Transductive Gaussian Processes for Image Denoising

Shenlong Wang
University of Toronto
slwang@cs.toronto.edu

Lei Zhang
Hong Kong Polytechnic University
cslzhang@comp.polyu.edu.hk

Raquel Urtasun
University of Toronto
urtasun@cs.toronto.edu

Abstract

In this paper we are interested in exploiting self-similarity information for discriminative image denoising. Towards this goal, we propose a simple yet powerful denoising method based on transductive Gaussian processes, which introduces self-similarity in the prediction stage. Our approach allows to build a rich similarity measure by learning hyper parameters defining multi-kernel combinations. We introduce perceptual-driven kernels to capture pixel-wise, gradient-based and local-structure similarities. In addition, our algorithm can integrate several initial estimates as input features to boost performance even further. We demonstrate the effectiveness of our approach on several benchmarks. The experiments show that our proposed denoising algorithm has better performance than competing discriminative denoising methods, and achieves competitive result with respect to the state-of-the-art.

1. Introduction

In recent years, camera manufactures have increased the number of units per sensor chip in order to meet the consumers' increasing demands for low cost high-resolution cameras. This has made the latest devices more sensitive to noise. Furthermore, with the boom of cellphone cameras, low-light imagery has become a real problem, making denoising an important component of most low-cost consumer devices. Despite decades of research in both image processing and computer vision communities, we are still in need of good denoising algorithms.

During the past decade, generative models have played a dominant role in image denoising. This is due to the fact that denoising is an ill-posed problem, and prior models can help disambiguate between the set of possible solutions. However, these models are limited by the fact that the employed prior models are relatively simplistic and do not capture well the statistics of neither natural images nor real-world noise processes.

More recently, several approaches have used discriminative models for denoising [4, 11, 19], directly modeling the conditional distribution between input features computed from noisy input images and output clean images. As a consequence these methods do not need to explicitly parameterize natural images. In this paper we argue that most discriminative approaches fail to use the information contained within the test image, which is key for accurate denoising. Utilizing self-similarity entails extending data-driven methods to be transductive, taking into account the test data when learning. A notable exception is the work of Mosseri *et al.* [14], which utilized reweighed sums of nearest neighbors collected from both training and testing patches. However, a heuristic was employed to balance the importance of training and testing examples, and only very simple statistical models (i.e., nearest neighbors), which require large collection of training examples to generalize well, were exploited.

In this paper, we propose a simple yet powerful discriminative denoising method based on transductive Gaussian processes, which is able to exploit self-similarity. Towards this goal, we propose several perceptual-driven kernels that capture pixel-wise, gradient-based and local-structure similarities. Furthermore, hyper parameters can be learned in an easy and principled way, avoiding the use of heuristics. In addition, our algorithm can integrate several initial estimations as inputs to boost the performance even further. Our experiments show that our proposed denoising algorithm has better performance than competing discriminative denoising methods on two different benchmark datasets, and achieves competitive result with respect to the state-of-the-art.

In the following, we first conduct a literature review on existing denoising methods and their relationships with our proposed method. We then discuss our proposed method in detail, show our experimental evaluation and conclusions.

2. Related Work

Most previous image restoration methods are based on generative models. The key issue in those approaches is how to construct a suitable image prior. A variety of natural image prior models have been proposed. A popular approach is to use a Markov random field (MRF) to encode pixel similarity in a local neighborhood [16, 5, 17, 19, 10]. The connectivity employed is either a grid, which includes most gradient-based prior models [5] or an MRF with high-order cliques [16, 17, 19, 10]. Another popular approach exploits patch-based mixture models [15, 25, 26]. Gaussian mixture models (GMMs) still perform among the best to model image statistics [25, 26]. Sparse coding [9, 13, 7] is also an effective way to model natural image statistics. These methods mainly focus on modeling complex probability distributions over high-dimensional spaces, and assume that pixels are only correlated among local regions. Another alternative exploits image self-similarities in large neighborhoods [3, 24, 12]. These approaches utilize highly correlated contents within the test image to impose similar noisy input image patches to have similar outputs. State-of-the-art generative methods combine different sources of information to achieve better results [6, 13, 14]. Despite decades of research, generative models still have limitations due to the fact that the employed prior models are over-simplistic compared with the highly complex statistics of natural images. Moreover, in real-world applications, due to the difficulties in modeling the noise-generating mechanism during photography, many types of noise cannot be explicitly modeled under some well-known probability distribution assumptions. Under such circumstances, it is difficult to use generative models for denoising, even if a good image prior can be acquired. Therefore, building strong probabilistic model to learn conditional relations between noisy and clean images pairs is a reasonable solution.

With the development of statistical learning methods, researchers have recently begun to tackle image restoration problems in a discriminative way, achieving promising results [19, 4, 11]. In these works, the parameters of the models are learned from training samples. A notable example is the Gaussian conditional random field (GCRF) method proposed by Tappen *et al.* [19]. In GCRF, Gaussian potential functions are adopted due to their efficiency and an anisotropic weighting function is introduced to reduce over-smoothing. Jancsary *et al.* [11] proposed a non-parametric graphical model called regression tree field (RTF), where each leaf is a single loss-specific GCRF. This method achieves best results based on ensemble of several state-of-the-arts methods. Burger *et al.* [4] proposed to train a large scale multi-layer perceptron (MLP) on millions of natural image patch pairs (clean and noisy). While effective, all these discriminative methods share a common drawback, that is, they fail to fully use the nonlocal information contained within the test image, which we believe is key for accurate denoising.

