This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2015.2389797, IEEE Transactions on Pattern Analysis and Machine Intelligence

Efficient Learning of Image Super-resolution and Compression Artifact Removal with Semi-local Gaussian Processes

Younghee Kwon, Kwang In Kim, James Tompkin, Jin Hyung Kim, and Christian Theobalt

Abstract—Improving the quality of degraded images is a key problem in image processing, but the breadth of the problem leads to domain-specific approaches for tasks such as super-resolution and compression artifact removal. Recent approaches have shown that a general approach is possible by learning application-specific models from examples; however, learning models sophisticated enough to generate high-quality images is computationally expensive, and so specific per-application or per-dataset models are impractical. To solve this problem, we present an efficient *semi-local* approximation scheme to large-scale Gaussian processes. This allows efficient learning of task-specific image enhancements from example images without reducing quality. As such, our algorithm can be easily customized to specific applications and datasets, and we show the efficiency and effectiveness of our approach across five domains: single-image super-resolution for scene, human face, and text images, and artifact removal in JPEG- and JPEG 2000-encoded images.

1 Introduction

Image degradation has many different causes, from limitations of the optical system, such as limited sensor resolution or lens defocus, to lossy image compression, creating block or ring artifacts. The removal of these degradations through image processing has been approached in both application-specific and data-specific ways, with different approaches for tasks such as super-resolution and compression artifact removal, and variations for different data such as general scenes, faces, and text images [1], [2], [3], [4], [5].

The inflexibility of application-specific models of image degradation has motivated methods which try to generalize different image enhancement processes. Previous successful approaches have integrated *a priori* knowledge in a Bayesian framework in the form of a *generic* prior on natural images, and have coupled this with hand-designed application-specific parametric degradation, or *noise*, models. However, typically these noise models are difficult to design, especially for non-Gaussian noise, which makes it difficult to adapt these methods to new applications and data.

Recent work has generalized further by learning a function directly from example pairs of degraded and clean images. For instance, instead of parametric noise models, Kim and Kwong [6] apply non-parametric kernel ridge regression to map input degraded images to desired clean images, and so relieve the user from designing a noise model. However, applying models sophisticated enough to generate high-quality images to application- and data-generic image enhancement is hindered by the high computational cost of learning, e.g., [6] requires ≈36 hours of training.

Our major contribution[1] is a method to remove the high computational complexity of training a conditional noise model, without affecting final enhancement quality. Gaussian process (GP) regression is often used as it has powerful generalization capability that leads to better performance over simple nearest neighbor or linear regressors; however, learning times are prohibitive, and sparse GP approximations require solving a difficult non-linear optimization problem. We introduce a new efficient *semi-local* approximation scheme to large-scale GP regression: Instead of time-consuming training and testing of a single GP model on a large dataset, a set of *sparse* models are constructed *on-line* such that the prediction at each test data point is made by the corresponding sparse GP approximating the underlying global model. We will demonstrate that during inference, i.e., enhancement, this method has a similar run-time complexity and performance to general sparse models [6], [8], [9], [10]. However, unlike existing models, by avoiding the time-consuming training stage, our approach facilitates easy adaptation to specific image degradation problems.

As a prior, we adopt the *product of edge-perts* framework [1]. This model adopts a *sparsity* prior (i.e., Laplacian) over the pair-wise joint distribution of wavelet coefficients which, overall, prefers simultaneous activation of few coefficients in nearby scales and spatial locations [1]. As a result, the dependencies between wavelet coefficients localized in frequency, space, and scale are effectively represented through *product of experts* type factorization. With this framework and our semi-local GP regression, our algorithm allows building of an image enhancement system in 5 minutes from a set of example pairs of clean and degraded images, by deferring

- *Y. Kwon is with Google Inc., 1600 Amphitheatre Parkway, Mountain View, CA 94043, USA. youngheek@google.com*
- *K. I. Kim is with the School of Computing and Communications, Lancaster University, LA1 4WA, UK. k.kim@lancaster.ac.uk.*
- *J. Tompkin is with the School of Engineering and Applied Sciences, Harvard University, MA 02138, USA. jtompkin@seas.harvard.edu.*
- *C. Theobalt is with the Max-Planck-Institut für Informatik, Campus E1-4, 66123 Saarbrücken, Germany. theobalt@mpi-inf.mpg.de*
- *J. H. Kim is with the CS Dept., KAIST, 291 Daehak-ro, Yuseong-gu, Daejeon 305-701, Korea. jkim@kaist.ac.kr*

1. A short version appeared in Proc. BMVC 2012 [7].

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2015.2389797, IEEE Transactions on Pattern Analysis and Machine Intelligence

the learning of a case specific conditional model to an online approximation which, crucially, does not increase testing time and retains enhancement quality.

We demonstrate the performance of our algorithm with two specific applications across six datasets, one being 500 images large: with single-image super-resolution on general, face, and text images, and with artifact removal in JPEG- and JPEG 2000-encoded images, which have block and ring artifacts respectively. Our experiments show that our algorithm outperforms state-of-the-art systems that are specific to each task ([11], [12], [13], [14] for super-resolution, and [15], [16], [17], [18], [19], [20] for JPEG and JPEG 2000 artifact removal). In terms of inference or enhancement quality, our algorithm is on par with the state-of-the-art learning-based super-resolution algorithm of Kim and Kwon [6]; however, our experiments demonstrate that the significantly shorter training time of our algorithm makes per-application and per-dataset image degradation learning practical.

2 Related Work

Many image enhancement approaches estimate a function that maps from the noise-affected image space to the clean image space. In general, *a priori* knowledge about natural images can help, and, in principle, an image model incorporating a generic prior of natural images can be applied to many enhancement applications with modification of the noise model (or sometimes even with a Gaussian noise model).

The theory of *projection onto convex sets* (POCS) models prior knowledge as a set of convex constraints (e.g., spatial smoothness, quantization constraints) [21] which cast image enhancement into a POCS iteration framework [22], [23]. One direct way to use *a priori* knowledge is to encode it into a distribution or an energy functional. Roth and Black describe a *field of experts* [3], where the prior is modeled as a Markov random field (MRF) with clique potentials learned from a set of natural images.

Laparra *et al*. [18] proposed a generic wavelet domain framework where the clean and noise image distributions were estimated non-parametrically with support vector regression (SVR). As a non-parametric model, this approach can be used in various image enhancement applications. For Gaussian noise removal, this method outperformed one of the best image denoising methods — Gaussian scale mixtures (GSM) — with the perceptually-oriented structural similarity metric [24].

2.1 Learning-based single-image super-resolution

Single-image super-resolution algorithms enlarge a single low-resolution image to high resolution. Existing approaches are discussed in the literature [25], [26], [27]; the most closely related approaches to our algorithm are *example-based* methods which identify a function mapping a low-resolution image (or patch) to a high-resolution counterpart based on example pairs.

Freeman *et al*. proposed a nearest neighbor (NN)-based algorithm [14]. For each patch in the input low-resolution image, the corresponding high-resolution example patch is retrieved through NN-search that enforces spatial consistency. Chang *et al*. [11] extended this idea by additionally introducing a reconstruction constraint based on a manifold assumption.

In the context of regression estimation, Tappen *et al*. [28] performed multiple linear regressions on clustered example database and resolved the resulting multiple candidate outputs by imposing a prior on natural images. Kim and Kwon generalized this idea by combining sparse kernel ridge regression and adopting a prior on major edges [6]. Meanwhile, Yang *et al*. [12] adopted the idea of *sparse coding* in super-resolution. Here, a low-resolution input is represented as a sparse combination of stored example inputs. The combination coefficients are used to synthesize the corresponding outputs based on the retrieved example outputs.

Gaussian processes (GP) have been used in various image enhancement problems. Tipping and Bishop [29] proposed applying a GP prior for multi-frame image super-resolution while Liu [30], and He and Siu [13] used GP regression for non-example-based image denoising and super-resolution, respectively. Kim and Kwon [6] applied kernel ridge regression (KRR) that corresponds to non-Bayesian estimate of GP regression to example-based super-resolution. In the context of image super-resolution, KRR turned out to be easier to sparsify than support vector regression (SVR) and accordingly can potentially lead to more practical algorithms [6].[2]

Our algorithm achieves better results than these previous methods [11], [14], [28], and produces results which are comparable to [6]; our approach also dramatically enhanced applicability by two orders of magnitude faster training time. We also demonstrate that our GP regression-based algorithm outperforms Yang *et al*.'s sparse coding-based algorithm which is another commonly used regression algorithm [12].

2.2 JPEG/JPEG 2000 compression artifact removal

Block-based discrete cosine transform (BDCT) coding is widely used to compress still images and video sequences (e.g., JPEG/MPEG). However, at low bit rates, BDCT-encoded images can exhibit discontinuities at block boundaries, known as *block artifacts*. JPEG 2000 replaces the BDCT stage with a discrete wavelet transform. This prevents block artifacts, but *ringing artifacts* may still appear. Also, POCS has been successfully applied to JPEG [22] and JPEG 2000 [23] image enhancement.

One of the best established methods for block artifact removal is adaptive filtering with locally adjusted filter kernels to remove block edges while preserving image edges [31]. A similar technique has also been applied

2. As shown in [6], optimizing SVR hyper-parameters for image super-resolution leads to close to zero ϵ for ϵ-insensitive loss function of SVR. Accordingly, the corresponding optimal solution is *dense*.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2015.2389797, IEEE Transactions on Pattern Analysis and Machine Intelligence

to the removal of ringing artifacts in the context of trilateral filters [20]. Zhai *et al*. [19] proposed a block-shift filtering-based algorithm. For each pixel, the algorithm reconstructs a block encompassing that pixel based on a weighted combination of neighboring similar blocks. The overall result is a detail-preserving smoothing.

For learning approaches, Qiu [32] used a multi-layer Perceptron for JPEG deblocking while Lee *et al*. [33] proposed performing a piecewise linear regression in the space of DCT coefficients and showed comparable results to those of re-application of JPEG. Sun and Cham [2] proposed a *maximum a posteriori* (MAP) framework, building upon fields of experts [3], which led to improved performance over several existing methods including those methods based on POCS and overcomplete wavelet representations.

Finally, Nosratinia [15] proposed a surprising and promising non-learning-based method called *re-application of JPEG*. This algorithm generates a set of pixel-wise shifted versions of the input JPEG image, re-applies JPEG encoding to the shifted versions, and shifts them back to the original positions and averages. While simple, re-application of JPEG demonstrates superior performance to algorithms based on nonlinear filtering, POCS, and overcomplete wavelets. An application to JPEG 2000 enhancement is also feasible [16].

In our experiments, we demonstrate that by exploiting rich information from a database of images, our algorithm generates much better images in terms of quantitative criteria and visual quality than existing algorithms.

3 The Image Enhancement Algorithm

We wish to generate an enhanced image from a degraded image, with the aid of many example pairs of clean and degraded images. First, we describe how we build upon product of edge-perts (Edge-perts) and its prior [1] to learn the application- or dataset-specific conditional noise model. This learning relies upon Gaussian process regression to model the image enhancement process, and so second we motivate and explain the use of GP regression, specifically sparse GPs, and then explain our new efficient *semi-local* GP approximation.

