, and propose a triplely weighted scheme to deal with the complex realistic noise in real color noisy images. Experiments show that the proposed method performs better and faster than the nuclear norm based method mentioned in previous chapter.

Chapter 6: A Benchmark on Real Color Noisy Image, with Comprehensive Evaluation of State-of-the-art

To fully boost the research of real color noisy image denoising, we construct a large benchmark on real color noisy images. This dataset is collected from several representative cameras with comprehensive settings on contents, lighting, ISO, shutter, and aperture, etc. Based on this newly established dataset, we fully evaluated existing denoising methods, including the methods designed for synthetic Gaussian noise and the methods designed especially for real color noise. We believe that this new dataset will largely boost the research of the image denoising especially the realistic image denoising problems.

External Non-local Self-Similarity² Prior for Additive White Gaussian Noise

2.1 Introduction

As a classical problem in low level vision, image denoising has been extensively studied, yet it is still an active topic for that it provides an ideal test bed for image modeling techniques. In general, image denoising aims to recover the clean image x from its noisy observation $y = x + v$, where v is assumed to be additive white Gaussian noise. A variety of image denoising methods have been developed in past decades, including filtering based methods [Tomasi1998], diffusion based methods [PeronaMalik1990], total variation based methods [rudin1992nonlinear; osher2005iterative], wavelet/curvelet based methods [softthresholding; bayesshrink; curvelet], sparse representation based methods [ksvd; lssc; ncsr], nonlocal self-similarity based methods [nlm; bm3d; nnm; wnnm], etc.

Image modeling plays a central role in image denoising. By modeling the wavelet transform coefficients as Laplacian distributions, many wavelet shrinkage based denoising methods such as the classical soft-thresholding [softthresholding] have been proposed. Chang et al. modeled the wavelet transform coefficients as generalized Gaussian distribution, and proposed the BayesShrink [bayesshrink] algorithm. By considering the correlation of wavelet coefficients across scales, Portilla et al. [blsgsm] proposed to use Gaussian Scale Mixtures for image modeling and achieved promising denoising performance. It is widely accepted that natural image gradients exhibit heavy-tailed distributions [weiss], and the total variation (TV) based methods [rudin1992nonlinear; osher2005iterative] actually assume Laplacian distributions of image gradients for denoising. The Fields of Experts (FoE) [foe] proposed by Roth and Black models the filtering responses with Student's t-distribution to learn filters through Markov Random Field (MRF) [Bishop]. Recently, Schmidt and Roth proposed the cascade of shrinkage fields (CSF) to perform denoising efficiently [csf].

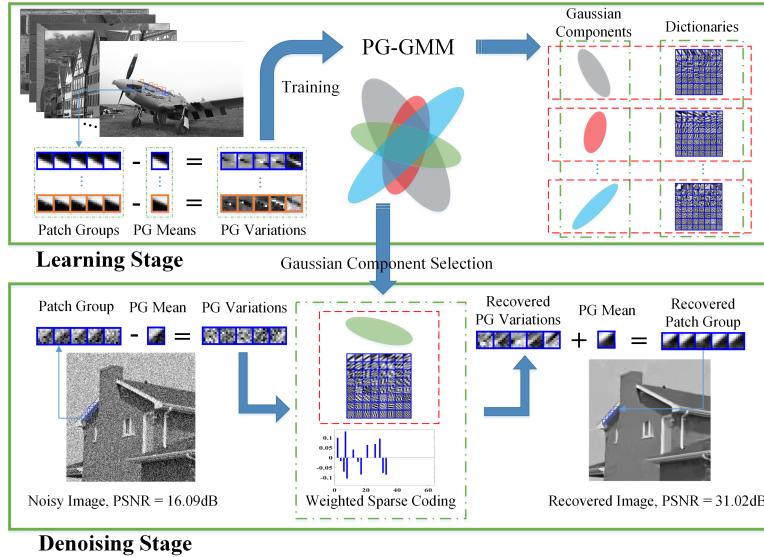


Fig. 2.1: Flowchart of the proposed patch group based prior learning and image denoising framework.

Instead of modeling the image statistics in some transformed domain (e.g., gradient domain, wavelet domain or filtering response domain), another popular approach is to model the image priors on patches. One representative is the sparse representation based scheme which encodes an image patch as a linear combination of a few atoms selected from a dictionary [olshausen1996emergence; olshausen1997sparse; ksvd]. The dictionary can be chosen from the off-the-shelf dictionaries (e.g., wavelets and curvelets), or it can be learned from natural image patches. The seminal work of K-SVD [ksvdtsp; ksvd] has demonstrated promising denoising performance by dictionary learning, which has yet been extended and successfully used in various image processing and computer vision applications [srcolor; srcvpr; lcksvd]. By viewing image patches as samples of a multivariate variable vector and considering that natural images are non-Gaussian, Zoran and Weiss [epll; gmmnips] and Yu et al. [ple] used Gaussian Mixture Model (GMM) to model image patches, and achieved state-of-the-art denoising and image restoration results, respectively.

Natural images often have many repetitive local patterns, and a local patch can have many similar patches to it across the whole image. The so-called nonlocal self-similarity (NSS) prior is among the most successful priors for image restoration. The nonlocal means [nlm] and nonlocal regularization [nlr] methods improve much the image denoising performance over the conventional local self-similarity based methods. Dabov et al. [bm3d]

constructed 3D cubes of nonlocal similar patches and conducted collaborative filtering in the sparse 3D transform domain. The so-called BM3D algorithm has become a benchmark in image denoising. Mairal et al. [**lssc**] proposed the LSSC algorithm to exploit NSS via group sparse coding. The NSP [**nsp**] method fits the singular values of NSS patch matrix by Laplacian distribution. Dong et al. [**ncsr**] unified NSS and local sparse coding into the so-called NCSR framework, which shows powerful image restoration capability. By assuming that the matrix of nonlocal similar patches has a low rank structure, the low-rank minimization based methods [**nmm**; **wnnm**] have also achieved very competitive denoising results.