Zontak and Irani tried to overcome this drawback [24]. They argued that 'complex' patches (with higher gradient magnitude) can be constructed better from training samples, while smoothed regions where gradients are dominated by noise can be constructed better with samples from the test images themselves. According to this observation, they proposed a heuristic informative measure called $PatchSNR$ to estimate clean images by seeking a trade-off weighted sum of training and testing samples. This heuristic only exploits very simple statistical models (i.e., nearest neighbors) which require large collection of training examples to generalize well.

3. Transductive Gaussian Processes

In this section, we propose to use transductive Gaussian processes for image denoising. We then introduce perceptual quality kernels and show how to learn the parameters of multiple kernel combinations in an easy and principled way.

3.1. Gaussian Processes for Image Denosing

We start our discussion by reviewing Gaussian process regression in the context of image denoising. Let $\mathbf{x} \in \mathcal{X}$ be the features extracted from the degraded images and let $\mathbf{y} \in \mathcal{Y}$ be the desired clean output. Discriminative approaches predict by maximizing the posterior probability as follows.

$$\hat{\mathbf{y}} = \arg\max_{\mathbf{y}\in\mathcal{Y}} p(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta}) \tag{1}$$

where $\boldsymbol{\theta}$ are the parameters of the conditional probability. Different from most of the existing generative methods, we do not rewrite the posterior into likelihood and prior, instead, we tackle this problem from a discriminative perspective, and directly estimate the output by learning a predictive function $g(\mathbf{x}) : \mathcal{X} \to \mathcal{Y}$ from training data. Note that here $\mathbf{x}$ and $\mathbf{y}$ are defined at the local patch level and overlapping patches are combine by averaging the responses. Due to the richness of image content and complexity of image noise, it is difficult to have an explicit model describing the relationship between $\mathbf{x}$ and $\mathbf{y}$. Instead, we use a non-parametric model, which assumes a GP prior $g(\mathbf{x}) \sim \mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}_i, \mathbf{x}_j))$ with $m(\mathbf{x}) = 0$, i.e.:

$$p(\mathbf{g}|\mathbf{X}) \sim \mathcal{N}(0, \mathbf{K}) \tag{2}$$

where $\mathbf{X} = [\mathbf{x}_1^{\text{train}}, ..., \mathbf{x}_N^{\text{train}}, \mathbf{x}_1^{\text{test}}, ..., \mathbf{x}_M^{\text{test}}]$ are the input features of N training samples and M testing samples, and $\mathbf{K}$ is a kernel matrix $K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$, with a valid kernel function $k(\mathbf{x_1}, \mathbf{x}_2) : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$. We denote $\mathbf{X}^{\text{train}}$ and $\mathbf{X}^{\text{test}}$ as matrices for training and testing data respectively. For simplicity we rewrite the kernel matrices $\mathbf{K}^{\text{train}}$ as $K(\mathbf{X}^{\text{train}}, \mathbf{X}^{\text{train}})$, $\mathbf{K}^{\text{cross}}$ as $K(\mathbf{X}^{\text{train}}, \mathbf{X}^{\text{test}})$ and $\mathbf{K}^{\text{test}}$ as $K(\mathbf{X}^{\text{test}}, \mathbf{X}^{\text{test}})$. For unknown observations $\mathbf{X}^{\text{test}}$, the posterior over $\mathbf{y}^{\text{test}}$ has a simple Gaussian form: $p(\mathbf{y}^{\text{test}}|\mathbf{X}^{\text{train}}, \mathbf{Y}^{\text{train}}, \mathbf{X}^{\text{test}}, \boldsymbol{\theta}) \sim \mathcal{N}(\boldsymbol{\mu}_y, \boldsymbol{\Sigma}_y)$, where:

$$\begin{aligned} \boldsymbol{\mu}_y &= \mathbf{K}^{\text{cross}\prime}(\sigma^2\mathbf{I} + \mathbf{K}^{\text{train}})^{-1}\mathbf{y}^{\text{train}} \\ \boldsymbol{\Sigma}_y &= \mathbf{K}^{\text{test}} - \mathbf{K}^{\text{cross}\prime}(\sigma^2\mathbf{I} + \mathbf{K}^{\text{train}})^{-1}\mathbf{K}^{\text{test}} \end{aligned} \tag{3}$$

Under the Gaussian assumption, $\boldsymbol{\mu}_y$ is the Bayes optimal estimator

$$\hat{f}(\mathbf{x}) = \boldsymbol{\mu}_y = \arg\max_y p(\mathbf{y}^{\text{test}} | \mathbf{X}^{\text{train}}, \mathbf{Y}^{\text{train}}, \mathbf{X}^{\text{test}}, \boldsymbol{\theta}) \tag{4}$$

For each single input $\mathbf{x}$, we define the kernel matrix between training and testing samples to be $\mathbf{K}^{\text{cross}} = [k(\mathbf{x}, \mathbf{x}_1^{\text{train}}), ..., k(\mathbf{x}, \mathbf{x}_N^{\text{train}})]$. We use this to rewrite $\boldsymbol{\mu}_y$ defined in Eq. (3) to get the Bayes optimal estimator $\hat{f}(\mathbf{x})$:

$$\hat{f}(\mathbf{x}) = \sum_{i=1}^{N} w_i k(\mathbf{x}, \mathbf{x}_i^{\text{train}}) \tag{5}$$

where the weight vector $\mathbf{w} \in \mathbb{R}^N$ is:

$$\mathbf{w} = (\sigma^2 \mathbf{I} + \mathbf{K}^{\text{train}})^{-1} \mathbf{y}^{\text{train}} \tag{6}$$