3.1 Our approach

We start with the Edge-perts approach, which provides a MAP framework in the decorrelated wavelet domain:

$$\mathbf{z}^* = \arg\max_{\mathbf{z}} \left(\log p(\tilde{\mathbf{z}}|\mathbf{z}) + \log p(\mathbf{z})\right) \quad (1)$$

$$= \arg\min_{\mathbf{z}} \left(\frac{1}{2}\|\tilde{\mathbf{z}} - \mathbf{z}\|^2 + \sigma_P (\sum_j w_j [\mathbf{z}]_j^2)^{\alpha_P} \right), \quad (2)$$

where $\mathbf{x}$ is the latent clean image, $\tilde{\mathbf{x}}$ is the noisy input image, $\mathcal{W}[\cdot]$ is the wavelet transform, $\tilde{\mathbf{z}} = \mathcal{W}[\tilde{\mathbf{x}}]$, and σ_P is the user-specified regularization hyper-parameter. The experts model parameters $\{w_j\}$ and $\alpha_P \in [0, 1]$ are estimated by an *expectation maximization* algorithm [1].

A straightforward approach to apply the Edge-perts framework to general image enhancement problems is to modify the noise model accordingly:

$$p(\tilde{\mathbf{z}}|\mathbf{z}) \quad \propto \quad \left\| \tilde{\mathbf{z}} - \mathcal{W}\left[\mathcal{I}[\mathcal{W}^{\#}(\mathbf{z})]\right] \right\|^2, \quad (3)$$

where $\mathcal{I}[\cdot]$ is the degradation process of interest and $\mathcal{W}^{\#}(\mathbf{z})$ is the pre-image of $\mathbf{z}$. However, Edge-perts assumes only Gaussian noise; for general noise, such as in super-resolution applications, modeling the proper noise distribution might be difficult or cumbersome. Furthermore, general noise can be non-differentiable and even non-continuous which leads to optimization difficulties.

Instead of a parametric noise model, we learn a conditional noise model from examples. However, naively implementing this step requires optimizing Eq. (3) through learning the degradation process, which may be computationally infeasible. To keep computations tractable, we bypass this complex optimization by decoupling the conditional model learning and the optimization of Eq. (3): We introduce auxiliary *reference variables* $\mathbf{s}$ that summarize the outputs of the conditional model, and only indirectly model the degradation process by encoding the information contained in the training examples. This model is computationally favorable and does not require the user to model the noise explicitly.

We penalize the cost functional:

$$\mathcal{E}(\mathbf{z}) = \frac{1}{2}\|\mathbf{z} - \mathcal{W}[\mathbf{s}]\|^2 + \sigma_P (\sum_j w_j [\mathbf{z}]_j^2)^{\alpha_P}. \quad (4)$$

The *reference variable* matrix $\mathbf{s}$ is constructed through regression. For each pixel location i in the input degraded image $\tilde{\mathbf{x}}$, a Gaussian process (GP) regressor receives a degraded patch ($M \times M$) centered at i, and produces estimates of an enhanced patch ($N \times N$). GP regression predicts a Gaussian distribution: in our context, a mean patch and a predictive variance patch. When $N > 1$, the output patch overlaps its spatial neighbors, and so for each i in $\tilde{\mathbf{x}}$, the contributions from $N \times N$ patch regressions constitute a set of candidates means $\mathbf{f}_i$, and corresponding predictive variances $\mathbf{v}_i$. Then, $\mathbf{s}_i$ is a convex combination of all overlapping patch candidate means $\mathbf{f}_i \in \mathbb{R}^N$ and their *confidences* $\mathbf{c}_i$, where $\mathbf{s}_i = \mathbf{f}_i^\top \mathbf{c}_i$, with $[\mathbf{c}_i]_j \geq 0$ and $\|\mathbf{c}_i\|_{L_1} = 1$. The confidence vector $\mathbf{c}_i \in \mathbb{R}^N$ is calculated from the inverse of the *predictive variances* $\mathbf{v}_i \in \mathbb{R}^N$ of the candidates $\mathbf{f}_i$:

$$[\mathbf{c}_i]_j = \exp\left(-\frac{[\mathbf{v}_i]_j}{\sigma_C}\right) \Big/ \sum_{k=1,\ldots,N} \exp\left(-\frac{[\mathbf{v}_i]_k}{\sigma_C}\right), \quad (5)$$

where the scale parameter σ_C is fixed at 0.2. Minimizing Eq. 4 and transforming from the wavelet basis then generates the enhanced image (as per [1]). Algorithm 1 practically explains our framework in pseudocode.

Discussion. Generating several candidate means $\mathbf{f}_i$ and predictive variances $\mathbf{v}_i$ across neighboring patches aggregates information from a wider image area than from a single patch. Increasing patch size to aggregate

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2015.2389797, IEEE Transactions on Pattern Analysis and Machine Intelligence

Algorithm 1: Image Enhancement (Super-resolution)

Input : Low resolution input image L,
Training low/high res. image pairs $\mathcal{X}$, $\mathcal{Y}$.
Output : Enhanced image S.
Training: Build NN-search structure, e.g., KD-tree.

Testing :

Construct a bicubic upsampling B of L;

for *each pixel* p_j *in* B **do**
  Extract a patch $\mathbf{x}_j$ (centered at p_j);
  Use semi-local GP regression to predict distribution $p(f_j|\mathcal{Y})$ on output patch $\mathbf{y}_j$:
  1) Retrieve n-NNs ($\mathcal{C}_n$) of $\mathbf{x}_j$ from $\mathcal{X}$;
  2) Predict $\mathbb{E}[f(\mathbf{x}_j)]$ and $\mathbb{V}[f(\mathbf{x}_j)]$ from the labels of $\mathcal{C}_n$ using Eq. (19);
end
Re-arrange $\{\mathbb{E}[f(\mathbf{x}_j)]\}$ and $\{\mathbb{V}[f(\mathbf{x}_j)]\}$ for overlapping patches so that for each pixel p_i obtains sets of candidate means $\{\mathbf{f}_i\}$ and corresponding variances $\{\mathbf{v}_i\}$;

Build reference variable matrix $\mathbf{s}$ from $\{\mathbf{f}_i\}$ and $\{\mathbf{v}_i\}$;

Minimize Eq. 4 and to find the enhanced image S.

more information is possible, but in preliminary experiments we found that the difficulty in learning in higher-dimensional spaces resulted in worse prediction than our patches (e.g., 7×7 for $2\times$ magnification super-resolution). Compared to single-pixel-output regression, patch-valued regression produced 0.6dB PSNR increase in super-resolution experiments (as per [6]).

Unlike elegant fully Bayesian image models (e.g., [4]), an intuitive probabilistic interpretation of the image enhancement process is no longer possible: the predictive distribution of a GP is a *posterior* distribution that is conditioned on the degraded input patches. Since this conditional model is not generative, sampling a degraded image is not directly possible. This functionality is not required for image enhancement applications. Using the local posterior distribution as a surrogate noise model has been well established in speech recognition [34] and similar strategies have been recently used in image enhancement as well [5], [28]. In general, Eq. 4 could be regarded as a regularization framework which trades between regularity enforcement in the wavelet domain and deviation from the reference variables $\{\mathbf{s}_i\}$.

3.2 Regression

The underlying idea for using GP regression is its powerful generalization capability. For instance, in classical example-based super-resolution applications, a typical choice for the conditional model is NN-based estimation [14], [26], [35]. However, from the general regression perspective, NN regression can be improved since it *overfits* to the data, i.e., we obtain a function which fits the training data perfectly but cannot generalize to new data. GP regression has powerful regularization capability which avoids overfitting and leads to a better generalization than NN regression (See Fig. 1, Sec. 4.1, and supplementary material for comparison with NN approaches [11], [14]). GP regression is much more flexible than simple linear regression, which fails when the problem is highly non-linear (Fig. 1). Furthermore, in our super-resolution experiments, we demonstrate that our algorithm outperforms Yang *et al.*'s sparse coding regression choice [12].

We review basic GP regression in Section 3.2.1, before reviewing more efficient sparse GP approximations which requires optimizing *inducing variables* (Section 3.2.2). Section 3.2.3 introduces our new GP approximation which bypasses completely the inducing variables optimization and therefore facilitates fast training.

3.2.1 Basic Gaussian process regression

Suppose a set of data points (in our case, vectorized patches) $\mathcal{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_l\} \subset \mathbb{R}^{M^2}$ and their corresponding labels $\mathcal{Y} = \{\mathbf{y}_1, \dots, \mathbf{y}_l\} \subset \mathbb{R}^{N^2}$. We adopt a Gaussian noise model with mean $\mathbf{0}$ and the covariance matrix $\sigma^2\mathbf{I}$:

$$\mathbf{y}_i = f(\mathbf{x}_i) + \epsilon, \text{ where } \epsilon \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I}), \tag{6}$$

where $\mathcal{N}(\mu, \mathbf{\Sigma})$ is the probability density of the Gaussian random variable with mean μ and covariance $\mathbf{\Sigma}$, and $f : \mathbb{R}^{M^2} \mapsto \mathbb{R}^{N^2}$ is the underlying *latent* function. Then, a zero-mean Gaussian process (GP) prior is placed over f, which for a given set of test points $\mathcal{X}_* = \{\mathbf{x}_{*(1)}, \dots, \mathbf{x}_{*(l')}\}$ is realized as [36]:[3]

$$p(\mathbf{f}_*, \mathbf{f}) = \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} \mathbf{K}_{\mathbf{f},\mathbf{f}} & \mathbf{K}_{\mathbf{f},*} \\ \mathbf{K}_{*,\mathbf{f}} & \mathbf{K}_{*,*} \end{bmatrix}\right), \tag{7}$$

where the subscripts $\mathbf{f}$ and $*$ represent indexing across training and testing data points, respectively (e.g., $\mathbf{f} = [f(\mathbf{x}_1), \dots, f(\mathbf{x}_l)]^\top$, $\mathbf{f}_* = [f(\mathbf{x}_{*(1)}), \dots, f(\mathbf{x}_{*(l')})]^\top$, and $[(\mathbf{K}_{*,\mathbf{f}})_{(i,j)}]_{l',l} = k(\mathbf{x}_{*(i)}, \mathbf{x}_j)$). While any positive definite function can be used as the *covariance* function k, we adopt the standard Gaussian kernel:

$$k(\mathbf{x}, \mathbf{y}) = \exp\left(-b\|\mathbf{x} - \mathbf{y}\|^2\right).$$