Though NSS has demonstrated its great success in image denoising, in most existing methods only the NSS of noisy input image is used for denoising. For example, in BM3D [**bm3d**] the nonlocal similar patches of a noisy image are collected as a cube for collaborative filtering. In NCSR [**ncsr**], the nonlocal means are subtracted in the sparse domain to regularize the sparse coding of noisy patches. In WNNM [**wnnm**], the low-rank regularization is enforced to recover the latent structure of the matrix of noisy patches. We argue that, however, such utilizations of NSS are not effective enough because they neglect the NSS of clean natural images, which can be pre-learned for use in the denoising stage. To the best of our knowledge, unfortunately, so far there is not an explicit NSS prior model learned from natural images for image restoration.

With the above considerations, in this work we propose to learn explicit NSS models from natural images, and apply the learned prior models to noisy images for high performance denoising. The flowchart of the proposed method is illustrated in Fig. 2.1. In the learning stage, we extract millions of patch groups (PG) from a set of clean natural images. A PG is formed by grouping the similar patches to a local patch in a large enough neighborhood. A PG based GMM (PG-GMM) learning algorithm is developed to learn the NSS prior for the PGs. In the denoising stage, the learned PG-GMM will provide dictionaries as well as regularization parameters, and a simple weighted sparse coding model is developed for image denoising. Our extensive experiments validated that the proposed PG prior based denoising method outperforms many state-of-the-art algorithms quantitatively (in PSNR) while being much more efficient. More importantly, it delivers the best qualitative denoising results with finer details and less artifacts, owe to the NSS prior learned from clean natural images.

2.2 Patch Group Based Prior Modeling of Nonlocal Self-Similarity

Image nonlocal self-similarity (NSS) has been widely adopted in patch based image denoising and other image restoration tasks [**nlm**; **bm3d**; **lssc**; **ncsr**; **wnnm**]. Despite the great success of NSS in image restoration, most of the existing works exploit the NSS only from the degraded image. Usually, for a given patch in the degraded image, its nonlocal similar patches are collected, and then the nonlocal means [**nlm**], or 3D transforms [**bm3d**], or some regularization terms [**lssc**; **ncsr**; **wnnm**; **srmcolor**] can be introduced for image restoration. However, how to learn the NSS prior from clean natural images and apply it to image restoration is still an open problem. In this work, we make the first attempt on this problem, and develop a patch group (PG) based NSS prior learning scheme.

2.2.1 Patch Group and Group Mean Subtraction

For each local patch (size: $p \times p$) of a given clean image, we can find the first M most similar nonlocal patches to it across the whole image. In practice, this can be done by Euclidean distance based block matching in a large enough local window of size $W \times W$. A PG is formed by grouping the M similar patches, denoted by $\{\mathbf{x}_m\}_{m=1}^M$, where $\mathbf{x}_m \in \mathbb{R}^{p^2 \times 1}$ is a patch vector. The mean vector of this PG is $\boldsymbol{\mu} = \frac{1}{M} \sum_{m=1}^M \mathbf{x}_m$, and $\bar{\mathbf{x}}_m = \mathbf{x}_m - \boldsymbol{\mu}$ is the group mean subtracted patch vector. We call

$$\bar{\mathbf{X}} \triangleq \{\bar{\mathbf{x}}_m\}, m = 1, \dots, M \quad (2.1)$$

the group mean subtracted PG, and it will be used to learn the NSS prior in our work.

In Fig. 2.2, we show two different PGs, their group means, and the PGs after mean subtraction. One can see that before mean subtraction, the two PGs have very different local structures. After mean subtraction, the two PGs will have very similar variations. This greatly facilitates the prior learning because the possible number of patterns is reduced, while the training samples of each pattern are increased. We will discuss further the benefits of mean subtraction and the associated prior model learning in Section 2.4.

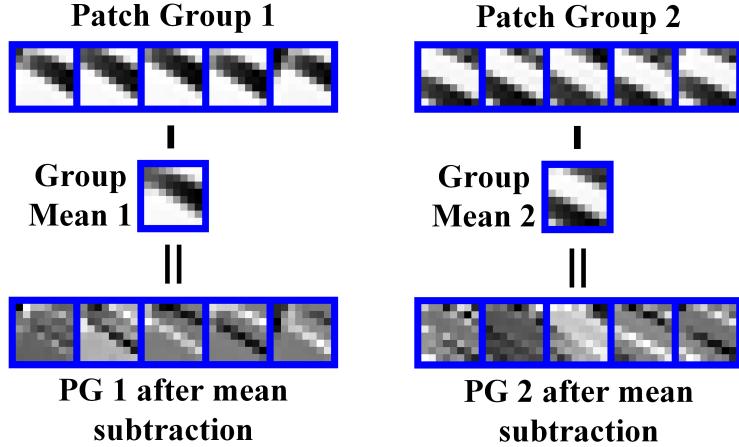


Fig. 2.2: Different patch groups (PG) share similar PG variations.

2.2.2 PG-GMM Learning

From a given set of natural images, we can extract N PGs, and we denote one PG as

$$\bar{\mathbf{X}}_n \triangleq \{\bar{\mathbf{x}}_{n,m}\}_{m=1}^M, n = 1, \dots, N. \quad (2.2)$$

The PGs $\{\bar{\mathbf{X}}_n\}$ contain a rich amount of NSS information of natural images, and the problem turns to how to learn explicit prior models from $\{\bar{\mathbf{X}}_n\}$. Considering that Gaussian Mixture Model (GMM) has been successfully used to model the image patch priors in EPLL [epll] and PLE [ple], we propose to extend patch based GMM to patch group based GMM (PG-GMM) for NSS prior learning.