3.2. Transductive Regression

In natural image restoration, it has been proven that self-similarity information is crucial for prediction. Due to the recurrence of local image patterns, the test image itself may contains local patches that have very similar patterns. According to Zontak and Irani [24] this extent of self-similarity can only be achieved by hundreds of thousands of external image patches. In our method, a simple transductive regressor can then be used to introduce self-similarity. Intuitively, for a given local patch $\mathbf{x}^j$ in the test image, we expect that there exist some other patches with estimated outputs $\hat{\mathbf{y}}^{\text{test}/j}$ similar its denoised output $\hat{\mathbf{y}^j}$. We can substitute $\mathbf{K}$ in Eq (2) with our transductive kernel:

$$\mathbf{K} = \begin{bmatrix} \mathbf{K}^{\text{train}} & \mathbf{K}^{\text{train,test/j}} & \mathbf{K}^{\text{train},j} \\ \mathbf{K}^{\text{test}/j,\text{train}} & \mathbf{K}^{\text{test}/j} & \mathbf{K}^{\text{test}/j,\text{j}} \\ \mathbf{K}^{j,\text{train}} & \mathbf{K}^{j,\text{test}/j} & 1 \end{bmatrix} \tag{7}$$

Assume $\mathbf{y}^{\text{test}/j}$ is known, we can predict $\mathbf{y}^j$ by considering $(\mathbf{y}^{\text{test}/j}, \mathbf{X}^{\text{test}/j})$ as training pairs as follows,

$$\hat{f}(\mathbf{x}_j) = \sum_{i=1}^{N} w_i^{\text{train}} k(\mathbf{x}_j, \mathbf{x}_i^{\text{train}}) + \sum_{i=1}^{M-1} w_i^{\text{test/j}} k(\mathbf{x}_j, \mathbf{x}_i^{\text{test/j}}) \tag{8}$$

with $\mathbf{w}^{\text{train}} = (\sigma^2 \mathbf{I} + \mathbf{K}^{\text{train}})^{-1} \mathbf{y}^{\text{train}}$ and $\mathbf{w}^{\text{test/j}} = (\sigma^2 \mathbf{I} + \mathbf{K}^{\text{test/j}})^{-1} \hat{\mathbf{y}}^{\text{test/j}}$, where the initial estimation $\hat{\mathbf{y}}^{\text{test/j}}$ can be calculated from Eq. (5), i.e.

$$\hat{\mathbf{y}} = \mathbf{K}^{\text{cross/j}} (\sigma^2 \mathbf{I} + \mathbf{K}^{\text{train}})^{-1} \mathbf{y}^{\text{train}} \tag{9}$$

Using Eqs. (5) and (8) we have:

$$\hat{f}(\mathbf{x}_j) = \mathbf{K}^{\text{trans}} (\mathbf{K}^{\text{train}} + \sigma^2 I)^{-1} \mathbf{y}^{\text{train}} \tag{10}$$

where

$$\mathbf{K}^{\text{trans}} = \left[\mathbf{K}^{j,\text{train}} + \mathbf{K}^{j,\text{test}} (\mathbf{K}^{\text{test,test}} + \sigma^2 I)^{-1} \mathbf{K}^{\text{test,train}} \right] \tag{11}$$

From this equation we can see that the transductive setting reweights training samples not only by measuring their similarities to the test sample itself but also to nonlocal similar patches. Note that the increase in complexity of the transductive setting is small. In the standard regression setting, for each image, kernel functions will be called $\mathcal{O}(MN)$ times, where N and M are the number of testing and training image patches respectively, while in this transductive setting, due to the need of $\mathbf{K}^{\text{test}}$ the kernel functions will be called $\mathcal{O}(MN + N^2)$ times. Given that M is typically larger than N, this does not increase the complexity while introducing rich self-similarity information.

3.3. Perceptual Quality Driven Kernels

A key issue in our model is what covariance function should we use to measure the similarity between two patches. Simply representing images in $\mathbb{R}^n$ and using a linear kernel cannot measure perceptual similarity well. Fortunately, good results have been achieved in the field of perceptual image quality measurement (IQA), and many effective perceptual quality measures have been proposed [21, 18, 23]. The recent success of applying SSIM-index to image classification [2] motivates our use of a linear combination of several perceptual similarity functions

$$K(\mathbf{x}_i, \mathbf{x}_j) = \sum_q \theta_q K_q(\mathbf{x}_i, \mathbf{x}_j) \tag{12}$$

as kernel functions, where $K_q(\mathbf{x}_i, \mathbf{x}_j) : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is an IQA function.