Combining (6) and (7), the joint distribution of $p(\mathbf{f}_*, \mathcal{Y})$:

$$p(\mathbf{f}_*, \mathcal{Y}) = \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} \mathbf{K}_{\mathbf{f},\mathbf{f}} + \sigma^2\mathbf{I} & \mathbf{K}_{\mathbf{f},*} \\ \mathbf{K}_{*,\mathbf{f}} & \mathbf{K}_{*,*} \end{bmatrix}\right), \tag{8}$$

from which, the predictive distribution (for the j-th output dimension; index j in $\mathbf{f}_*$ is omitted) is constructed by conditioning $\mathbf{f}_*$ on the labels $\mathcal{Y}$:

$$\begin{aligned} p(\mathbf{f}_*|\mathcal{Y}) = &\mathcal{N}\big(\mathbf{K}_{*,\mathbf{f}}(\mathbf{K}_{\mathbf{f},\mathbf{f}} + \sigma^2\mathbf{I})^{-1}\mathbf{Y}_{[:,j]}, \\ &\mathbf{K}_{*,*} - \mathbf{K}_{*,\mathbf{f}}(\mathbf{K}_{\mathbf{f},\mathbf{f}} + \sigma^2\mathbf{I})^{-1}\mathbf{K}_{\mathbf{f},*}\big), \end{aligned} \tag{9}$$

where $\mathbf{Y} = [\mathbf{y}_1^\top, \dots, \mathbf{y}_l^\top]^\top$ and $\mathbf{A}_{[:,j]}$ is the j-th column of the matrix $\mathbf{A}$. For the simplicity of exposition, we adopt a

3. For computational convenience, we treat each output independently and identically. For notational convenience, we omit conditioning on input variables.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2015.2389797, IEEE Transactions on Pattern Analysis and Machine Intelligence

slight misuse of notations and summarize the predictive distribution (9) across every output dimensions as:

$$p(\mathbf{f}_*|\mathcal{Y}) = \mathcal{N}\big(\mathbf{K}_{*,\mathbf{f}}(\mathbf{K}_{\mathbf{f},\mathbf{f}}+\sigma^2\mathbf{I})^{-1}\mathbf{Y}, \mathbf{K}_{*,*}-\mathbf{K}_{*,\mathbf{f}}(\mathbf{K}_{\mathbf{f},\mathbf{f}}+\sigma^2\mathbf{I})^{-1}\mathbf{K}_{\mathbf{f},*}\big), \quad (10)$$

where each output dimension shares the covariance.

The diagonal terms of the covariance matrix (predictive variances) in the predictive distribution (9) represent the uncertainty of regression output, while off-diagonal terms represent the dependency between variables.

While GP regression has been shown to be competitive on a wide range of small-scale applications, its application to large-scale problems is limited due to its unfavorable scaling behavior: The training (i.e., the calculation of $\mathbf{K}_{\mathbf{f},\mathbf{f}}$ and the corresponding regularized inversion) takes $O(Ml^2 + l^3)$ time, while for a given test point, testing time complexity is $O(Ml + lN)$ and $O(Ml + l^2)$ for the mean and the predictive variance, respectively (see [36] for details).

3.2.2 Sparse Gaussian processes

A standard approximate approach to overcome the unfavorable scaling behavior of GPs is to introduce a small set of *inducing variables* $\mathbf{f}_\mathcal{U} = \{f(\mathbf{u}_1), \ldots, f(\mathbf{u}_m)\}$ (corresponding to *inducing inputs* $\mathcal{U} = \{\mathbf{u}_1, \ldots, \mathbf{u}_m\}$) through which the conditional independence of $\mathbf{f}_*$ and $\mathbf{f}$ is assumed in the approximation of the joint prior (see the unified framework of [10]):

$$p(\mathbf{f}_*, \mathbf{f}) \approx q(\mathbf{f}_*, \mathbf{f}) = \int q(\mathbf{f}_*|\mathbf{f}_\mathcal{U}) q(\mathbf{f}|\mathbf{f}_\mathcal{U}) p(\mathbf{f}_\mathcal{U}) d\mathbf{f}_\mathcal{U}. \quad (11)$$

The *training conditional* $q(\mathbf{f}|\mathbf{f}_\mathcal{U})$ is approximated subsequently. This leads to a set of approximations which are referred to as *sparse* GPs where the inference is carried out through $\mathcal{U}$ summarizing the entire training set $\mathcal{X}$ [8], [9], [37]. For instance, Seeger *et al*. [37] proposed an approximate prior:

$$q(\mathbf{f}_*, \mathbf{f}) = \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} \mathbf{Q}^\mathcal{U}_{\mathbf{f},\mathbf{f}} & \mathbf{Q}^\mathcal{U}_{\mathbf{f},*} \\ \mathbf{Q}^\mathcal{U}_{*,\mathbf{f}} & \mathbf{K}_{*,*} \end{bmatrix}\right), \quad (12)$$

where $\mathbf{Q}^\mathcal{U}_{\mathbf{r},\mathbf{s}} = \mathbf{K}_{\mathbf{r},\mathbf{u}}\mathbf{K}^{-1}_{\mathbf{u},\mathbf{u}}\mathbf{K}_{\mathbf{u},\mathbf{s}}$ and $\mathbf{u}$ represents indexing across $\mathcal{U}$. The corresponding predictive distribution is:

$$q(\mathbf{f}_*|\mathcal{Y}) = \mathcal{N}\big(\mathbf{Q}^\mathcal{U}_{*,\mathbf{f}}(\mathbf{Q}^\mathcal{U}_{\mathbf{f},\mathbf{f}}+\sigma^2\mathbf{I})^{-1}\mathbf{Y}, \mathbf{K}_{*,*}-\mathbf{Q}^\mathcal{U}_{*,\mathbf{f}}(\mathbf{Q}^\mathcal{U}_{\mathbf{f},\mathbf{f}}+\sigma^2\mathbf{I})^{-1}\mathbf{Q}^\mathcal{U}_{\mathbf{f},*}\big). \quad (13)$$

With this prior, the predictive mean is obtained as a linear combination of evaluations of m basis functions $\{k(\mathbf{u}_1, \cdot), \ldots, k(\mathbf{u}_m, \cdot)\}$ (explaining the name *sparse GPs*). The time complexity of calculating the predictive distribution becomes $O(Mlm + lm^2)$ off-line plus $O(Mm + mN + m^2)$ per test point.

Once we identify the inducing inputs $\mathcal{U}$, they are fixed throughout the entire test set. The problem is then cast into an optimization where one constructs $\mathcal{U}$ based on a certain measure of approximation quality (e.g., marginal likelihood and information gain; see [8], [9], [36], [37] for more examples and details). The performance of a sparse approximation depends heavily on the inducing inputs $\mathcal{U}$. However, usually the corresponding optimization problem is non-convex and so requires a non-linear optimization which is not easy to solve.

3.2.3 Semi-local approximation of Gaussian processes

Our approach is fundamentally different from existing algorithms, and bypasses non-linear optimization through inducing inputs $\mathcal{U}$ using a simple heuristic: Instead of sharing $\mathcal{U}$ for all test data points, we simply approximate the inducing variables as the nearest neighbors (NNs) of each test input. As such, we build a specially tailored sparse GP for each test input $\mathbf{x}_*$, i.e., $\mathcal{U} \equiv \mathcal{U}_*$ is chosen depending on $\mathbf{x}_*$ — the corresponding GP model is constructed only when it is presented with a test point $\mathbf{x}_*$. An important advantage of this *on-line* approach is that, in general, it enables us to build more flexible approximations than existing *off-line* approaches.

Furthermore, it leads to an extremely simple but powerful strategy for identifying $\mathcal{U}_*$: If we introduce a Markov assumption on $\{f_*, \mathbf{f}\}, (f_* \equiv f(\mathbf{x}_*))$:

$$p(f_*|\mathbf{f}, \mathcal{B}(f_*)) \approx q(f_*|\mathcal{B}(f_*)), \quad (14)$$

where $\mathcal{B}(f_*)$ denotes the values of f for inputs in the domain neighborhoods $\mathcal{B}(\mathbf{x}_*) \subset \mathcal{X}$ of $\mathbf{x}_*$, then the corresponding conditional independence of f_* and $\mathbf{f}$ given $\mathcal{B}(f_*)$ makes $\mathcal{B}(\mathbf{x}_*)$ a valid candidate for $\mathcal{U}_*$, i.e., the approximation (11) becomes exact once we use $\mathcal{B}(\mathbf{x}_*)$ for $\mathcal{U}_*$. Now, applying Eq. 14 to Eq. 13 we obtain $\mathbf{Q}^\mathcal{U}_{\mathbf{r},\mathbf{s}}$ specified as $\mathbf{Q}^\mathcal{B}_{\mathbf{r},\mathbf{s}} = \mathbf{K}_{\mathbf{r},\mathbf{b}}\mathbf{K}^{-1}_{\mathbf{b},\mathbf{b}}\mathbf{K}_{\mathbf{u},\mathbf{s}}$ with $\mathbf{b}$ indexing across $\mathcal{B}(\mathbf{x}_*)$:

$$\begin{aligned} q(f_*|\mathcal{Y}) =& \mathcal{N}\big(\mathbf{Q}^\mathcal{B}_{*,\mathbf{f}}(\mathbf{Q}^\mathcal{B}_{\mathbf{f},\mathbf{f}}+\sigma^2\mathbf{I})^{-1}\mathbf{Y}, \\ & \mathbf{K}_{*,*}-\mathbf{Q}^\mathcal{B}_{*,\mathbf{f}}(\mathbf{Q}^\mathcal{B}_{\mathbf{f},\mathbf{f}}+\sigma^2\mathbf{I})^{-1}\mathbf{Q}^\mathcal{B}_{\mathbf{f},*}\big) \\ =& \mathcal{N}\big(\sigma^{-2}\mathbf{K}_{*,\mathbf{b}}\Sigma^\mathcal{B}\mathbf{K}_{\mathbf{b},\mathbf{f}}\mathbf{Y}, \\ & \mathbf{K}_{*,*}-\mathbf{Q}^\mathcal{B}_{*,*}+\mathbf{K}_{*,\mathbf{b}}\Sigma^\mathcal{B}\mathbf{K}_{\mathbf{b},*}\big), \end{aligned} \quad (15)$$

where $\Sigma^\mathcal{B} = \big(\sigma^{-2}\mathbf{K}_{\mathbf{b},\mathbf{f}}\mathbf{K}_{\mathbf{f},\mathbf{b}} + \mathbf{K}_{\mathbf{b},\mathbf{b}}\big)^{-1}$ and the second equality is obtained by applying *Sherman-Morrison-Woodbury* formula.

This new approximation dramatically reduces the computation time during training as we only need to build a data structure for NN-search to identify $\mathcal{B}(f_*)$. During prediction, it takes $O(Mlm + lm^2)$ time for each test point with $m = |\mathcal{B}(f_*)|$, which includes the time spent building a model. However, for large l ($\approx 2 * 10^5$ in current applications), this might *still* be impractical.