With PG-GMM, we aim to learn a set of K Gaussians $\{\mathcal{N}(\boldsymbol{\mu}_k, \Sigma_k)\}$ from N training PGs $\{\bar{\mathbf{X}}_n\}$, while requiring that all the M patches $\{\bar{\mathbf{x}}_{n,m}\}$ in PG $\bar{\mathbf{X}}_n$ belong to the same Gaussian component and assume that the patches in the PG are independently sampled. Note that such an assumption is commonly used in patch based image modeling [ksvd; lssc]. Then, the likelihood of $\{\bar{\mathbf{X}}_n\}$ can be calculated as

$$P(\bar{\mathbf{X}}_n) = \sum_{k=1}^K \pi_k \prod_{m=1}^M \mathcal{N}(\bar{\mathbf{x}}_{n,m} | \boldsymbol{\mu}_k, \Sigma_k). \quad (2.3)$$

By assuming that all the PGs are independently sampled, the overall objective likelihood function is $\mathcal{L} = \prod_{n=1}^N P(\overline{\mathbf{X}}_n)$. Taking the log of it, we maximize the following objective function for PG-GMM learning

$$\ln \mathcal{L} = \sum_{n=1}^N \ln \left(\sum_{k=1}^K \pi_k \prod_{m=1}^M \mathcal{N}(\overline{\mathbf{x}}_{n,m} | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right). \quad (2.4)$$

As in GMM learning [Bishop], we introduce hidden variables $\{\Delta_{nk} | n = 1, \dots, N; k = 1, \dots, K\}$ to optimize (2.4). If PG $\overline{\mathbf{X}}_n$ belongs to the k th component, $\Delta_{nk} = 1$; and $\Delta_{nk} = 0$ otherwise. Then the EM algorithm [em] can be used to optimize (2.4) via two alternative steps. In the E-Step, by the Bayes' formula, the expected value of Δ_{nk} is

$$\gamma_{nk} = \frac{\pi_k \prod_{m=1}^M \mathcal{N}(\overline{\mathbf{x}}_{n,m} | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{l=1}^K \pi_l \prod_{m=1}^M \mathcal{N}(\overline{\mathbf{x}}_{n,m} | \boldsymbol{\mu}_l, \boldsymbol{\Sigma}_l)}. \quad (2.5)$$

In the M-step, since for each PG $\overline{\mathbf{X}}_n$, $\sum_{m=1}^M \overline{\mathbf{x}}_{n,m} = \mathbf{0}$, we have

$$\boldsymbol{\mu}_k = \frac{\sum_{n=1}^N \gamma_{nk} \sum_{m=1}^M \overline{\mathbf{x}}_{n,m}}{\sum_{n=1}^N \gamma_{nk}} = \mathbf{0}, \quad (2.6)$$

$$\boldsymbol{\Sigma}_k = \frac{\sum_{n=1}^N \gamma_{nk} \sum_{m=1}^M \overline{\mathbf{x}}_{n,m} \overline{\mathbf{x}}_{n,m}^T}{\sum_{n=1}^N \gamma_{nk}}. \quad (2.7)$$

The calculations of $\pi_k = \frac{1}{N} \sum_{n=1}^N \gamma_{nk}$ are similar to [Bishop].

By alternating between the E-step and the M-step, the model parameters will be updated iteratively, and the update in each iteration can guarantee to increase the value of the log-likelihood function (2.5), and the EM algorithm will converge [Bishop; emconvergence]. Fig. 2.3 shows the convergence curve of the proposed PG-GMM algorithm by using the Kodak PhotoCD Dataset (<http://r0k.us/graphics/kodak/>) for training.

2.2.3 Complexity Analysis

In the training stage, there are N PGs, each of which has M patches, and hence we have $N \times M$ patches. In the M-step, we only need to calculate the covariance matrices since the mean of each Gaussian component is zero. The cost of this step is $O(p^4 MN)$. In the E-step, the cost is $O(p^6 MN)$. Suppose

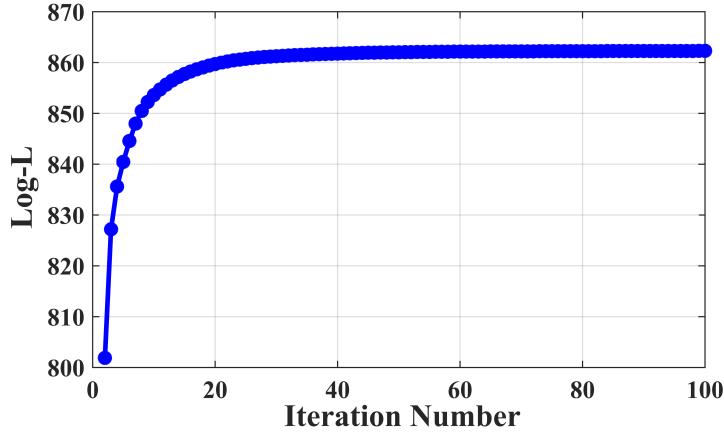


Fig. 2.3: The convergence curve of log-likelihood in PG-GMM training on the Kodak PhotoCD Dataset.

that the number of iterations is T , the overall complexity of PG-GMM training is $O(p^6 MNT)$.

2.2.4 Discussions

GMM has been used for patch based image prior learning and achieved promising results, e.g., EPLL [epll] and PLE [ple]. In this paper, we extend the patch based image prior learning to PG based prior learning to model the NSS information. The developed PG-GMM method has some important advantages over the patch based GMM method.

First, in patch based GMM, the mean value of each patch is subtracted before learning the Gaussian components. This is to remove the DC (direct current) of each patch but will not change the essential structure of a patch. However, in PG-GMM the mean vector of all patches in a group is calculated and subtracted from each patch, and hence the structure of each patch is changed. As a result, many patches which originally have different local patterns may become similar after group mean subtraction (please refer to Fig. 2.2 for an example). This makes the PG-GMM learning process easier and more stable.

Second, as can be seen in Eq. (2.9), the mean vector of each Gaussian component in PG-GMM is naturally a zero vector. This implies that we only need to learn the covariance matrix of each component without considering

its mean. However, in patch based GMM [epll], the mean vectors of Gaussians can only be forced to zero and there is no theoretical guarantee for this.

Third, due to reduction of possible patterns in PG-GMM and the reduced number of variables to learn, we do not need to set a large number of Gaussian components in PG-GMM learning. For example, in EPLL [epll], 200 Gaussian components are learned to achieve competing denoising performance with BM3D [bm3d], while in PG-GMM learning only 32 Gaussian components are enough to outperform BM3D (please refer to the experimental section for details).