However, considering that most IQA functions, like SSIM, do not satisfy Mercer's condition, we cannot directly use them as covariance functions. Therefore, we produce several alternative kernels which approximate three types of local image IQA measures, namely structural similarity index (SSIM), gradient magnitude similarity (GMS), as well as peak-to-noise ratio (PSNR). Firstly, for PSNR, we simply choose an RBF kernel $K_1(\mathbf{x}^1, \mathbf{x}^2) = \frac{1}{Z} \exp(\frac{(\mathbf{x}^1 - \mathbf{x}^2)^T (\mathbf{x}^1 - \mathbf{x}^2)}{h^2})$, which reflects the image similarity in terms of Eulidean distance. According to Wang *et al.* [21], SSIM can be written as:

$$SSIM(\mathbf{x}^1, \mathbf{x}^2) = \frac{2\mu_{\mathbf{x}^1}\mu_{\mathbf{x}^2} + C_1}{\mu_{\mathbf{x}^1}^2 + \mu_{\mathbf{x}^2}^2 + C_1} \cdot \frac{\sigma_{\mathbf{x}^1,\mathbf{x}^2}^2 + C_2}{\sigma_{\mathbf{x}^1}^2 + \sigma_{\mathbf{x}^2}^2 + C_2} \tag{13}$$

where $\mu_{\mathbf{x}^1}, \mu_{\mathbf{x}^2}$ are the mean of $\mathbf{x}^1, \mathbf{x}^2$ respectively, $\sigma_{\mathbf{x}^1}^2, \sigma_{\mathbf{x}^2}^2$ are the variance, and $\sigma_{\mathbf{x}^1,\mathbf{x}^2}^2$ is the covariance. Clearly, under the assumptions that $\mu_{\mathbf{x}^1} = \mu_{\mathbf{x}^2}$ and $\sigma_{\mathbf{x}^1} =$

$\sigma_{\mathbf{x}^2}$, we have

$$SSIM(\mathbf{x}^1, \mathbf{x}^2) = \frac{\sigma_{\mathbf{x}^1,\mathbf{x}^2} + C_2}{\sigma^2_{\mathbf{x}^1} + \sigma^2_{\mathbf{x}^2} + C_2} \tag{14}$$

$$= \frac{\langle \mathbf{x}^1 - \mu_{\mathbf{x}^1}, \mathbf{x}^2 - \mu_{\mathbf{x}^2} \rangle + C_2}{(\sqrt{2\sigma_x^2 + C_2})^2} \tag{15}$$

Motivated by this, we use $K_2(\mathbf{x}^1, \mathbf{x}^2) = \phi_2(\mathbf{x}^1)^T \phi_2(\mathbf{x}^2)$ as the SSIM-describing perceptual kernel, where the feature map is defined as $\phi_2(\mathbf{x}) = \frac{\mathbf{x} - \mu_x}{\sqrt{\sigma_x^2 + C_2/2}}$. In fact, as discussed by Wang *et al.* [21], this term plays the most vital role in describing structural-similarity. This kernel satisfies the Mercer's condition, therefore, we use it to compute the structural similarity. In addition, Xue *et al.* [22] proposed the gradient magnitude similarity (GMS), which is another good way to measure perceived similarities, as the human visual system is very sensitive to gradient variations. GMS is defined as

$$GMS(\mathbf{x}^1, \mathbf{x}^2) = \frac{\sigma_{A\mathbf{x}^1, A\mathbf{x}^2}}{\sigma^2_{A\mathbf{x}^1} + \sigma^2_{A\mathbf{x}^2} + C} \tag{16}$$

where A is a gradient operator. Similarly to SSIM, by assuming $\sigma^2_{A\mathbf{x}^1} = \sigma^2_{A\mathbf{x}^2}$, we get the GMS-based perceptual kernel $K_3(\mathbf{x}^1, \mathbf{x}^2) = \phi_3(\mathbf{x}^1)^T \phi_3(\mathbf{x}^2)$, with feature map $\phi_3(\mathbf{x}) = \frac{A\mathbf{x} - \mu_{A\mathbf{x}}}{\sqrt{\sigma^2_{A\mathbf{x}} + C}}$. We use the filter-banks provided in Tappen *et al.* [19] [1], choosing two first-order derivative filters and three second-order derivative filters.

In high noise regimes and for small local patches the magnitude of noise is dominant, which severely influences the accuracy of the similarity computation. However, choosing multiple kernels as described above improves the robustness for computing the similarity.

3.4. Learning Parameters

In the training stage, we optimize our parameters $\boldsymbol{\theta}$ by minimizing the negative log-likelihood on training data:

$$\begin{aligned} \boldsymbol{\theta}^* &= \arg\min_{\boldsymbol{\theta}} - \log p(\mathbf{y}^{\text{train}} | \mathbf{X}^{\text{train}}, \boldsymbol{\theta}) \\ &= \arg\min_{\boldsymbol{\theta}} \mathbf{y}^{\text{train}^T} \boldsymbol{\Sigma}^{-1} \mathbf{y}^{\text{train}} + \log |\boldsymbol{\Sigma}| \end{aligned} \tag{17}$$

where $\boldsymbol{\Sigma} = \mathbf{K}^{\text{train}} + \sigma^2 \mathbf{I}$. The partial derivative of the loss function w.r.t θ_q in Eq. (17) can be written as:

$$\frac{\partial \mathcal{L}}{\partial \theta_q} = \frac{1}{2} \mathbf{y}^{\text{train}^T} \boldsymbol{\Sigma}^{-1} \frac{\partial \mathbf{K}^{\text{train}}}{\partial \theta_q} \boldsymbol{\Sigma}^{-1} \mathbf{y}^{\text{train}} - \frac{1}{2} tr(\boldsymbol{\Sigma}^{-1} \frac{\partial \mathbf{K}^{\text{train}}}{\partial \theta_q}) \tag{18}$$

Since all of our parameters are linear combination parameters, the partial derivative $\frac{\partial \mathbf{K}^{\text{train}}}{\partial \theta_q}$ is equal to $\mathbf{K}_q^{\text{train}}$, a.k.a. the $q-$th kernel matrix evaluated on the training data. In our implementation, in order to make sure the weighted sum is still a valid IQA function (between 0 and 1) we impose the constraint that the sum is a convex combination. i.e. the weights sum to one. In each step after the standard gradient descent, an additional step is required to project the updated vector back onto the simplex. This can be done efficiently in $\mathcal{O}(n)$. We refer the reader to [8] for details.