Hence, the second step of our approximation is obtained by introducing an additional Markov-like assumption on the observations $\mathcal{Y}$:

$$p(f_*|\mathcal{Y}, \mathcal{B}_1(\mathbf{y}_*)) \approx q(f_*|\mathcal{B}_1(\mathbf{y}_*)), \quad (16)$$

where $\mathcal{B}_1(\mathbf{y}_*)$ denotes the observed training target values in the domain neighborhood $\mathcal{B}_1(\mathbf{x}_*)$ of $\mathbf{x}_*$. In general, neither (14) nor (16) is stronger than the other since neither is implied by the other — only when the noise

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2015.2389797, IEEE Transactions on Pattern Analysis and Machine Intelligence

level is zero do the two assumptions become equivalent. However, practically, (16) can be regarded as a stronger assumption than (14), since (16) implies that given $\mathcal{B}_1(\mathbf{y}_*)$, all the remaining training data points are irrelevant in making predictions of f_*, which is not the case for (14). Accordingly, we set $\mathcal{B}_1(\mathbf{x}_*)$ much wider than $\mathcal{B}(\mathbf{x}_*)$ (i.e., $\mathcal{B}_1(\mathbf{x}_*) \supset \mathcal{B}(\mathbf{x}_*)$). This guarantees that the resulting GPs are non-locally regularized. In Sec. 5, for the interested reader, we discuss the motivation behind this second approximation step, which is based on analysis of the large-scale behavior of full GPs.

In practice, we choose m and n-nearest neighbors $\mathcal{C}_n(\mathbf{x}_*)(\mathcal{C}_m(\mathbf{x}_*) \subset \mathcal{C}_n(\mathbf{x}_*) \subset \mathcal{X})$ of $\mathbf{x}_*$, instead of $\mathcal{B}(f_*)$ and $\mathcal{B}_1(\mathbf{y}_*)$, such that prediction is performed based on n data points summarized by m inducing inputs. Now, applying Eq. 16 to Eq. 15 in this context, we obtain:

$$q(f_*|\mathcal{Y}_\mathcal{C}) = \mathcal{N}\big(\sigma^{-2}\mathbf{K}_{*,\mathbf{b}}\Sigma_\mathcal{C}^\mathcal{B}\mathbf{K}_{\mathbf{b},\mathbf{c}}\mathbf{Y}_\mathcal{C}),\\ \mathbf{K}_{*,*} - \mathbf{Q}_{*,*}^\mathcal{B} + \mathbf{K}_{*,\mathbf{b}}\Sigma_\mathcal{C}^\mathcal{B}\mathbf{K}_{\mathbf{b},*}\big), \qquad (17)$$

where in a compact notation $\mathcal{Y}_\mathcal{C}$ and $\mathbf{Y}_\mathcal{C}$ represent the subset of $\mathcal{Y}$ and the rows of $\mathbf{Y}$, respectively corresponding to the elements of $\mathcal{C}_n(\mathbf{x}_*) \subset \mathcal{X}$, $\Sigma_\mathcal{C}^\mathcal{B} = \big(\sigma^{-2}\mathbf{K}_{\mathbf{b},\mathbf{c}}\mathbf{K}_{\mathbf{c},\mathbf{b}} + \mathbf{K}_{\mathbf{b},\mathbf{b}}\big)^{-1}$, and $\mathbf{c}$ represents indexing across $\mathcal{C}_n(\mathbf{x}_*)$. The sizes of the neighborhoods m and n are decided based on prescribed computational complexity requirement (Sec. 4).

We refer to this new approximation as *semi-local GP*. With the same number of inducing inputs, this input-dependent selection should, in general, provide more flexibility than standard sparse methods because the inducing variables are specifically tailored to each test data point. As shown in Fig. 1, for smaller numbers of inducing inputs our semi-local GPs perform especially better than sparse methods, which use relatively large numbers of carefully chosen inducing inputs.

Furthermore, given hyper-parameters, the only training component for semi-local GPs is to build a data structure for NN-search, and so off-line processing is very fast. Therefore, the algorithm is very flexible as the system can be easily adapted to the distribution of a specific (non-generic) class of images (see Sec. 4.1).

Discussion. The Markov model used in the first step of approximation (Eq. 14) has proven to be effective in many different applications. However, we exploit the Markov assumptions only in the approximation of the prior through its factorization and the marginalization over $\mathbf{f}_\mathcal{U}$ in Eq. 11. Accordingly, although the resulting sparse model can represent only local variations around $\mathbf{x}_*$, the corresponding prediction takes into account the entire data set through the dependency between $\mathbf{f}$ and $\mathbf{f}_{\mathcal{U}_*}$ (see Eq. 11). This implies that for each test variable f_*, the corresponding joint distribution $q(f_*, \mathbf{f})$ fits into the approximation (Eq. 12) and it is a valid probabilistic approximation of the full GP. In particular, no overfitting occurs since the model is globally regularized. This is in contrast to the *moving least-squares* algorithm which is not directly related to any global regularization.

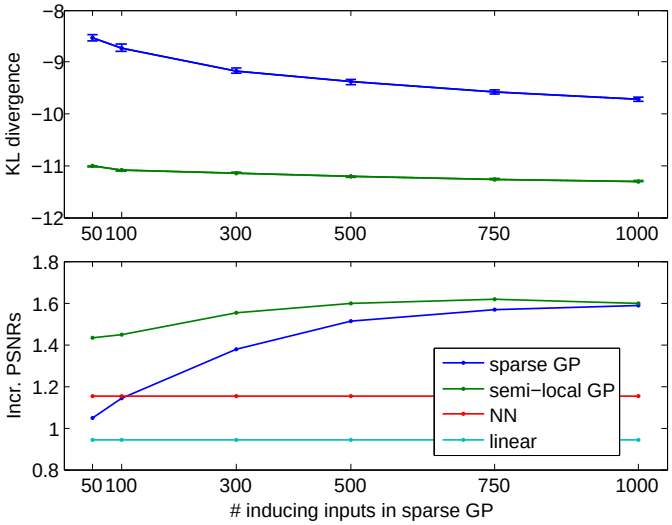


Fig. 1: Approximation accuracies of sparse GPs and semi-local GPs wrt. full GP for the super-resolution experiment data points. *Top:* The Kullback-Leibler divergences from full GPs of the predictive distributions of approximate GPs, plotted against m' sparse GP inducing inputs. To compare, we use only $20,000$ data points to train all models, sampled from a large training set of 200,000. For each m', a semi-local GP was trained, with number of inducing inputs m and local training data points n such that the time complexity of test point prediction roughly matches $((m')^2 \approx m^2 n)$. Experiments were repeated ten times with randomly selected training data sets. Error bar lengths are $2\times$ std. dev. *Bottom:* Average PSNR increase from bicubic resampling, measured from final super-resolution results. For comparison, we replace our regression by linear regression and NN regression. The inducing inputs were optimized by maximizing the *marginal likelihood* $p(\mathbf{f}_*|\mathbf{u})$ [8].

Theoretically, one drawback of our on-line model is that it does not correspond to any *consistent* global GP framework: In our model, the prior is defined through the inducing variables (Eq. 11) that depend on each test input. Since, in general, the Gaussian property of marginal distributions $q(f_*, \mathbf{f})$ does not imply the Gaussian property of the corresponding joint distribution, it may not be possible to construct a Gaussian joint prior $q(\mathbf{f}_*, \mathbf{f})$ over the entire set of potential testing points $\mathbf{f}_*$.

A direct consequence of this inconsistency is that, since there is no globally defined covariance function, no prediction can be made for off-diagonal elements of the predictive covariance, which represents the dependency between predictions.[4] However, this is not a major concern as we only use the means and variances of individual predictions $\{p(f_*|\mathcal{Y})\}$, which are valid probability distributions (i.e., $p(f_*|\mathcal{Y})$ is a fully Bayesian prediction).

4. Snelson and Ghahramani proposed an inconsistent GP approximation [38] where the training data are pre-partitioned during training into a set of clusters which constitute input-dependent inducing inputs.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2015.2389797, IEEE Transactions on Pattern Analysis and Machine Intelligence

3.3 Noise model

There are two sources of uncertainty in making predictions with GPs [39]. One is the fact that, in general, the test input may deviate from the training inputs ($U1$). The other is the *noise* in the data ($U2$): Due to the ill-posed nature of the problem, even if the test input $\mathbf{x}_*$ exactly matches one of the training inputs (say $\mathbf{x}_i$), the corresponding training output $\mathbf{y}_i$ might not be the underlying ground truth output $f(\mathbf{x}_*)$. In the current context of GP regression, $U1$ and $U2$ uncertainties are independently modeled with the noise parameter (σ^2) and the kernel parameter (b), respectively. This clear separation is due to the use of an *i.i.d.* Gaussian noise model (Eq. 6) which is important because it leads to an analytical model for predictive distribution (Eq. 9).

In general, the noise ($U2$) is correlated and depends on the input (and so on $U1$). However, sophisticated noise models which reflect this dependency may lead to non-analytic predictive distributions and so are not computationally favorable for image enhancement applications.

We present a simple noise model which exploits the dependency between these two sources of uncertainty. Our scheme considers the empirically-observed correlation between two quantities which are related to $U1$ and $U2$. The first quantity $P1$ is the average distance from a test input to its nearest training inputs, which represents $U1$. The second quantity $P2$ is the average distance among the corresponding retrieved training outputs. This is not directly related to $U2$; however, when $P1$ is zero, $P2$ should correspond to the standard deviation of the output conditioned on an input and, accordingly, be an empirical estimate of the noise level $U2$.

By construction, $P1$ and $P2$ are mildly correlated: If $P1$ is small, the corresponding training inputs should be close to each other and so $P2$ tends to be small. However, Fig. 2 shows there are cases where the correlation is much stronger: when $P1$ is very small, $P1$ and $P2$ are especially strongly correlated (e.g., $P1 < 0.005$, when the test input is very close to some training inputs). As $P1$ increases, the correlation becomes weaker and eventually disappears. This observation led us to conjecture that $U2$ is correlated to $U1$ especially when $U1$ is small.