2.3 Image Denoising by Patch Group Priors

2.3.1 Denoising Model

Given a noisy image \mathbf{y} , like in the PG-GMM learning stage, for each local patch we search for its similar patches in a window centered on it to form a PG, denoted by $\mathbf{Y} = \{\mathbf{y}_1, \dots, \mathbf{y}_M\}$. Then the group mean of \mathbf{Y} , denoted by μ_y , is calculated and subtracted from each patch, leading to the mean subtracted PG $\bar{\mathbf{Y}}$. We can write $\bar{\mathbf{Y}}$ as $\bar{\mathbf{Y}} = \bar{\mathbf{X}} + \mathbf{V}$, where $\bar{\mathbf{X}}$ is the corresponding clean PG and \mathbf{V} contains the corrupted noise. The problem then turns to how to recover $\bar{\mathbf{X}}$ from $\bar{\mathbf{Y}}$ by using the learned PG-GMM priors. Note that the mean μ_y of $\bar{\mathbf{Y}}$ is very close to the mean of $\bar{\mathbf{X}}$ since the mean vector of noise \mathbf{V} is nearly zero. μ_y will be added back to the denoised PG to obtain the denoised image.

Gaussian Component Selection

For each $\bar{\mathbf{Y}}$, we select the most suitable Gaussian component to it from the trained PG-GMM. As in [epll], suppose that the variance of Gaussian white noise corrupted in the image is σ^2 , the covariance matrix of the k th component will become $\Sigma_k + \sigma^2 \mathbf{I}$, where \mathbf{I} is the identity matrix. The selection can be done by checking the posterior probability that $\bar{\mathbf{Y}}$ belongs to the k th Gaussian component:

$$P(k|\bar{\mathbf{Y}}) = \frac{\prod_{m=1}^M \mathcal{N}(\bar{\mathbf{y}}_m | \mathbf{0}, \Sigma_k + \sigma^2 \mathbf{I})}{\sum_{l=1}^K \prod_{m=1}^M \mathcal{N}(\bar{\mathbf{y}}_m | \mathbf{0}, \Sigma_l + \sigma^2 \mathbf{I})}. \quad (2.8)$$

Taking log-likelihood of (2.8), we have

$$\ln P(k|\bar{\mathbf{Y}}) = \sum_{m=1}^M \ln \mathcal{N}(\bar{\mathbf{y}}_m | \mathbf{0}, \Sigma_k + \sigma^2) - \ln C \quad (2.9)$$

where C is the denominator in Eq. (2.9) and it is the same for all components. Finally, the component with the highest probability $\ln P(k|\bar{\mathbf{Y}})$ is selected to process $\bar{\mathbf{Y}}$.

Weighted Sparse Coding with Closed-Form Solution

Suppose that the k th Gaussian component is selected for PG $\bar{\mathbf{Y}}$. For notation simplicity, we remove the subscript k and denote by Σ the covariance matrix of this component. In PG-GMM, the PGs actually represent the variations of the similar patches in a group, and these variations are assigned to the same Gaussian distribution. By singular value decomposition (SVD), Σ can be factorized as

$$\Sigma = \mathbf{D}\Lambda\mathbf{D}^T, \quad (2.10)$$

where \mathbf{D} is an orthonormal matrix composed by the eigenvectors of Σ and Λ is the diagonal matrix of eigenvalues. With PG-GMM, the eigenvectors in \mathbf{D} capture the statistical structures of NSS variations in natural images, while the eigenvalues in Λ represent the significance of these eigenvectors. Fig. 4 shows the eigenvectors for 3 Gaussian components. It can be seen that these eigenvectors encode the possible variations of the PGs. For one Gaussian component, the first eigenvector represents its largest variation, while the last eigenvector represents its smallest variation. For different Gaussian components, we can see that their eigenvectors (with the same index) are very different. Hence, \mathbf{D} can be used to represent the structural variations of the PGs in that component.

For each patch $\bar{\mathbf{y}}_m$ in the PG $\bar{\mathbf{Y}}$, we propose to use \mathbf{D} as the dictionary to sparsely encode $\bar{\mathbf{y}}_m$ as $\bar{\mathbf{y}}_m = \mathbf{D}\alpha + \mathbf{v}$, where α is the vector of sparse coding coefficients and \mathbf{v} is the corrupted noise. Meanwhile, we propose to introduce a weighting vector \mathbf{w} to weight the coding vector α (we will see in (2.16) that \mathbf{w} is related to the eigenvalues in Λ), resulting in the following simple but highly effective weighted sparse coding model:

$$\min_{\alpha} \|\bar{\mathbf{y}}_m - \mathbf{D}\alpha\|_2^2 + \|\mathbf{w}^T \alpha\|_1. \quad (2.11)$$

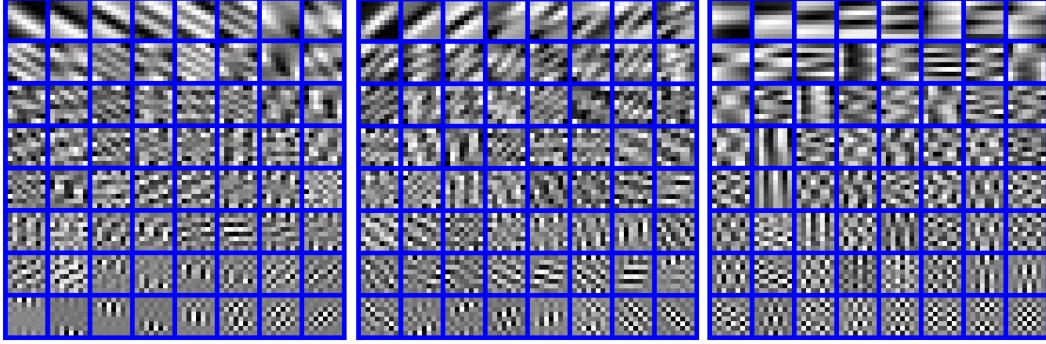


Fig. 2.4: Eigenvectors of 3 Gaussian components from the learned PG-GMM, sorted by the values of corresponding eigenvalues.