3.5. Extensions

Our method can be extended in a variety of ways to further improve performance. First, we can augment the input features with the results of several existing methods. Moreover, GP has $\mathcal{O}(n^3)$ complexity for training and $\mathcal{O}(n)$ for inference, where n is the number of training examples. Similar to previous works, we also introduce sparsification for fast computation. Considering the specific clustering structures of natural image patches, we simply use clustering to partition the space. Since natural image patches are highly sparsely distributed, we argue that the boundary effects due to clustering are not significant if a proper number of clusters are chosen. For each cluster, a unique weight vector for kernel combination is learned. More sophisticated sparsification techniques such as mixture of local GPs could also be used [20].

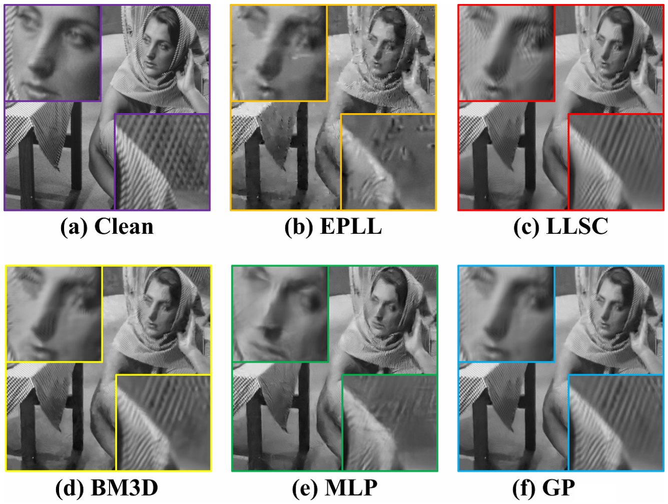
4. Experimental Evaluation

The proposed framework is simple yet generalizable. It can be further adapted to solve various image restoration problems, given some initial estimations. In this paper, we focus on its application in image denoising. Due to the space limits only partial results are shown in the paper. We refer the reader to the supplementary material for more results and visual comparisons.

Implementation Details: We use 9×9 local patches centered at the current pixel to compute all kernels, providing a good balance between speed and accuracy. The use 100 clusters in all experiments and employ a bootstrap strategy to ensure that each cluster has at least 1000 members. In order to eliminate the influence of uncorrelated patches, for each patch we only choose its 25 nearest samples to do transductive inference. Motivated by [11, 14], we also experiment by taking existing denoising methods' output as input features to our algorithm. We augment our kernels with three methods, namely BM3D, EPLL and ESSC. We employ peak-signal-to-noise ratio (PSNR), structural similarity index (SSIM) [21], and feature similarity index (FSIM) [23] as our metrics.

We conducted our first denoising experiment on 13 images (see supplementary material), which are commonly used for image denoising evaluation. We added Gaussian

[1] http://www.cs.ucf.edu/~mtappen/code/gcrf_demo.zip


(a) Clean (b) EPLL (c) LLSC

(d) BM3D (e) MLP (f) GP

Figure 1. Denoising results comparison (*barbara*) under $\sigma = 50$

white noise with 5 different standard deviations (10, 15, 20, 50, 100) to the original images to simulate noise. Our model is trained on the Kodak PhotoCD dataset, which contains 24 images. The algorithms used for initial estimates are BM3D [6], EPPL [25] and LSSC [13]. Apart from the three algorithms above, we choose FoE [16], KSVD [9], CSR [7], and MLP [4] as additional baselines as these algorithms are considered to be state-of-the-art denoising methods. As shown in Table 1 our approach outperforms all baselines in terms of PSNR. Note that learning the weights is beneficial, as shown by the ”UniAverage” baseline which employs uniform weights of value $1/3$. Fig. 1 and Fig. 2 shows a visual comparison. We can see that artifacts in all initial estimates are significantly reduced when using our proposed method, and the perceptual quality is dramatically enhanced in the final estimate obtained by our model.

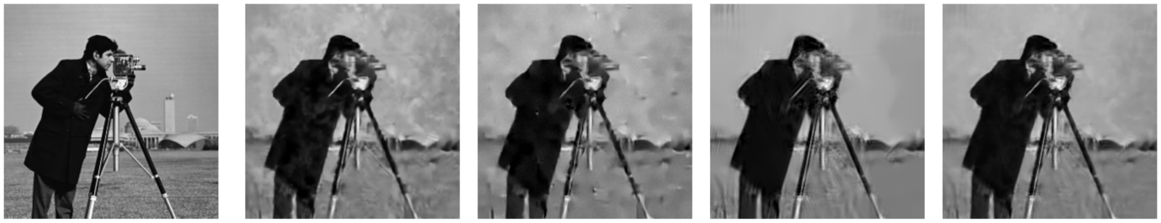
Furthermore, to validate the generalization ability of the proposed method, we use the model trained under $\sigma = 25$ to evaluate the denoising performance under different noise levels. We denote the corresponding method as GP$_{\sigma=25}$. The results are shown in the bottom row of Table 1. We can see that it also shows very competitive performance. For comparison, we report denoising results under all levels with the MLP model trained under $\sigma = 25$ (denoted as MLP$_{\sigma=25}$).

