

Efficient Learning-based  
Image Enhancement:  
Application to Compression  
Artifact Removal and  
Super-resolution

K. I. Kim, Y. Kwon, J.-H. Kim,  
C. Theobalt

MPI-I-2011-4-002 February 2011

## **Authors' Addresses**

K. I. Kim  
MPI Informatik  
Campus E14  
D-66123 Saarbrücken

Y. Kwon  
Google Inc.  
1600 Amphitheatre Parkway  
Mountain View, CA 94043, USA

J.-H. Kim  
CS Dept., KAIST  
335 Gwahangno, Yuseong-gu  
Daejeon 305-701, Korea

C. Theobalt  
MPI Informatik  
Campus E14  
D-66123 Saarbrücken

## **Acknowledgements**

The ideas presented in this study have greatly profited from discussions with Christian Walder, James Tompkin, Nils Hasler, Miguel Granados, and Carsten Stoll. Part of this work was done while K. I. Kim was with machine learning group, Saarland University.

# Efficient Learning-based Image Enhancement: Application to Compression Artifact Removal and Super-resolution

*Kwang In Kim, Younghée Kwon, Jin Hyung Kim, and Christian Theobalt*

**Abstract.** Many computer vision and computational photography applications essentially solve an image enhancement problem. The image has been deteriorated by a specific noise process, such as aberrations from camera optics and compression artifacts, that we would like to remove. We describe a framework for learning-based image enhancement. At the core of our algorithm lies a generic regularization framework that comprises a prior on natural images, as well as an application-specific conditional model based on Gaussian processes. In contrast to prior learning-based approaches, our algorithm can instantly learn task-specific degradation models from sample images which enables users to easily adapt the algorithm to a specific problem and data set of interest. This is facilitated by our efficient approximation scheme of large-scale Gaussian processes. We demonstrate the efficiency and effectiveness of our approach by applying it to example enhancement applications including single-image super-resolution, as well as artifact removal in JPEG- and JPEG 2000-encoded images.

---

## 1 Introduction

Many widely used imaging operations lead to specific degradations of an image with respect to the ground truth. The algorithmic removal of these degradations is one of the most important tasks in computer vision, image processing, and computational photography. For instance, image encoding deficiencies such as block artifacts have to be removed frequently. Deterioration and information loss due to the limitations of the optical system, such as limited sensor resolution or defocusing, should also be removed. This paper presents a framework to solve a variety of such enhancement operations which is based on efficient learning of application-specific enhancement models. Given such models, it is possible to perform an inference procedure to restore the original image.

Motivated by the recent success of Bayesian approaches in related image enhancement applications [1, 2, 3, 4, 5, 6], we model the imaging process as a combination of a *generic* prior on natural images and *application-specific* conditional models which take into account the degradation process corresponding to the specific enhancement application of interest.

In general, we expect that the result of any enhancement process should match the statistics of natural images: In wavelet representations, natural images exhibit **1)** a leptokurtic distribution of coefficients (the representation of images in the wavelet domain) and **2)** a high degree of redundancy which can result in high-order dependencies between coefficients. In this paper, we adopt the Product of Edge-perts (PoEdges) model as a prior. This model adopts a *sparsity* prior (i.e., Laplacian) over the pair-wise joint distribution of coefficients which, overall, prefers simultaneous activation of few coefficients in nearby scales and spatial locations [1]. As a result, the clustering behavior (i.e., dependencies of coefficients localized in frequency, space, and scale; which accounts for high-order dependencies [5]) of wavelet activities are effectively represented through the *product of experts* type factorization. The PoEdges model has been successfully applied to denoising under Gaussian noise. However, it is generic and, as we show in this paper, can also handle non-Gaussian noise when a suitable conditional model is applied.

For exploiting the knowledge of a degradation process, we apply non-parametric Gaussian process (GP) regression, instead of commonly used parametric noise models (Sec. 3). This relieves the user from the extremely difficult task of designing an analytical noise model, in particular for general non-Gaussian noise. Instead, with our framework, one can build an image enhancement system by preparing a set of example pairs of clean and degraded images and learning case specific conditional models from such training data.

We are not the first to apply learning-based image enhancement (see Sec. 2). However, we explicitly overcome the obstructive high run-time complexity of previous approaches. One of our main contributions is that, instead of time-consuming training and testing of a single GP model on a large dataset, a set of *sparse* models is constructed

*on-line* such that the prediction at each test data point is made by the corresponding sparse GP approximating the underlying global model (Sec. 3.2.3). We will demonstrate that during inference, i.e., enhancement, this method has a similar run-time complexity and performance as general sparse models [7, 8, 9, 10]. However, unlike existing models, our approach avoids the time-consuming training stage, and therefore facilitates easy customization of the framework to specific image enhancement problems. Our framework is generic, and is not restricted to any specific application. In the current paper, we demonstrate it with three exemplary image enhancement operations that can benefit from the addition of magnitude higher efficiency: single-image super-resolution, and enhancement of JPEG- and JPEG 2000-encoded images (Sec. 4). The experiments show that the proposed framework outperforms state-of-the-art systems that are specific to each task. While our algorithm is on par with the recent learning-based super-resolution algorithm of Kim and Kwon [9] in terms of performance and run-time, the experiments further demonstrate that the significantly shorter training time of our algorithm enables favorable new application scenarios which are infeasible with the sparse model of Kim and Kwon [9].

The paper continues with a review of relevant related work in Sec. 2. Thereafter, the general image enhancement model that we use is described, and our efficient semi-local approximation of Gaussian processes is described that quickly learns domain specific degradation models (Sec. 3). Section 4 demonstrates the utility of this generic algorithm with three example applications while Sec. 5 concludes the paper.