From the viewpoint of Maximum A-Posterior (MAP) estimation, the optimal solution of (2.11) is $\hat{\boldsymbol{\alpha}} = \arg \max_{\boldsymbol{\alpha}} \ln P(\boldsymbol{\alpha} | \bar{\mathbf{y}}_m)$. By Bayes' formula, it is equivalent to

$$\hat{\boldsymbol{\alpha}} = \arg \max_{\boldsymbol{\alpha}} \{ \ln P(\bar{\mathbf{y}}_m | \boldsymbol{\alpha}) + \ln P(\boldsymbol{\alpha}) \}. \quad (2.12)$$

The log-likelihood term $\ln P(\bar{\mathbf{y}}_m | \boldsymbol{\alpha})$ is characterized by the statistics of noise \mathbf{v} , which is assumed to be white Gaussian with standard deviation σ . Hence, we have

$$P(\bar{\mathbf{y}}_m | \boldsymbol{\alpha}) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{1}{2\sigma^2} \|\bar{\mathbf{y}}_m - \mathbf{D}\boldsymbol{\alpha}\|_2^2\right). \quad (2.13)$$

We assume that the sparse coding coefficients in $\boldsymbol{\alpha}$ follow i.i.d. Laplacian distribution. More specifically, for entry α_i , which is the coding coefficient of patch $\bar{\mathbf{y}}_m$ over the i th eigenvector in \mathbf{D} , we assume that it follows distribution $\frac{c}{\sqrt{2\lambda_i}} \exp(-c\sqrt{2}|\alpha_i|/\lambda_i)$, where $\lambda_i = \Lambda_i^{1/2}$ and c is a constant. Note that we adjust the scale factor of the distribution by (square root of) the i th eigenvalue Λ_i . This is because the larger the eigenvalue Λ_i is, the more important the i th eigenvector in \mathbf{D} is, and hence the distribution of the coding coefficients over this eigenvector should have a longer tail (i.e., less sparse). Finally, we have

$$P(\boldsymbol{\alpha}) = \prod_{i=1}^{p^2} \frac{c}{\sqrt{2\lambda_i}} \exp\left(-\frac{c\sqrt{2}|\alpha_i|}{\lambda_i}\right). \quad (2.14)$$

Putting (2.13) and (2.14) into (2.12), we have

$$\hat{\boldsymbol{\alpha}} = \arg \min_{\boldsymbol{\alpha}} \|\bar{\mathbf{y}}_m - \mathbf{D}\boldsymbol{\alpha}\|_2^2 + \sum_{i=1}^{p^2} \frac{c * 2\sqrt{2}\sigma^2}{\lambda_i} |\alpha_i|. \quad (2.15)$$

By comparing (2.15) with (2.11), we can see that the i th entry of the weight-ing vector \mathbf{w} should be

$$\mathbf{w}_i = c * 2\sqrt{2}\sigma^2 / (\lambda_i + \varepsilon), \quad (2.16)$$

where ε is a small positive number to avoid dividing by zero.

With \mathbf{w} determined by (2.16), let's see what the solution of (2.11) should be. Since the dictionary \mathbf{D} is orthonormal, it is not difficult to find out that (2.11) has a closed-form solution (detailed derivation can be found in the supplementary material):

$$\hat{\alpha} = \text{sgn}(\mathbf{D}^T \bar{\mathbf{y}}_m) \odot \max(|\mathbf{D}^T \bar{\mathbf{y}}_m| - \mathbf{w}/2, 0), \quad (2.17)$$

where $\text{sgn}(\bullet)$ is the sign function, \odot means element-wise multiplication, and $|\mathbf{D}^T \bar{\mathbf{y}}_m|$ is the absolute value of each entry of vector $|\mathbf{D}^T \bar{\mathbf{y}}_m|$. The closed-form solution makes our weighted sparse coding process very efficient.

2.3.2 Denoising Algorithm

With the solution $\hat{\alpha}$ in (2.17), the clean patch in a PG can be estimated as $\hat{\mathbf{x}}_m = \mathbf{D}\hat{\alpha} + \mu_y$. Then the clean image $\hat{\mathbf{x}}$ can be reconstructed by aggregating all the estimated PGs. In practice, we could perform the above denoising procedures for several iterations for better denoising outputs. In iteration t , we use the iterative regularization strategy [osher2005iterative] to add back to the recovered image $\hat{\mathbf{x}}^{(t-1)}$ some estimation residual in iteration $t-1$. The standard deviation of noise in iteration t is adjusted as $\sigma^{(t)} = \eta * \sqrt{\sigma^2 - \|\mathbf{y} - \mathbf{y}^{(t-1)}\|_2^2}$, where η is a constant. The proposed denoising algorithm is summarized in Algorithm 1 (Alg. 1).

In the proposed algorithm, there are N PGs in an image and M patches in each PG. Then the computational cost for Gaussian component selection is $O(p^6NMK)$. The cost for iterative regularization and noise estimation is negligible. The cost for closed-form weighted sparse coding is $O(p^4NM)$. Suppose that there are T iterations, the overall complexity of our denoising algorithm is $O(p^6NMKT)$.

Alg. 1: Patch Group Prior based Denoising (PGPD)

Input: Noisy image \mathbf{y} , PG-GMM model

1. Initialization: $\hat{\mathbf{x}}^{(0)} = \mathbf{y}, \mathbf{y}^{(0)} = \mathbf{y}$;

for $t = 1 : IteNum$ **do**

2. Iterative Regularization:

$$\mathbf{y}^{(t)} = \hat{\mathbf{x}}^{(t-1)} + \delta(\mathbf{y} - \mathbf{y}^{(t-1)});$$

3. Estimate the standard deviation of noise;

for each PG Y **do**

 4. Calculate group mean μ_y and form PG \bar{Y} ;

 5. Gaussian component selection via (2.9);

 6. Denoising by Weighted Sparse Coding (2.15);

 7. Recover each patch in this PG via $\hat{\mathbf{x}}_m = \mathbf{D}\hat{\boldsymbol{\alpha}} + \mu_y$;

end for

8. Aggregate the recovered PGs to form the recovered image $\hat{\mathbf{x}}^{(t)}$;

end for

Output: The recovered image $\hat{\mathbf{x}}^{(IteNum)}$.