Table 1. Denoising Results on 13 Testing Images.

Noise Level	10	15	20	50	100
FoE	33.33	31.16	29.50	16.11	8.67
KSVD	33.92	31.89	30.49	25.80	22.12
BM3D	34.40	34.42	31.04	26.71	23.10
EPLL	33.79	31.78	30.39	26.04	22.91
ESSC	34.24	32.23	30.85	26.54	23.29
NSCR	34.22	32.21	30.83	26.44	23.14
MLP	34.14	-	-	26.77	-
UniAverage	34.42	32.46	31.09	26.78	23.31
GP	**34.60**	**32.75**	**31.40**	**27.19**	**23.83**
GP$_{\sigma=25}$	**34.48**	**32.65**	**31.40**	**27.10**	**23.32**
MLP$_{\sigma=25}$	29.79	30.10	30.36	17.39	11.86

We conducted our second experiment on the BSDS500 dataset [1] following exactly the protocol of Burger *et al.* [4], where 200 images in the test set are used to evaluate denoising performance. We conduct the experiment under three noise levels $\sigma = \{10, 25, 50\}$ in order to compare with MLP. Table 2 shows the average PSNR, SSIM and FSIM scores for each method under each noise level. We can see that our method is very competitive with respect to MLP. We also illustrate the PSNR gain of different competing methods agains BM3D in Fig. 3. From this figure we can see that both our algorithm and MLP have around 0.4db gain over BM3D on average. However, the proposed method is more stable than MLP as only around 2% of our results are worse than BM3D, while 7% of MLP's results are worse than BM3D. Fig. 4 shows visual comparisons between the competing algorithms.

In the next experiment we compare our algorithm and the PatchSNR approach of Mosseri *et al.* [14], which is a discriminative approach that utilizes both information from the training and test set. Unlike our transductive approach, the PatchSNR method adopts an empirical function $\sqrt{\frac{var(p)}{var(n)}}$ of local patches to measure if the denoising method should trust more the training data or the test image. The best performance of their method is achieved by utilizing this criteria to combine EPLL and BM3D. Since we do not have



(a) Clean **(b) BM3D** **(c) EPLL** **(d) LSSC** **(e) GP**

Figure 2. Denoising results comparison (*Cameraman*) under $\sigma = 50$

Table 2. Denoising Results on BSDS500 Test Dataset (**Red**: Best; **Blue**: Second Best)

Noise Level	$\sigma = 10$			$\sigma = 25$			$\sigma = 50$		
Method	PSNR	SSIM	FSIM	PSNR	SSIM	FSIM	PSNR	SSIM	FSIM
BM3D[6]	33.60	0.9254	0.9524	28.77	0.8183	0.8835	25.69	0.7077	0.8089
EPLL[25]	33.58	0.9289	0.9551	28.81	0.8254	0.8899	25.71	0.7049	0.8120
ESSC[13]	33.75	**0.9279**	0.9544	28.82	0.8246	0.8886	25.70	0.7091	0.8106
UniAverage	33.77	0.9301	**0.9549**	28.87	0.8245	0.8889	25.72	0.7088	0.8080
MLP[4]	**33.72**	0.9273	0.9539	**29.10**	**0.8332**	**0.8915**	**26.06**	**0.7256**	**0.8183**
GP[2]	**33.81**	**0.9294**	**0.9552**	**29.07**	**0.8304**	**0.8917**	**26.02**	**0.7192**	**0.8164**

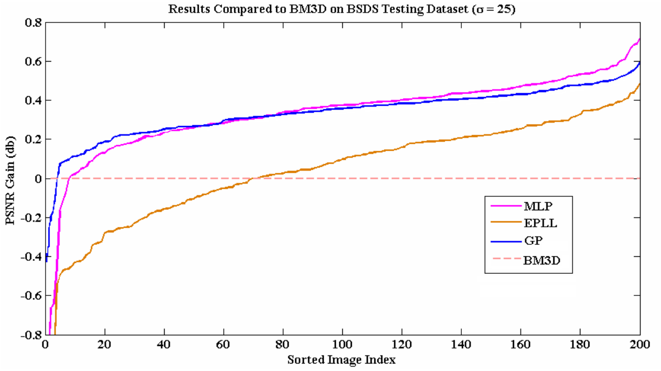
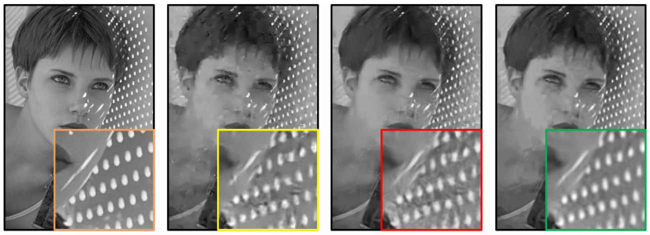


Figure 3. Sorted PSNR Gain against BM3D on the BSDS Testing Dataset.



(a) Clean **(b) EPLL** **(c) MLP** **(d) GP**

Figure 4. Denoising results comparison (BSDS *388066*) under $\sigma = 25$

the source code of PatchSNR, we follow their experimental setup, and test our method on 100 BSDS300 test images. As show in Table 3 our method outperforms the best result of PatchSNR by more than 0.1db.