We validate this conjecture by implementing it into our noise model. Our semi-local GP model is adaptive in the sense that the model itself depends on each test input. Naturally, the noise parameter σ^2 can also be adapted to each test input $\mathbf{x}_*$ (and its distances to the stored training inputs). For computational efficiency, we still use a Gaussian noise model but adapt it to the local density at the point of evaluation. Eq. 17 then becomes:

$$q(f_*|\mathcal{Y}_\mathcal{C}) = \mathcal{N}\big(\mathbf{K}_{*,\mathbf{b}}(\Sigma_\mathcal{C}^\mathcal{B})'\mathbf{K}_{\mathbf{b},\mathbf{c}}\Gamma\mathbf{Y}_\mathcal{C}, \mathbf{K}_{*,*} - \mathbf{Q}_{*,*}^\mathcal{B} + \mathbf{K}_{*,\mathbf{b}}(\Sigma_\mathcal{C}^\mathcal{B})'\mathbf{K}_{\mathbf{b},*}\big), \quad (18)$$

where

$$(\Sigma_\mathcal{C}^\mathcal{B})' = (\mathbf{K}_{\mathbf{b},\mathbf{c}}\Gamma\mathbf{K}_{\mathbf{c},\mathbf{b}} + \mathbf{K}_{\mathbf{b},\mathbf{b}})^{-1}, \quad (19)$$

$$\Gamma = \text{diag}\left[N_c \exp\left(-N_d b\mathbf{d}\right)\right] + \sigma^{-2}\mathbf{I}, \quad (20)$$

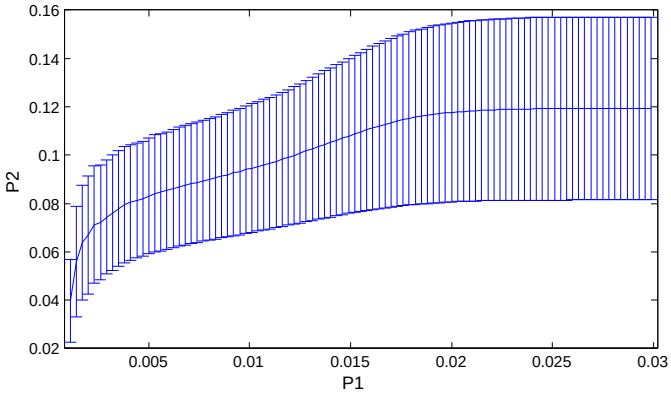


Fig. 2: Variation of $P2$ as a function $P1$ in super-resolution experiments (see text for $P1$ and $P2$ descriptions): For each test input, we select 10 nearest neighbors and calculate the corresponding $P1$ and $P2$ values.

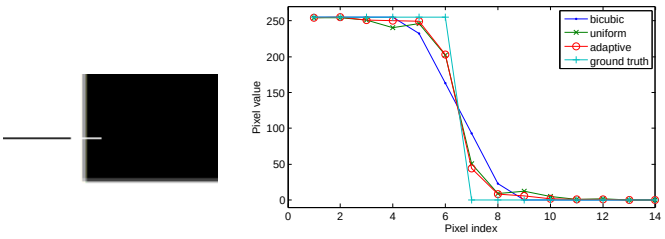
$\mathbf{d}$ is a vector containing the squared distances between $\mathbf{x}_*$ and the elements of $\mathcal{C}_n(\mathbf{x}_*)$, N_c and N_d are the hyper-parameters, and diag$[\cdot]$ is an operator which converts a vector into a diagonal matrix.

From the regularization perspective, the noise variance σ^2 is the parameter controlling the contribution of training error and the global regularization term. An intuitive explanation of our noise model is that when the given input is sufficiently close to the training data, we rely more on data than on regularization. Furthermore, within the set of training data points, we emphasize more the data points which are closer to the test input. The computational complexity of this model is the same as that of the uniform noise model (17). However, this model resulted in on average 0.08 improvement of PSNR values in our super-resolution experiments.

Discussion. We observe that most *easy* test points with small $P1$ values lie at major edges (see supplementary material for examples). Typically, the major edges show clean and strong change of pixel values and do not contain complex textures. Intuitively, for those patterns, the noise level must be low, i.e., the desired output should be less uncertain given the input. This explains a strong correlation between $P1$ and $P2$ for small $P1$ values. The role of our adaptive noise model (18) is then to regularize less for those patterns lying at major edges.

A visually noticeable consequence is that ringing artifacts are significantly reduced. Typically the results of regularized regression show a certain fluctuation when there is an abrupt and significant change of the signal to compensate the resulting loss of smoothness. By placing more emphasis on observed data than the regularizer, we can effectively suppress these regularization artifacts which appear as ringing artifacts (Figure 3). Kim and Kwon [6] adopted a post-processing step to explicitly remove the ringing artifact, which is not necessary when we use an adaptive noise model.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2015.2389797, IEEE Transactions on Pattern Analysis and Machine Intelligence



(a) (b)

Fig. 3: An example of a major edge: (a) an image showing a strong edge (bicubic-resampled image is shown) and (b) cross-sections of the super-resolution results at the locations marked with a white horizontal bar in (a). In the result of our algorithm that uses the homogeneous noise model (denoted as 'uniform') a fluctuation occurred at the vicinity of the edge (pixel indices 4 and 9). This ringing is suppressed with the adaptive noise model.

4 Results

This section demonstrates two applications: super-resolution (Sec. 4.1) and JPEG artifact removal (Sec. 4.2). In the supplementary material, we demonstrate JPEG 2000 artifact removal. To evaluate performance, we used two databases (supplemental material Fig. 1): DB1, containing sixteen images familiar to the community (512×512 or 256×256), and DB2, containing 500 images from a personal photo collection (512×512). For training, $200,000$ data points are sampled for GP regressions. For quantitative evaluation, each clean image was degraded, e.g., JPEG encoding for JPEG artifact removal, and blurring plus sub-sampling for super-resolution. The enhanced images were compared with the original images via the increase in both peak signal-to-noise ratio (PSNR) and perceptually-based structural similarity (SSIM) [24]. On a 3.6GHz machine in MATLAB, training takes 5 minutes (degrading images, sampling training data, and building an NN-search tree), with JPEG enhancement taking three minutes and super-resolution taking two minutes (see supplemental material for run-times of competing algorithms). We use the YIQ space for color images, with enhancement performed only on Y.

Parameters are the input and output patch sizes (M, N), the regularization parameter (σ_P), and the hyper-parameters for regression: the noise parameters σ^2, N_c, and N_d, the kernel parameter b, and the numbers of training and inducing inputs n and m (Table 1). m and n were fixed at 50 and 200, determined by trading performance against computational complexity: performance increased steadily as m and n increase, while run-time grew roughly quadratically and linearly with respect to m and n, respectively. Our experiments suggested that optimal values of other parameters varied depending on the application, except for N_c, N_d (Eq. 19), and N. As such, these values were fixed by a set of validation images which were disjoint from training and testing images. The remaining parameters were optimized for each application based on a set of validation images.

TABLE 1: Parameters used in the experiments

Expr. idx.	M	σ^2	b	σ_P	N_c	N_d	N	m	n
JPEG	7	$5 * 10^{-3}$	10	2.0	10	20	5	50	200
Super-res.	7	$5 * 10^{-8}$	20	0.5	10	20	5	50	200

4.1 Single-image super-resolution

The input low-resolution image is enlarged to the target scale by bicubic resampling, which is subsequently band-frequency filtered based on the Laplacian of Gaussian (LOG) filter ([6], [14]). Given a LOG filtered image patch, the regressor estimates a patch containing the difference between the input and the hypothesized ground truth such that the output candidates $\{\mathbf{f}_i\}$ are obtained by adding the regression result to the input image. Before the regression step, each pair of input and output patches is contrast normalized by the L^1 norm of the input patch, and the regression output is inverse normalized. For edge cases, input images were extended by symmetrically replicating pixel values across the image boundary. We magnify by 2 along each image dimension (for other factors, see the supplementary material).

Figures 4-5 show the results of super-resolution. For comparison, we display the results of Chang *et al.*'s algorithm [11], Kim and Kwon's algorithm [6], Freeman *et al.*'s algorithm [14], and He and Siu's non-example-based algorithm [13]. A comparison with Yang *et al.*'s algorithm [12] is provided in the supplementary material.

All tested super-resolution algorithms outperformed the bicubic resampling baseline method. However, Freeman *et al.*'s results are noisy (lighthouse and astronauts, Fig. 4c). Chang *et al.*'s results are less noisy but more blurry than the results of [14] and [12]. He and Siu's algorithm [13] coherently restored sharp edges; however, it tends to smooth texture details and sometimes over-sharpen edges (astronauts and woman, Fig. 4d). For general images, the results of our approach and [6] are equally good except for the lighthouse image where, thanks to our adaptive noise model, we suppress slight ringing artifacts (Fig. 4). Overall, both algorithms are as sharp as but less noisy than both [14] and [12], and are more detailed than [13]. This is confirmed through PSNR and SSIM measures: Our algorithm gives better PSNR and worse SSIM values than [6].

While Kim and Kwon's algorithm already outperforms several state-of-the-art algorithms [6], there is an important limitation which we explicitly overcome: to achieve the high-level of accuracy and reasonable execution time, Kim and Kwon's algorithm requires 36 hours training time while our algorithm only takes 5 minutes for training. This difference is important especially when *a priori* knowledge is available in terms of a class-specific set of example images, e.g., a face has distinct statistical properties from a document. We can quickly generate class-specific examples on which to train, which leads to much better results as shown in Fig. 5. This is infeasible in [6] due to its high complexity in training. The

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2015.2389797, IEEE Transactions on Pattern Analysis and Machine Intelligence