2.4 Experiments

In this section, we perform image denoising experiments on 20 widely used natural images (shown in Fig. ??). More experiments on the Berkeley Segmentation Data Set [bsds] can be found in the supplementary file. As a common experimental setting in literature, additive white Gaussian noise with zero mean and standard deviation σ is added to the image to test the performance of competing denoising methods. We call our method *PG Prior based Denoising* (PGPD) in the following experiments. The Matlab source code of our PGPD algorithm can be downloaded at <http://www4.comp.polyu.edu.hk/~cslzhang/code/PGPD.zip>.

2.4.1 Implementation Details

Our proposed PGPD method contains two stages, the prior learning stage and the denoising stage. In the PG-GMM learning stage, there are 4 parameters: p , M , W and K . The patch size ($p \times p$) is set as $p = 6$ for $0 < \sigma \leq 20$, $p = 7$ for $20 < \sigma \leq 30$, $p = 8$ for $30 < \sigma \leq 50$, and $p = 9$ for $50 < \sigma \leq 100$. The window size (W) for PG searching is set to $W = 31$. The number (M) of patches in a PG is set to $M = 10$. The number (K) of Gaussian components is set to $K = 64$ for $p = 6$ and $K = 32$ otherwise. We extracted about one million PGs from the Kodak PhotoCD Dataset to train the PG-GMM.

In the denoising stage, there are 3 parameters: c , δ , and η . In our implementation, (c, δ, η) are set to $(0.33, 0.10, 0.79)$, $(0.29, 0.09, 0.73)$, $(0.19, 0.08, 0.89)$, $(0.15, 0.07, 0.98)$, $(0.12, 0.06, 1.05)$, $(0.09, 0.05, 1.15)$, $(0.06, 0.05, 1.30)$ when $\sigma = 10, 20, 30, 40, 50, 75, 100$, respectively. In addition, on all noise levels we stop Algorithm 1 in 4 iterations.

2.4.2 Comparison Methods

We compare the proposed PGPD algorithm with BM3D [**bm3d**], EPLL [**epll**], LSSC [**lssc**], NCSR [**ncsr**], and WNNM [**wnnm**], which represent the state-of-the-arts of modern image denoising techniques and all of them exploit image NSS. The source codes of all competing algorithms are downloaded from the authors' websites and we use the default parameter settings.

To more clearly demonstrate the effectiveness of PG based NSS prior learning, we also compare with an extreme case of PGPD, i.e., letting $M = 1$ in the PG-GMM learning stage¹. Clearly, this reduces to a patch based prior learning scheme and no NSS prior will be learned. We call this extreme case as *Patch Prior based Denoising* (PPD). The number of Gaussian components in PPD is set to 64, and the weighted sparse coding framework in it is the same as that in PGPD. All the other parameters in PPD are tuned to achieve its best performance.

2.4.3 Results and Discussions

We evaluate the competing methods from three aspects: PSNR, Speed, and Visual Quality.

PSNR. In Table 2.1, we present the PSNR results on four noise levels $\sigma = 30, 40, 50, 75$. The results on noise levels $\sigma = 10, 20, 100$ can be found in the supplementary material. From Table 2.1, we have several observations. Firstly, PGPD achieves much better PSNR results than PPD. The improvements are 0.24~0.52dB on average. This clearly demonstrates the effectiveness of PG-GMM in NSS prior learning. Secondly, PGPD has higher PSNR values than BM3D, LSSC, EPLL and NCSR, and is only slightly inferior to WNNM.

¹Since there is only 1 patch in the PG, the group mean vector cannot be subtracted and we subtract the mean value of the patch from it.

However, PGPD is much more efficient than WNNM (see next paragraph). This validates the strong ability of PG based NSS prior in image denoising.

Speed. Efficiency is another important factor to evaluate an algorithm. We then compare the speed of all competing methods. All experiments are run under the Matlab2014b environment on a machine with Intel(R) Core(TM) i7-4770K CPU of 3.50GHz and 12.0 GB RAM. The run time (s) of competing methods on the test images is shown in Table 2.2. One can easily see that BM3D is the fastest method. The proposed PGPD is the second fastest, and it is much faster than the other methods. For a 256×256 image, BM3D costs about 0.8s while PGPD costs about 10s. However, please note that BM3D is implemented with compiled C++ mex-function and with parallelization, while PGPD is implemented purely in Matlab. EPLL is about 4 times slower than PGPD. Both LSSC and NCSR are very slow since they need to train online dictionary. Though WNNM has the highest PSNR, it suffers from huge computational cost due to the many online SVD operations. It is 10~16 times slower than PGPD.

Visual Quality. Considering that human subjects are the ultimate judge of the image quality, the visual quality of denoised images is also critical to evaluate a denoising algorithm. Fig. ?? and Fig. ?? show the denoised images of *Airplane* and *Cameraman* by the competing methods, respectively. Due to the page limit, the results of PPD are not shown here, and more visual comparisons can be found in the supplementary file. We can see that BM3D tends to over-smooth the image, while EPLL, LSSC, NCSR and WNNM are likely to generate artifacts when noise is high. Owe to the learned NSS prior, the proposed PGPD method is more robust against artifacts, and it preserves edge and texture areas much better than the other methods. For example, in image *Airplane*, PGPD reconstructs the numbers “01568” more clearly than all the other methods including WNNM. In image *Cameraman*, PGPD recoveries more faithfully the fine structures of the camera area.

In summary, the proposed PGPD method demonstrates powerful denoising ability quantitatively and qualitatively, and it is highly efficient.

2.5 Conclusion

How to learn explicit models of nonlocal self-similarity (NSS) prior for image restoration is an open problem, and we made a good attempt on this by lifting the patch based image modeling to patch group (PG) based image modeling. A PG is a group of similar patches in an image region. After group mean subtraction, a PG can naturally represent the NSS variations of natural images. A PG based Gaussian Mixture Model (PG-GMM) learning algorithm was developed to learned the NSS prior from natural images, and an associated weighted sparse coding algorithm was developed for high performance image denoising. The so-called *PG Prior based Denoising* (PGPD) algorithm not only achieves highly competitive PSNR results with state-of-the-art denoising methods, but also is highly efficient and preserves better the image edges and textures. The proposed method can be extended to other image processing tasks such as deblurring and super-resolution.