Table 3. Denoising Results on BSDS300 Test Dataset

σ	BM3D	LSSC	EPLL	PatchSNR	GP
25	28.38	28.46	28.48	**28.54**	**28.66**
35	26.89	26.98	26.99	**27.07**	**27.19**
45	25.83	25.90	25.94	**26.06**	**26.17**
55	25.11	25.10	25.13	**25.29**	**25.37**

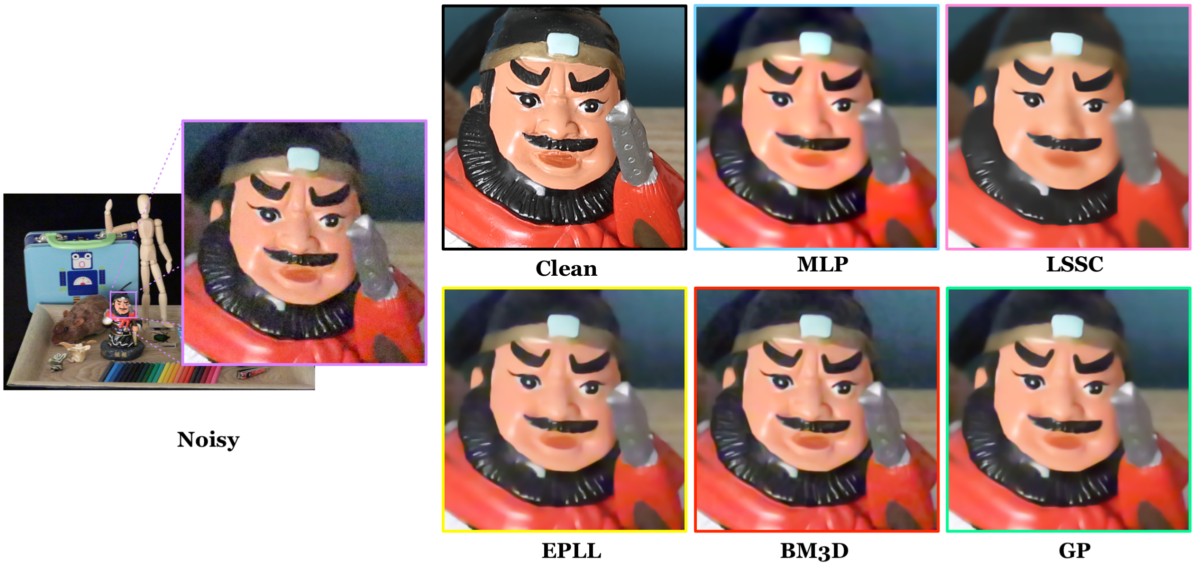
In the last experiment, we compare our algorithm to Regression Tree Fields (RTF) [11], which also employ existing denoising algorithms' outputs as input features. To ensure a fair comparison we use the same experimental setting as in [11][3]. However, in [11], the authors re-scaled the images in BSDS500 dataset to 50% of their original size, introducing a significant loss of self-similarity information. We re-run our algorithm on BSDS500 based on this setting and report the results in Table 4. Comparing Table 2 with Table 4, it can be seen that the results of our method is reduced due to the loss of self-similarity information, but it is still very competitive and outperforming all baselines but RTF.

Moreover, in order to test the real-world denoising per-

[3] We would thank the author for generously providing us the detailed configuration and their images for comparison.

Table 4. Denoising Results on BSDS500 Test Dataset with 50% Scaling. Noise Level $\sigma = 50$. (**Red**: Best; **Blue**: Second Best)

	BM3D [6]	EPLL [25]	LSSC [13]	Average	RTF$_{\text{PSNR,ALL}}$[4] [11]	MLP [4]	GP
PSNR	25.09	25.22	25.09	25.25	**25.51**	25.05	**25.39**
SSIM	0.6993	0.7029	0.7002	0.7051	**0.7170**	0.6999	**0.7156**
FSIM	0.8117	0.8073	0.8174	0.8094	**0.8239**	0.7989	**0.8194**


Noisy

Clean

MLP

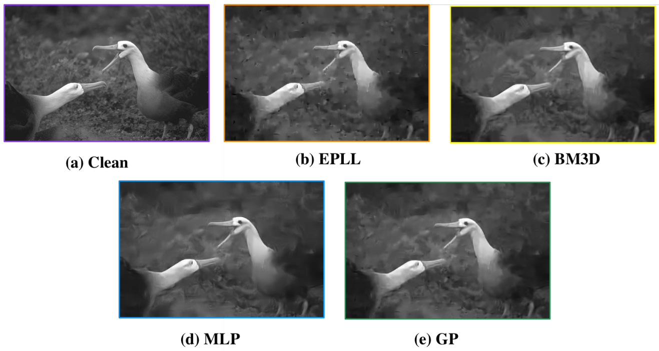
LSSC

EPLL

BM3D

GP

Figure 6. Real-world High ISO Image Denoising Results (ISO 51200)


(a) Clean

(b) EPLL

(c) BM3D

(d) MLP

(e) GP

Figure 5. Denoising results comparison (BSDS *103029*) under $\sigma = 50$

formance, we use several testing images taken under low-light conditions with high ISO settings. In this experiment, we use several testing images captured by a Canon 5D Mark III [5]. In this small testing dataset, images are of the same scene captured by fixing the camera with a tripod and employing the same exposure value by modifying shutter speed under different ISO. We directly use our model trained under the Gaussian noise settings. We pick the most appropriate noise-level σ under different ISO with a validation image. For DSLR experiment, three levels of ISO, namely 25600, 51200 and 102400 are used as noisy images and ISO50 is considered to be the clean image. Fig. 6 shows a visual comparison, showing that BM3D and EPLL keep more detailed information, while bringing color shift effects in smooth areas. MLP and LSSC keep significant boundaries sharp, but over-smooth too much detailed textures. The proposed method, seeks a better balance among keeping details, sharp edges and avoiding color-shift.