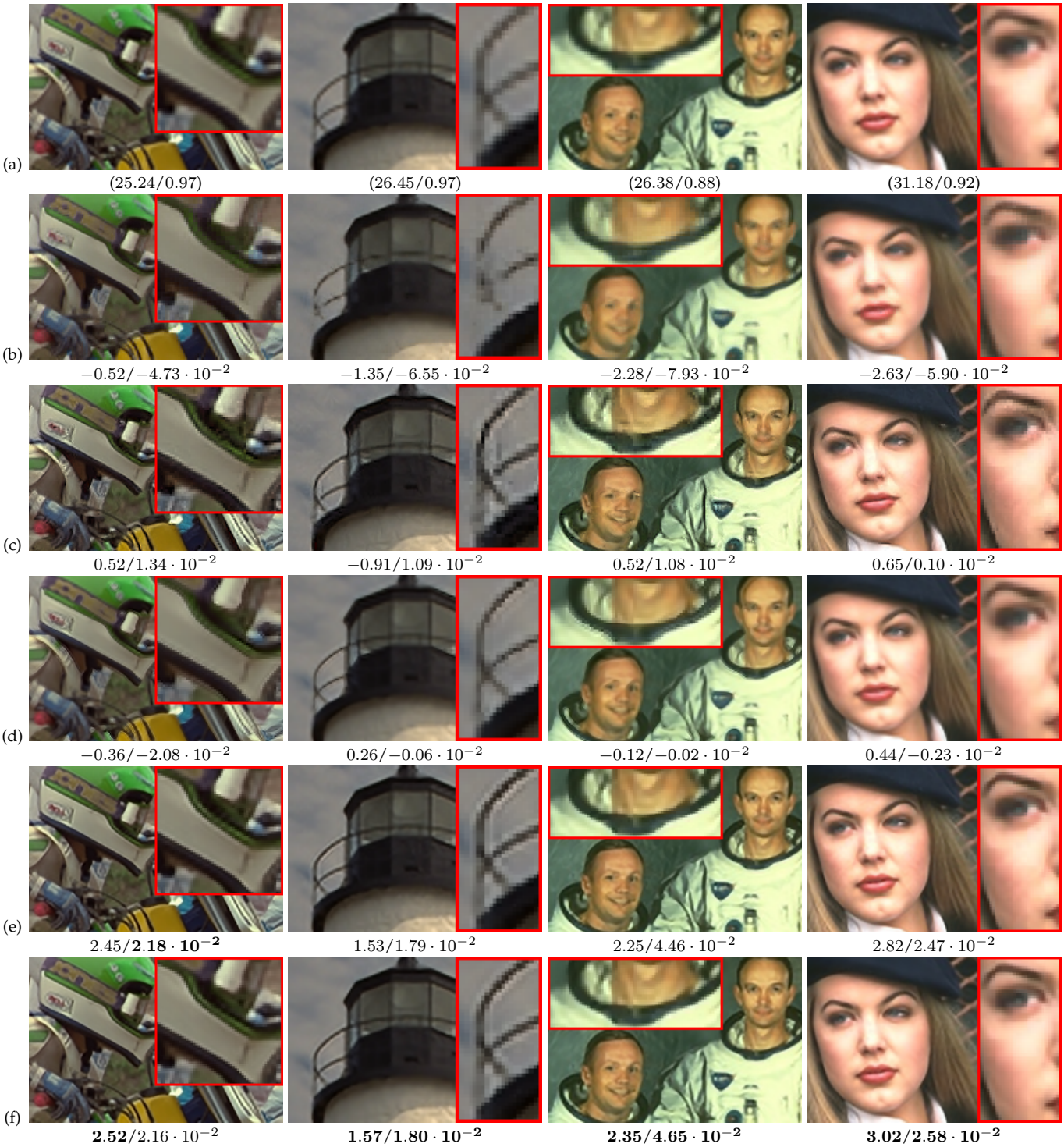


Fig. 4: Examples of image super-resolution — please zoom into the electronic version! (a) Bicubic resampling, (b) Chang *et al*. [11], (c) Freeman *et al*. [14], (d) He and Siu [13], (e) Kim and Kwon [6], and (f) our method. Increases of PSNRs (in dB) and SSIMs with respect to the input bicubic resampled images (displayed below each column) were calculated based on the complete images. For the input images (a), the original PSNR and SSIM values are shown. The best results are marked with bold letters. See supplemental material Figure 17 for uncropped originals. Note noise in (c), blur in (b), smoothed texture details (woman; fourth column) and over-sharpen edges (astronauts; third column) in (d). (e) and (f) are similar, but (f) shows fewer ringing artifacts (lighthouse; second column).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2015.2389797, IEEE Transactions on Pattern Analysis and Machine Intelligence

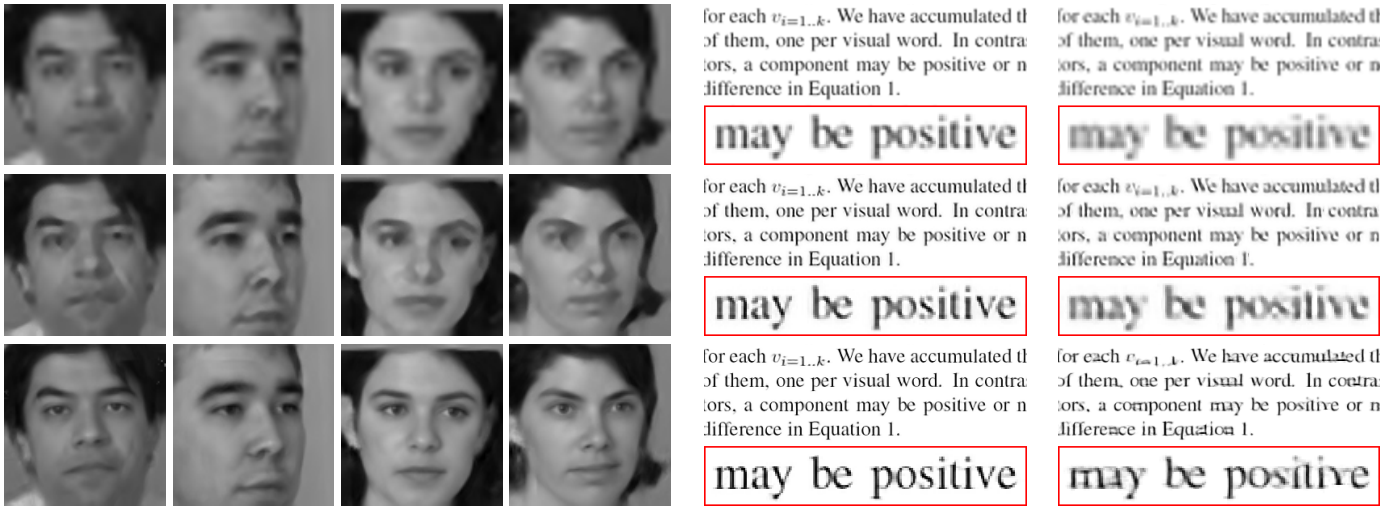


Fig. 5: *Top:* Bicubic resampling. *Middle:* Kim and Kwon [6]. *Bottom:* Our method with dataset-specific DBs. *Left:* Face-specific super-resolution: [6] ΔPSNR: 1.18dB, ΔSSIM: 0.032; our method ΔPSNR: **2.10**dB, ΔSSIM: **0.054**; see supplemental material for experimental setup. *Right:* Document-specific super-resolution; 2× and 3× magnification. Note our results (*bottom*) are more detailed (faces), and sharper (documents) than competing results (*middle*, [6]).

hyper-parameters for our each task-specific system were taken from the generic DBs so that the time-consuming parameter optimization stage was avoided.

4.2 Enhancement of JPEG images

We wish to remove DCT-coding block artifacts that are typical for JPEG images. In principle, one could build a single large model for processing compressed images at various different compression factors. However, a more economical approach might be to train a model for each compression factor. For practical applications, we propose two different scenarios: In the *off-line scenario*, many models specialized to each small interval of compression factors are trained such that the whole range of compression factors is covered. Then, the enhancement of a given encoded image is performed by choosing the closest model based on the compression ratio. In the *on-line scenario*, for every given image a model is instantly trained from the example image pairs generated with a known compression ratio.[5]

While our algorithm affords both scenarios, the second scenario is preferable since the system is specifically tailored to each input image.[6] Our semi-local approximation makes this feasible: Basic GP regression for 200,000 data points is infeasible and training the sparse Gaussian process model [6] took around 36 hours in preliminary experiments. As such, we focus on the on-line scenario and on a specific compression ratio: The quantization table (determining the compression ratios) Q2 in Table 2 of [15] (see supplemental for other compression factors).

5. Time-consuming parameter optimization can be avoided by optimizing parameters off-line in a manner similar to the off-line scenario.

6. Applying a Q3-trained system to Q2-compressed images decreased PSNR by 0.33dB.

To preprocess, input images undergo re-application of JPEG [15] which suppresses block-artifacts in an efficient way, but tends to leave ringing artifacts. Then, we use the same preprocessing steps as for super-resolution enhancement, and apply our algorithm. The compression ratios and DB1 images (Fig. 1, supplementary material) are used in many published JPEG and JPEG 2000 artifact removal works (e.g., [2], [19], [20], [22], [33]), allowing comparison. Standard images (e.g., 'Goldhill', 'Lena', and 'pepper') indicate that our JPEG artifact removal method is significantly better by PSNR. We compare against state-of-the-art approaches: the re-application of JPEG [15], and shape-adaptive DCT [17]. See supplemental material for comparison with another state-of-the-art algorithm by Laparra *et al*. [18].

All three methods already outperform many existing algorithms (cf. other algorithms reported in [15], [17], [18]). We also compare to two generic image prior-based algorithms with Gaussian noise models: Fields of Experts (FOEs) and the Product of Edge-perts (Edge-perts). The latter is used in our algorithm as a prior. This demonstrates how much performance gain can be achieved by adopting GP regression-based distortion models (plus preprocessing) instead of an i.i.d. Gaussian likelihood. As our algorithm preprocesses with re-application of JPEG, we also report FOE and Edge-perts on images preprocessed with re-application of JPEG. For these models, hyper-parameters (filter size and noise variance for FOE; noise variance for Edge-perts) were set to values which provide the best average PSNR values.

Visual inspection reveals that our method produces fewest artifacts and best preserves real image features (Fig. 6). This is numerically confirmed through PSNR and SSIM values, Table 2 (with Table 3 in supplementary material). Specific observations: Re-application of JPEG

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2015.2389797, IEEE Transactions on Pattern Analysis and Machine Intelligence

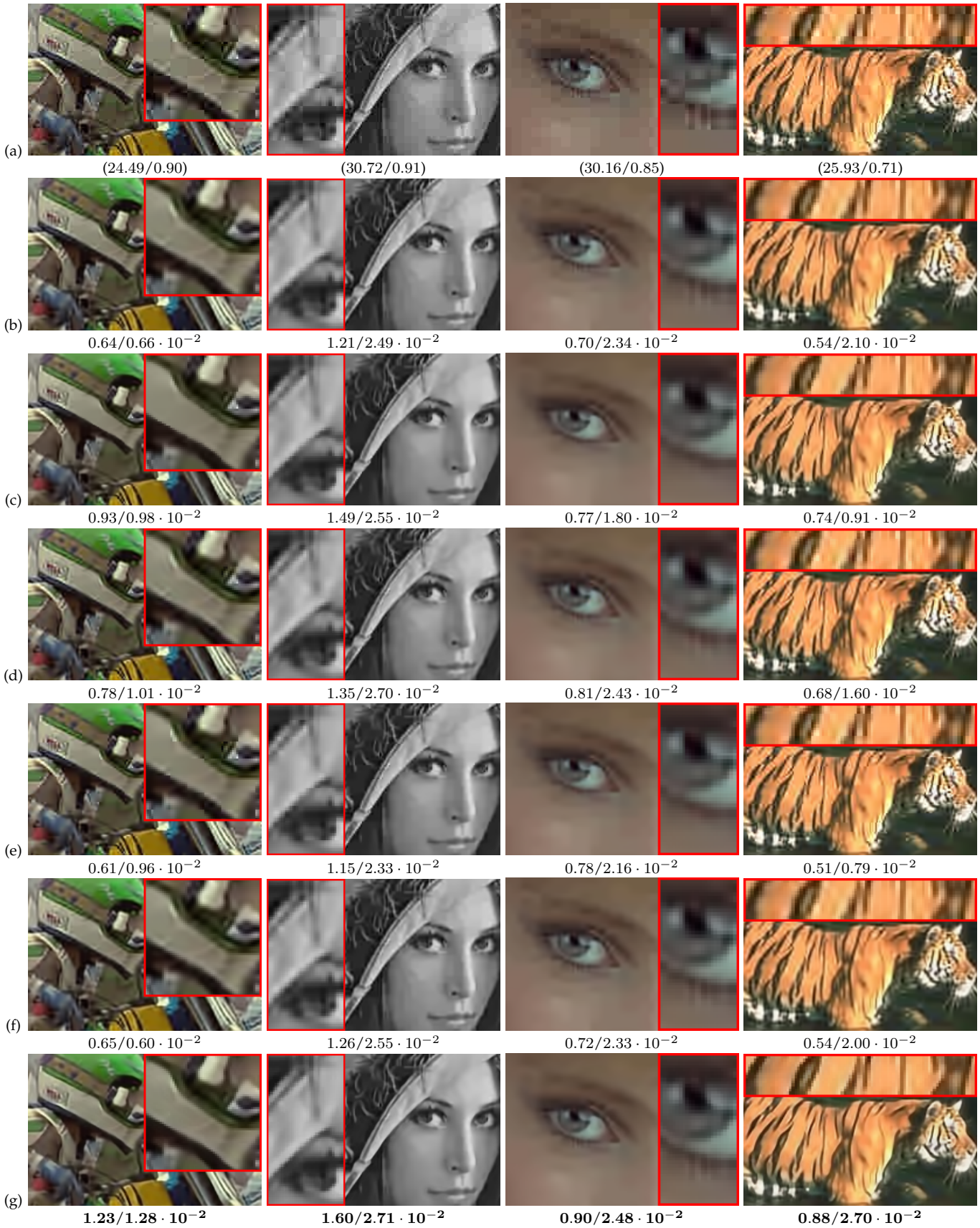


Fig. 6: Examples of JPEG artifact suppression — please zoom into the electronic version! (a) Input JPEG images, (b) re-application of JPEG [15], (c) SADCT [17], (d) Edge-perts [1], (e) and (f) FOE [3] applied to (a) and (b), respectively, and (g) our method. See supplemental material Figure 18 for uncropped originals. Note blur in (b), (d), and (f) and remaining block artifacts (biker; first column) in (e). Our approach (g) is slightly more detailed (woman; third column) and has fewer ringing artifacts (tiger; fourth column) than (c).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2015.2389797, IEEE Transactions on Pattern Analysis and Machine Intelligence