Table 2.1: PSNR(dB) results of different denoising algorithms on 20 natural images.

Images	$\sigma = 30$							$\sigma = 40$						
	BM3D	LSSC	EPLL	NCSR	WNNM	PPD	PGPD	BM3D	LSSC	EPLL	NCSR	WNNM	PPD	PGPD
Airfield	26.41	26.68	26.52	26.36	26.67	26.33	26.46	25.10	25.51	25.36	25.07	25.48	25.20	25.30
Airplane	30.71	30.62	30.68	30.70	30.97	30.62	30.80	29.20	29.21	29.28	29.28	29.58	29.21	29.44
Baboon	24.57	24.78	24.70	24.63	24.85	24.54	24.63	23.11	23.51	23.35	23.28	23.58	23.23	23.39
Barbara	29.81	29.60	27.64	29.62	30.31	27.97	29.38	27.99	28.17	26.06	28.20	28.76	26.29	27.97
Boat	29.12	29.06	28.97	28.94	29.24	28.80	29.05	27.74	27.77	27.72	27.65	27.96	27.51	27.82
C. Man	28.64	28.63	28.40	28.58	28.80	28.25	28.53	27.18	27.34	27.10	27.12	27.47	27.05	27.33
Carhouse	28.78	28.79	28.70	28.72	28.94	28.62	28.80	27.38	27.49	27.38	27.40	27.58	27.29	27.51
Couple	28.87	28.76	28.69	28.57	28.98	28.54	28.84	27.48	27.41	27.34	27.24	27.62	27.16	27.53
Elaine	30.45	30.54	30.26	30.26	30.46	30.24	30.37	29.52	29.55	29.46	29.59	29.60	29.42	29.62
Hat	29.37	29.22	29.22	29.16	29.44	29.05	29.31	27.74	27.60	27.73	27.66	27.85	27.43	27.90
Hill	29.16	29.09	28.94	28.97	29.25	28.85	29.09	27.99	28.00	27.86	27.83	28.12	27.76	28.06
House	32.09	32.40	31.48	32.07	32.52	31.62	32.24	30.65	31.10	30.20	30.80	31.31	30.32	31.02
Lake	28.34	28.36	28.41	28.31	28.59	28.30	28.38	26.98	27.13	27.19	26.99	27.34	27.03	27.15
Leaves	27.81	27.65	27.36	28.14	28.60	27.51	27.99	25.69	26.04	25.80	26.24	26.95	25.88	26.29
Lena	31.26	31.18	30.98	31.06	31.43	30.98	31.27	29.86	29.91	29.69	29.92	30.11	29.67	30.10
Man	28.86	28.87	28.87	28.78	29.00	28.72	28.86	27.65	27.64	27.68	27.54	27.80	27.53	27.73
Monarch	28.36	28.20	28.50	28.46	28.91	28.27	28.49	26.72	26.87	27.05	26.85	27.47	26.81	27.02
Paint	28.29	28.29	28.45	28.10	28.58	28.39	28.42	26.69	26.77	27.00	26.50	27.10	26.88	26.94
Peppers	31.26	31.17	31.10	31.11	31.38	31.13	31.25	29.97	30.00	29.93	30.07	30.18	29.95	30.18
Zelda	30.45	30.27	30.44	30.16	30.48	30.35	30.43	29.10	28.91	29.18	28.94	29.12	29.07	29.23
Average	29.13	29.11	28.92	29.03	29.37	28.85	29.13	27.69	27.80	27.62	27.71	28.05	27.53	27.88
	$\sigma = 50$							$\sigma = 75$						
Images	BM3D	LSSC	EPLL	NCSR	WNNM	PPD	PGPD	BM3D	LSSC	EPLL	NCSR	WNNM	PPD	PGPD
Airfield	24.20	24.58	24.46	24.18	24.51	24.33	24.44	22.71	22.85	22.85	22.57	22.94	22.69	22.90
Airplane	28.24	28.15	28.19	28.18	28.55	28.10	28.38	26.40	26.16	26.14	26.10	26.68	25.90	26.39
Baboon	22.35	22.60	22.35	22.43	22.73	22.30	22.47	21.11	21.18	20.85	21.03	21.36	20.71	21.09
Barbara	27.23	27.03	24.83	26.99	27.79	24.94	26.81	25.12	25.01	22.94	24.72	25.81	22.84	24.84
Boat	26.78	26.77	26.74	26.67	26.97	26.52	26.85	25.12	25.03	25.01	24.87	25.29	24.72	25.19
C. Man	26.12	26.35	26.10	26.15	26.42	26.13	26.46	24.33	24.41	24.29	24.22	24.55	24.36	24.64
Carhouse	26.53	26.48	26.39	26.41	26.67	26.27	26.53	24.89	24.85	24.65	24.53	25.04	24.44	24.85
Couple	26.46	26.35	26.30	26.19	26.65	26.07	26.50	24.70	24.51	24.51	24.33	24.85	24.22	24.70
Elaine	28.94	28.75	28.77	28.85	28.97	28.69	28.90	27.41	27.27	27.38	27.16	27.53	27.26	27.47
Hat	26.77	26.41	26.62	26.51	26.78	26.28	26.76	24.77	24.31	24.65	24.48	24.77	24.19	24.79
Hill	27.19	27.14	27.04	26.99	27.34	26.91	27.22	25.68	25.57	25.60	25.40	25.88	25.34	25.73
House	29.69	29.99	29.12	29.62	30.32	29.17	29.93	27.51	27.75	27.09	27.22	28.25	26.81	27.81
Lake	26.13	26.15	26.24	26.02	26.41	26.05	26.20	24.49	24.25	24.50	24.26	24.66	24.19	24.49
Leaves	24.68	24.78	24.55	24.96	25.47	24.56	25.03	22.49	22.17	22.12	22.60	23.06	21.94	22.61
Lena	29.05	28.95	28.68	28.90	29.25	28.61	29.11	27.26	27.22	26.88	27.00	27.54	26.68	27.40
Man	26.81	26.72	26.79	26.67	26.94	26.63	26.86	25.32	25.10	25.26	25.10	25.42	25.01	25.36
Monarch	25.82	25.88	25.94	25.76	26.31	25.66	26.00	23.91	23.66	23.88	23.67	24.31	23.51	24.00
Paint	25.67	25.59	25.87	25.36	25.98	25.70	25.82	23.80	23.52	23.88	23.44	24.07	23.50	23.89
Peppers	29.12	29.06	28.98	29.07	29.34	28.99	29.22	27.28	27.14	27.15	26.96	27.55	27.04	27.42
Zelda	28.25	27.90	28.22	27.97	28.21	28.06	28.24	26.60	26.09	26.55	26.21	26.44	26.37	26.56
Average	26.80	26.78	26.61	26.69	27.08	26.50	26.89	25.04	24.90	24.81	24.79	25.30	24.59	25.11