5. Conclusion

We have proposed a novel denoising method, which combines information from training data and the testing image by employing transductive Gaussian process regression. We have shown that our approach can easily combine multiple perceptual quality kernels with learned parameters. We have demonstrated the effectiveness of our approach in a wide variety of denoting tasks. Although promising, current discriminative restoration approaches, including ours, have some disadvantages. Training on degraded and clean image pairs inevitably weakens generalization ability, even

[5] `http://www.dpreview.com/galleries/reviewsamples/albums/`

if self-similarity information can alleviate this problem to some extent. This is illustrated in our experiments by the fact that 'dataset bias' happens in some methods, although millions of natural images patches have been used for training. In addition, all current discriminative methods can only be trained under a specific degrading level, which restricts their practical use. We plan to model the image degrading level as latent variables in our approach to implement blind restoration, improving its generalization ability.

References

[1] P. Arbelaez, C. Fowlkes, and D. Martin. The berkeley segmentation dataset and benchmark. 2007.

[2] D. Brunet, E. R. Vrscay, and Z. Wang. On the mathematical properties of the structural similarity index. *TIP*, 21(4):1488–1499, 2012.

[3] A. Buades, B. Coll, and J.-M. Morel. A non-local algorithm for image denoising. In *CVPR*, volume 2, pages 60–65, 2005.

[4] H. C. Burger, C. J. Schuler, and S. Harmeling. Image denoising with multi-layer perceptrons, part 1: comparison with existing algorithms and with bounds. *arXiv preprint arXiv:1211.1544*, 2012.

[5] T. S. Cho, N. Joshi, C. L. Zitnick, S. B. Kang, R. Szeliski, and W. T. Freeman. A content-aware image prior. In *CVPR*, pages 169–176, 2010.

[6] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian. Image denoising by sparse 3-d transform-domain collaborative filtering. *TIP*, 16(8):2080–2095, 2007.

[7] W. Dong, L. Zhang, G. Shi, and X. Li. Nonlocal centralized sparse representation for image restoration. *TIP*, 2013.

[8] J. Duchi, S. Shalev-Shwartz, Y. Singer, and T. Chandra. Efficient projections onto the l 1-ball for learning in high dimensions. In *ICML*, 2008.

[9] M. Elad and M. Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *TIP*, 15(12):3736–3745, 2006.

[10] J. T. Freeman W.T. and P. E.C. Example-based super-resolution. *CGA*, 22(2):56–65, 2002.

[11] J. Jancsary, S. Nowozin, and C. Rother. Loss-specific training of non-parametric image restoration models: A new state of the art. In *ECCV*, 2012.

[12] A. Levin, B. Nadler, F. Durand, and W. T. Freeman. Patch complexity, finite pixel correlations and optimal denoising. In *Computer Vision–ECCV 2012*, pages 73–86. Springer, 2012.

[13] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman. Non-local sparse models for image restoration. In *ICCV*, 2009.

[14] I. Mosseri, M. Zontak, and M. Irani. Combining the power of internal and external denoising. In *ICCP*, 2013.

[15] J. Portilla, V. Strela, M. J. Wainwright, and E. P. Simoncelli. Image denoising using scale mixtures of gaussians in the wavelet domain. *TIP*, 12(11):1338–1351, 2003.

[16] S. Roth and M. Black. Fields of experts: A framework for learning image priors. In *CVPR*, 2005.

[17] U. Schmidt, Q. Gao, and S. Roth. A generative perspective on mrfs in low-level vision. In *CVPR*, pages 1751–1758, 2010.

[18] H. R. Sheikh, M. F. Sabir, and A. C. Bovik. A statistical evaluation of recent full reference image quality assessment algorithms. *TIP*, 15(11):3440–3451, 2006.

[19] M. F. Tappen, C. Liu, E. H. Adelson, and W. T. Freeman. Learning gaussian conditional random fields for low-level vision. In *CVPR*, 2007.

[20] R. Urtasun and T. Darrell. Local Probabilistic Regression for Activity-Independent Human Pose Inference. In *CVPR*, 2008.

[21] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *TIP*, 13(4):600–612, 2004.

[22] W. Xue, L. Zhang, X. Mou, and A. C. Bovik. Gradient magnitude similarity deviation: A highly efficient perceptual image quality index. *CoRR*, abs/1308.3052, 2013.

[23] L. Zhang, L. Zhang, X. Mou, and D. Zhang. Fsim: a feature similarity index for image quality assessment. *TIP*, 20(8):2378–2386, 2011.

[24] M. Zontak and M. Irani. Internal statistics of a single natural image. In *CVPR*, 2011.

[25] D. Zoran and Y. Weiss. From learning models of natural image patches to whole image restoration. In *ICCV*, 2011.

[26] D. Zoran and Y. Weiss. Natural images, gaussian mixtures and dead leaves. In *NIPS*, 2012.