			[15]	[3]	[15]+ [3]	[1]	[17]	Ours
JPEG artifact removal	DB1	PSNR increase	0.76(0.21)	0.77(0.22)	0.77(0.22)	0.81(0.21)	0.97(0.29)	**1.09(0.30)**
		SSIM increase	2.06(0.83)	1.84(1.00)	2.08(0.88)	2.01(0.84)	1.91(1.22)	**2.41(1.01)**
	DB2	PSNR increase	0.71(0.27)	0.79(0.31)	0.72(0.28)	0.77(0.29)	0.91(0.34)	**1.00(0.36)**
		SSIM increase	1.32(0.73)	1.37(0.81)	1.31(0.77)	**1.53(0.71)**	1.05(0.98)	1.48(0.75)
				[11]	[14]	[13]	[6]	Ours
Super-resolution	DB1	PSNR increase		−1.64(0.63)	0.03(0.52)	−0.08(0.26)	1.74(0.50)	**1.80(0.54)**
		SSIM increase		−5.59(2.43)	0.72(0.45)	−0.55(0.69)	2.01(1.49)	**2.03(1.54)**
	DB2	PSNR increase		−0.05(0.43)	−2.41(0.62)	−0.13(0.31)	1.70(0.61)	**1.75(0.65)**
		SSIM increase		−5.10(2.16)	0.67(0.40)	−0.73(0.07)	**1.24(0.68)**	1.22(0.67)

TABLE 2: Performance of different image enhancement algorithms for two example applications: increases of PSNRs (dB) and SSIMs ($\times 10^{-2}$); mean and standard deviation. Re-application of JPEG [15], FOE [3], FOE applied to the results of Re-application of JPEG ([15]+ [3]), Edge-perts [1], SADCT [17], Chang *et al*. [11], Freeman *et al*. [14], He and Siu [13], and Kim and Kwon [6]. Best results are marked bold.

significantly reduces block artifacts which improves both visual quality and PSNR values. However, averaging differently encoded images results in slightly blurred edges and texture details. Still, the results are not completely free of block artifacts (see Lena). FOEs and Edge-perts successfully remove block artifacts which overall results in improvements of both PSNR and SSIM index over re-application of JPEG. However, as regularization methods, they do not show any noticeable enhancement of edge and texture details. Furthermore, similar to re-application of JPEG, they contain ringing artifacts, especially in JPEG 2000 images (see supplementary material).

Preprocessing with re-application of JPEG for Edge-perts and FOE did not significantly improve performance: noise variance hyper-parameter optimization for Edge-perts with re-application of JPEG resulted in zero noise variance: the results obtained with re-application of JPEG were worse than their corresponding original results. On the other hand, FOE resulted in, on average, $1.44 \cdot 10^{-2}$dB improved PSNR values from the input re-application of JPEG. However, in this case, the optimal noise variance turned out to be very small (2 in the scale of 256 images) and, therefore, no noticeable visual improvement was observed (Fig. 6b and f).

Shape-adaptive DCT (SADCT) successfully removes block artifacts and produces sharp edges. However, visual inspection (e.g., tiger stripe pattern, woman eyebrow, Fig. 6) reveals that our results are more detailed with fewer ringing artifacts. Even with a Gaussian noise assumption, two generic image prior-based algorithms (FOEs and Edge-perts) produce better results than re-application of JPEG (one of the best approaches). This result supports the use of a generic image prior for natural image enhancement; our additional improvement demonstrates the combined power of a generic image prior and an application-specific conditional model.

5 Discussion: Semi-local GP Locality

Our semi-local GP approximation relies on two steps of approximations. We motivate the second step (16) based on analysis of large-scale behavior of full GPs: For large l, the prediction $p(f_*|\mathcal{Y})$ is not affected by the data points which are sufficiently distinct from $\mathbf{x}_*$. This can be shown by first noting that the predictive distribution $p(f_*|\mathcal{Y})$ for an input $\mathbf{x}_*$ (see Eq. 9) can be rewritten as:

$$[\mathbb{E}[f(\mathbf{x}_*)]]_j = \sum_{i=1}^{l} \kappa(\|\mathbf{x}_* - \mathbf{x}_i\|)[\mathbf{Y}]_{i,j}, \tag{21}$$

$$\mathbb{V}[f(\mathbf{x}_*)] = k(\mathbf{x}_*, \mathbf{x}_*) - \sum_{i=1}^{l} \kappa(\|\mathbf{x}_* - \mathbf{x}_i\|)[\mathbf{K}_{\mathbf{f},*}]_i, \tag{22}$$

where κ is the *equivalent kernel* corresponding to k:

$$\kappa(\|\mathbf{x}_* - \mathbf{x}_i\|) \triangleq [\mathbf{K}_{*,\mathbf{f}}(\mathbf{K}_{\mathbf{f},\mathbf{f}} + \sigma^2\mathbf{I})^{-1}]_i, \tag{23}$$

which shows that actually $p(f_*|\mathcal{Y})$ is specified by two *kernel smoothers*.

An interesting property of the equivalent kernel is that it is *spatially localized* (i.e., $\kappa(\|\mathbf{x}_* - \cdot\|)$ diminishes quickly with distance from $\mathbf{x}_*$) regardless of the shape of the corresponding kernel k [40].

In the context of spline smoothing, Silverman [41] showed that there is an asymptotically exact approximation $\tilde{m}_j(\cdot)$ of $[\mathbb{E}[f(\mathbf{x}_*)]]_j$,

$$[\mathbb{E}[f(\mathbf{x}_*)]]_j \approx \tilde{m}_j(\mathbf{x}_*) = \sum_{i=1}^{l} \tilde{\kappa}(\|\mathbf{x}_* - \mathbf{x}_i\|)[\mathbf{Y}]_{i,j}, \tag{24}$$

where the corresponding approximate equivalent kernel $\tilde{\kappa}$ has the localization property. For the case of a Gaussian kernel k, an analytical approximation has been suggested by Sollich and Williams [42]:

$$\tilde{\kappa}(\|\mathbf{x} - \mathbf{y}\|) = \left(\frac{s_c}{\|\mathbf{x} - \mathbf{y}\|}\right)^{M^2/2} J_{M^2/2}(2\pi s_c \|\mathbf{x} - \mathbf{y}\|), \tag{25}$$

where $s_c^2 = \log\left(\frac{l(\pi b)^{M^2/2}}{\sigma^2}\right)/(\pi^2 b)$ and J is the Bessel function of the first kind. This result implies that as l increases, κ approaches to $\tilde{\kappa}$. Furthermore, the support of $\tilde{\kappa}(\|\mathbf{x}_* - \cdot\|)$ shrinks down and eventually converges to a single point $\mathbf{x}_*$ as $l \to \infty$. This becomes more explicit for one-dimensional signals i.e., $M^2 = 1$, since in this case,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2015.2389797, IEEE Transactions on Pattern Analysis and Machine Intelligence

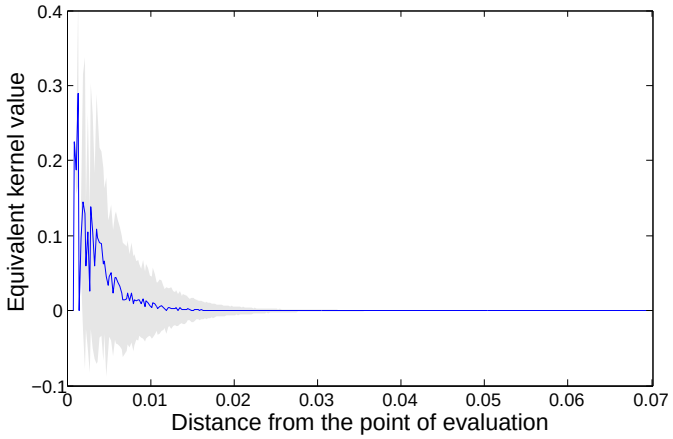


Fig. 7: Plot of $\kappa(r)$ as a function of distance $r(\cdot) = \|\mathbf{x}_* - \cdot\|^2$ for the data points used in super-resolution experiments: $20,000$ training data points are used in calculating $\mathbf{K}_{\mathbf{f},\mathbf{f}}$ while a distinct set of $20,000$ data points are used as the evaluation points $\{\mathbf{x}_*\}$. The gray area corresponds to twice the standard deviations.

$\tilde{\kappa}$ becomes (a constant multiple of) the *Sinc* function. See [42] for examples.

Accordingly, the localization behavior of κ should be especially prominent when l is large, which is the case for the current application of image enhancement. When the output variables $\mathbf{Y}$ correspond to the pixel-values of images, the variances of its elements are bounded by a constant. Together with the locality of κ, this shows that the *weight functions* $\{\kappa(\|\mathbf{x}_i - \cdot\|)\}$ corresponding to data points $\mathbf{x}_i$ that are distinct from $\mathbf{x}_*$ do not contribute significantly to the expansions (21). For large l, the expansions become mainly influenced by $\mathcal{B}_1(\mathbf{y}_*)$. In this case, Eq. 16 should be a good approximation and eventually, in the limit case, it becomes exact.

It is not straightforward to quantify the corresponding approximation error for finite l since (25) is only asymptotically exact. However, Fig. 7 shows that even for a relatively small $l(= 20,000)$, the qualitative behavior of κ is already in accordance with its analytic approximation $\tilde{\kappa}$ (i.e., κ oscillates locally and decays globally) and indeed, κ is strongly localized. This suggests that the prediction of f_* based on the observations made at the vicinity of $\mathbf{x}_*$ is a reasonable choice. This choice also maximizes the *differential entropy score* [43]. Since we do not know the proper values of $\mathcal{B}(f_*)$ and $\mathcal{B}_1(\mathbf{y}_*)$ in advance, we simply choose m and n-NNs.