Table 2.2: Average run time (seconds) with standard deviation of different methods on images of size 256×256 and 512×512 . BM3D uses parallelization and is implemented with compiled C++ mex-function while the other methods are implemented in Matlab.

	256×256						
σ	BM3D	LSSC	EPLL	NCSR	WNNM	PPD	PGPD
10	0.67 ± 0.09	186.90 ± 4.02	38.47 ± 0.10	126.43 ± 3.84	84.34 ± 1.42	10.15 ± 0.07	8.00 ± 0.05
20	0.70 ± 0.09	184.21 ± 5.82	38.47 ± 0.13	156.14 ± 5.26	84.70 ± 1.71	10.18 ± 0.15	8.09 ± 0.09
30	0.70 ± 0.09	212.07 ± 8.72	38.55 ± 0.09	149.31 ± 4.19	155.75 ± 0.94	10.34 ± 0.25	8.47 ± 0.07
40	0.67 ± 0.11	209.13 ± 6.99	38.51 ± 0.08	346.91 ± 18.65	157.35 ± 1.48	10.47 ± 0.21	9.80 ± 0.08
50	0.87 ± 0.04	221.36 ± 6.27	40.21 ± 1.82	326.93 ± 9.64	119.47 ± 4.65	10.88 ± 0.05	9.91 ± 0.13
75	0.89 ± 0.03	240.75 ± 6.08	40.91 ± 1.33	258.04 ± 11.80	179.30 ± 5.08	10.87 ± 0.27	11.73 ± 0.08
100	0.90 ± 0.03	257.25 ± 6.01	42.80 ± 1.93	252.74 ± 8.50	191.32 ± 1.47	10.90 ± 0.19	11.78 ± 0.08
	512×512						
σ	BM3D	LSSC	EPLL	NCSR	WNNM	PPD	PGPD
10	3.16 ± 0.12	746.53 ± 24.96	160.93 ± 2.81	624.83 ± 40.24	352.34 ± 3.87	41.79 ± 0.32	33.03 ± 0.25
20	3.32 ± 0.11	762.62 ± 31.25	159.80 ± 0.37	751.09 ± 42.89	351.09 ± 3.14	42.09 ± 0.41	33.26 ± 0.29
30	3.32 ± 0.09	856.82 ± 40.32	160.21 ± 0.18	709.90 ± 31.62	650.54 ± 7.23	42.36 ± 0.99	35.45 ± 0.24
40	3.18 ± 0.18	865.83 ± 40.96	160.23 ± 0.17	1620.74 ± 104.59	652.49 ± 10.49	41.70 ± 0.47	40.13 ± 0.23
50	3.85 ± 0.09	891.53 ± 48.60	161.36 ± 3.08	1492.78 ± 65.87	476.50 ± 12.34	41.75 ± 0.64	40.40 ± 0.28
75	3.91 ± 0.05	983.05 ± 69.96	165.66 ± 2.62	1156.82 ± 66.37	784.92 ± 18.32	41.88 ± 0.78	50.00 ± 0.25
100	3.94 ± 0.04	1087.57 ± 68.76	177.51 ± 7.16	1100.00 ± 26.64	824.56 ± 34.41	42.80 ± 1.09	50.32 ± 0.31

External Prior Guided Internal Prior

Learning for Real Noisy Image Denoising

3

“Innovation distinguishes between a leader and a follower.”

— Steve Jobs
(CEO Apple Inc.)

3.1 Learning External Nonlocal Self-Similarity Priors



Fig. 3.1: Figure example: (a) example part one, (c) example part two; (c) example part three

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between

this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.



Fig. 3.2: Another Figure example: (a) example part one, (c) example part two; (c) example part three

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

3.2 System Design

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

3.3 Demo System

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

3.4 Calibration

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

3.5 Conclusion

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

Internal Nonlocal Self-Similarity

Prior for Real Color Image

Denoising: A Low Rank based Method

“Users do not care about what is inside the box, as long as the box does what they need done.”

— Jef Raskin
about Human Computer Interfaces

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

4.1 Introduction

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language.

There is no need for special content, but the length of words should match the language.

4.2 Related Work

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

4.3 Method

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

4.4 Experimental Results

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language.

There is no need for special content, but the length of words should match the language.

4.5 Summary

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

Internal Nonlocal Self-Similarity

Prior for Real Color Image

Denoising: A Sparse Coding based method

“Users do not care about what is inside the box, as long as the box does what they need done.”

— Jef Raskin
about Human Computer Interfaces

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

5.1 Introduction

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language.

There is no need for special content, but the length of words should match the language.

5.2 Related Work

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

5.3 Summary

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

A Large Real Noisy Image Dataset, with A Comprehensive Evaluation of State-of-the-arts

“*Users do not care about what is inside the box, as long as the box does what they need done.*

— Jef Raskin
about Human Computer Interfaces

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

6.1 Introduction

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

6.2 Related Work

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

6.3 Summary

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

Conclusions

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

7.1 Section 1

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

7.2 Section 2

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language.

There is no need for special content, but the length of words should match the language.

7.3 Future Work

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.