6 Conclusion and Open Questions

Our learning-based approach can be quickly applied to new problems, even by users with no specific knowledge of the image enhancement operation to be performed. As suggested by the results in example applications, our approach can outperform or is on par with domain-specific algorithms. However, there is a trade-off between designing either a general framework or an application-specific approach, and we expect that the best enhancement quality can be achieved when one carefully selects the proper features and image representations. For instance, in our preliminary experiments, naïvely applying our algorithm to denoise images contaminated with Gaussian noise resulted in slightly worse reconstructions than GSM, which does not rely on any examples.[7]

Our algorithm provides interesting conceptual insights, allows for high-quality image enhancement in different scenarios, and allows us to customize the degradation models efficiently since the training time is very short. In the future, our algorithm could be applied to other computer vision problems such as video super-resolution, and image and video deblurring.

References

[1] P. V. Gehler and M. Welling, "Product of "edge-perts"," in *NIPS*, 2005, pp. 419–426.
[2] D. Sun and W.-K. Cham, "Postprocessing of low bit-rate block DCT coded images based on a fields of experts prior," *IEEE TIP*, vol. 16, no. 11, pp. 2743–2751, 2007.
[3] S. Roth and M. J. Black, "Fields of experts," *International Journal of Computer Vision*, vol. 82, no. 2, pp. 205–229, 2009.
[4] J. Portilla, V. Strela, M. Wainwright, and E. P. Simoncelli, "Image denoising using scale mixtures of Gaussians in the wavelet domain," *IEEE TIP*, vol. 12, no. 11, pp. 1338–1351, 2003.
[5] W. T. Freeman, E. C. Pasztor, and O. T. Carmichael, "Learning low-level vision," *International Journal of Computer Vision*, vol. 40, no. 1, pp. 25–47, 2000.
[6] K. I. Kim and Y. Kwon, "Single-image super-resolution using sparse regression and natural image prior," *IEEE TPAMI*, vol. 32, no. 6, pp. 1127–1133, 2010.
[7] Y. Kwon, K. I. Kim, J.-H. Kim, and C. Theobalt, "Efficient learning-based image enhancement: application to super-resolution and compression artifact removal," in *Proc. British Machine Vision Conference*, 2012, pp. 14.1–14.12.
[8] E. Snelson and Z. Ghahramani, "Sparse Gaussian processes using pseudo-inputs," in *NIPS*, 2006, pp. 1257–1264.
[9] C. Walder, K. I. Kim, and B. Schölkopf, "Sparse multiscale Gaussian process regression," in *Proc. International Conference on Machine Learning*, 2008, pp. 1112–1119.
[10] J. Quińonero-Candela and C. E. Rasmussen, "A unifying view of sparse approximate Gaussian process regression," *Journal of Machine Learning Research*, vol. 6, pp. 1939–1959, 2005.
[11] H. Chang, D.-Y. Yeung, and Y. Xiong, "Super-resolution through neighbor embedding," in *Proc. IEEE CVPR*, 2004, pp. 275–282.
[12] J. Yang, J. Wright, T. S. Huang, and Y. Ma, "Image super-resolution via sparse representation," *IEEE TIP*, vol. 19, no. 11, pp. 2861–2873, 2010.
[13] H. He and W.-C. Siu, "Single image super-resolution using Gaussian process regression," in *Proc. IEEE CVPR*, 2011, pp. 449–456.
[14] W. T. Freeman, T. R. Jones, and E. C. Pasztor, "Example-based super-resolution," *IEEE Computer Graphics and Applications*, vol. 22, no. 2, pp. 56–65, 2002.
[15] A. Nosratinia, "Denoising of JPEG images by re-application of JPEG," *Journal of VLSI Signal Processing*, vol. 27, no. 1, pp. 69–79, 2001.
[16] ——, "Postprocessing of JPEG-2000 images to remove compression artifacts," *IEEE Signal Processing Letters*, vol. 10, no. 10, pp. 296–299, 2003.
[17] A. Foi, V. Katkovnik, and K. Egiazarian, "Pointwise shape-adaptive DCT for high-quality denoising and deblocking of grayscale and color images," *IEEE TIP*, vol. 16, no. 5, pp. 1395–1411, 2007.
[18] V. Laparra, J. Gutiérrez, G. Camps-Valls, and J. Malo, "Image denoising with kernels based on natural image relations," *Journal of Machine Learning Research*, vol. 11, pp. 873–903, 2010.

7. On average, the PSNR values of our results were 0.04 lower than the results of GSM for the case of noise variance 0.01.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2015.2389797, IEEE Transactions on Pattern Analysis and Machine Intelligence

[19] G. Zhai, W. Lin, J. Cai, X. Yang, and W. Zhang, "Efficient quadtree based block-shift filtering for deblocking and deringing," *Journal of Visual Communication and Image Representation*, vol. 20, no. 8, pp. 595–607, 2009.

[20] T. Wang and G. Zhai, "JPEG2000 image postprocessing with novel trilateral deringing filter," *Optical Engineering*, vol. 47, no. 2, pp. 027 005–1–027 005–6, 2008.

[21] L. G. Gubin, B. T. Polyak, and E. V. Raik, "The method of projections for finding the common point of convex sets," *USSR Computational Mathematics and Mathematical Physics*, vol. 7, no. 6, pp. 1–24, 1967.

[22] Y. Yang, N. P. Galatsanos, and A. K. Katsaggelos, "Projection-based spatially adaptive reconstruction of block-transform compressed images," *IEEE TIP*, vol. 4, no. 7, pp. 896–908, 1995.

[23] X. Li, "Improved wavelet decoding via set theoretic estimation," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 15, no. 1, pp. 108–112, 2005.

[24] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE TIP*, vol. 13, no. 4, pp. 600–612, 2004.

[25] Z. Lin and H.-Y. Shum, "Fundamental limits of reconstruction-based superresolution algorithms under local translation," *IEEE TPAMI*, vol. 26, no. 1, pp. 83–97, 2004.

[26] S. Baker and T. Kanade, "Limits on super-resolution and how to break them," *IEEE TPAMI*, vol. 24, no. 9, pp. 1167–1183, 2002.

[27] S. Chaudhuri, *Super-Resolution Imaging*. Springer, 2001.

[28] M. F. Tappen, B. C. Russel, and W. T. Freeman, "Exploiting the sparse derivative prior for super-resolution and image demosaicing," in *Proc. International Workshop on Statistical and Computational Theories of Vision*, 2003.

[29] M. E. Tipping and C. M. Bishop, "Bayesian image super-resolution," in *NIPS*, 2003, pp. 1303–1310.

[30] P. J. Liu, "Using Gaussian process regression to denoise images and remove artefacts from microarray data," Ph.D. dissertation, Graduate Department of Computer Science, University of Toronto, Canada, 2007.

[31] D. Tschumperlé and R. Deriche, "Vector-valued image regularization with PDEs: a common framework for different applications," *IEEE TPAMI*, vol. 27, no. 4, pp. 506–517, 2005.

[32] G. Qiu, "MLP for adaptive postprocessing block-coded images," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 10, no. 8, pp. 1450–1454, 2000.

[33] K. Lee, D. S. Kim, and T. Kim, "Regression-based prediction for blocking artifact reduction in JPEG-compressed images," *IEEE TIP*, vol. 14, no. 1, pp. 36–49, 2005.

[34] H. A. Bourlard and N. Morgan, *Connectionist Speech Recognition: A Hybrid Approach*. Springer, 1993.

[35] L. C. Pickup, S. J. Roberts, and A. Zissermann, "A sampled texture prior for image super-resolution," in *NIPS*, 2004, pp. 1587–1594.

[36] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. MIT Press, 2006.

[37] M. Seeger, C. K. I. Williams, and N. Lawrence, "Fast forward selection to speed up sparse Gaussian process regression," in *Proc. Artificial Intelligence and Statistics*, 2003.

[38] E. Snelson and Z. Ghahramani, "Local and global sparse Gaussian process approximations," in *Proc. Artificial Intelligence and Statistics*, 2007.

[39] G. C. Cawley, N. L. C. Talbot, and O. Chapelle, "Estimating predictive variances with kernel ridge regression," in *Proc. Machine Learning Challenges: evaluating Predictive Uncertainty Visual Object Classification, and Recognizing Textual Entailment*, 2006, pp. 56–77.

[40] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.

[41] B. W. Silverman, "Spline smoothing: the equivalent variable kernel method," *The Annals of Statistics*, vol. 12, no. 3, pp. 898–916, 1984.

[42] P. Sollich and C. K. I. Williams, "Using the equivalent kernel to understand Gaussian process regression," in *NIPS*, 2005, pp. 1313–1320.

[43] N. Lawrence, M. Seeger, and R. Herbrich, "Fast sparse Gaussian process methods: the informative vector machine," in *NIPS*, 2003, pp. 625–632.

Younghee Kwon received BSc, MSc, and PhD degrees in Computer Science at KAIST in 2000, 2002, and 2007, respectively. Currently, he is a software engineer at Google. His research interest include computer vision and parallel and distributed systems.

Kwang In Kim received a BSc in computer engineering from the Dongseo University in 1996, and MSc and PhD in computer engineering from the Kyungpook National University in 1998 and 2000. He was a post-doctoral researcher at KAIST, at the Max-Planck-Institute for Biological Cybernetics, at Saarland University, and at the Max-Planck-Institute for Informatics, from 2000 to 2013. Currently, he is a lecturer at the School of Computing and Communications, Lancaster University. His research interests include machine learning, computer vision and computer graphics.

James Tompkin received an MSci degree in Computer Science from King's College London in 2006, and an EngD degree in Virtual Environments, Imaging, and Visualization from University College London in 2012. He was a post-doctoral researcher at the Max-Planck-Institute for Informatics from 2012 to 2014, and is currently a post-doctoral researcher at Harvard University. His research applies vision, graphics, machine learning, and interaction to create new visual computing tools and experiences.

Jin Hyung Kim worked at Hughes Research Laboratories after obtaining his PhD degree in Computer Science at University of California Los Angeles in USA. Since joining KAIST Computer Science department in 1985, he serves as the department Chair, as President of Korean R&D Information Center, and President of Korean Information Science Society. Prof. Kim is a member of the National Academy of Engineering of Korea, a fellow at the Korean Academy of Science and Technology and a fellow of International Association of Pattern Recognition. Prof. Kim received the Order of Service Merit Green Stripes from the Korean Government in 2001.

Christian Theobalt is a Professor of Computer Science at the Max-Planck-Institute for Informatics and Saarland University in Saarbrucken, Germany. He researches problems that lie on the boundary between the fields of Computer Vision and Computer Graphics, such as dynamic 3D scene reconstruction and marker-less motion capture. He received the Otto Hahn Medal of the Max-Planck Society in 2007, the EUROGRAPHICS Young Researcher Award in 2009, and the German Pattern Recognition Award in 2012.