## 2 Related Work

A variety of image enhancement operations can be approached with our framework. In this section, we illustrate this by reviewing related literature from the three specific applications that we exemplify in this paper.

### 2.1 Enhancement of JPEG- and JPEG 2000 encoded images

Block-based discrete cosine transform (BDCT) coding is one of the most widely used tools for compressing still images (e.g., JPEG) and video sequences (e.g., MPEG). In BDCT coding, an image is partitioned into small blocks (typically of size  $8 \times 8$ ) which are independently transformed using the DCT, quantized, and encoded. The BDCT has appealing properties such as energy compaction, orthogonality that leads to decorrelation, and efficiency in computation. 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 not only resulted in superior low bit-rate performance but also enabled progressive coding and region-of-interest coding which are particularly important in browsing images on the web [11, 12]. JPEG 2000 prevents block artifacts, but *ringing artifacts* may still appear.

Removal of encoding artifacts is an example of image enhancement under non-Gaussian and correlated noise [13]. In this perspective, several papers propose adaptive filtering for block artifact removal, i.e., locally adjusting filter kernels to remove block edges while preserving image edges [14]. A similar technique has also been applied to the removal of ringing artifacts in the context of trilateral filters [15]. Zhai *et al.* [16] recently 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.

In general image enhancement, incorporation of *a priori* knowledge about natural images can prove to be very beneficial. In principle, an image model incorporating a generic prior of natural images can be applied to any type of enhancement applications with suitable modification of the noise model (e.g. [3, 4]; or even with a Gaussian noise model). The theory of *projection onto convex sets* (POCS) is a widely used tool in this context. The basic ingredient of this approach is to represent *a priori* knowledge of natural images as a sequence of convex constraints (more specifically, a sequence of convex sets whose elements satisfy certain constraints corresponding to known characteristic of natural images). Then, for a given image, the element which is closest to that image and satisfies all the constraints can be retrieved by simple iteration through the projection onto each individual set. Commonly used constraints include spatial smoothness [17], quantization constraints, etc. POCS has been successfully applied to JPEG [17] and JPEG 2000 [18] image enhancement.

A rather direct way of utilizing *a priori* knowledge is to encode it into a distribution or an energy functional. Alter *et al.* [19] proposed weighting the degree of total variation (TV) penalization depending on the complexity of region of interest (adapted TV) such that block boundaries are penalized more, while texture-areas are less-penalized. Sun and Cham [2] proposed a *maximum a posteriori* (MAP) framework where the prior is modeled as a Markov random field with learned potentials. Then, the clique potential for MRF is learned from a set of natural

images based on the *Fields of Experts* [3], which led to an improved performance over several existing methods including those methods based on POCS and overcomplete wavelet representation.

Nosratinia [20] proposed another promising 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. Then, the denoised image is obtained by simply averaging these re-encoded images. Despite its simplicity, re-application of JPEG demonstrated superior performance over the algorithms based on nonlinear filtering, POCS, and overcomplete wavelets. An application to JPEG 2000 enhancement is also feasible [21].

The success of machine learning in many computer vision and image enhancement applications also inspired a new class of artifact suppression algorithms. The basic idea is to estimate a function which maps the given encoded image to the desired clean image, based on example pairs of JPEG-encoded images and the corresponding uncompressed images. Qiu [22] used a multi-layer Perceptron for JPEG deblocking, which receives the gradient of pixel values in a one-dimensional cross-section across the block boundary and produces an estimation of the difference between the encoded image and the original image. A similar algorithm was also developed in the MAP framework by Lee *et al.* [23] who proposed performing the piecewise linear regression in the space of DCT coefficients and showed comparable results to those of re-application of JPEG. Laparra *et al.* [24] recently proposed a generic wavelet domain framework where the distributions of the source (clean image) and the noise were estimated non-parametrically based on support vector regression (SVR). Unlike the other machine learning-based approaches, this method is unsupervised in the sense that it only utilizes clean images as examples. The role of SVR is to regularize the noisy image in wavelet representation such that the histogram of the residual (the noise estimate) plus the reconstruction (the signal estimate) approaches to that of the corresponding empirical distributions. As a non-parametric model, it can be used in various image enhancement applications including the enhancement of JPEG-encoded images. In the case of the removal of a Gaussian noise, this method outperformed Gaussian scale mixture (GSM) which is one of the best image denoising method, in terms of a perceptually oriented quality measure (structural similarity (SSIM) index [25]).

## 2.2 Single-image super-resolution

Single-image super-resolution is the task of constructing a high-resolution enlargement of a single low-resolution image. While discussions on various existing approaches related to image super-resolution can be found in [26, 27, 28] and references listed therein, the most closely related approaches to our proposed framework are the example-based methods which identify a function mapping a low-resolution image (patch) to a high-resolution counterpart based on example pairs. Freeman *et al.* proposed a nearest neighbor (NN)-based algorithm [29]. For each patch in the input low-resolution image, the corresponding high-resolution example patch is retrieved through NN-search which enforces spatial consistency. Baker and Kanade [27] represented images based on the Laplacian pyramid and estimated each pixel in high-resolution image using an NN search enforcing the consistency of the gradients around the pixel of interest. Similar algorithms were also derived as a special instance of image analogies [30] and in the context of MAP framework [31]. Chang *et al.* [32] extended this idea by additionally introducing a reconstruction constraint obtained based on a manifold assumption. In the context of regression estimation, Tappen *et al.* [33] 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 [9]. This combination resulted in an improvement over several existing algorithms including [29, 32, 33]. Meanwhile, Yang *et al.* [34] adopted the idea of *sparse coding* in super-resolution, which has shown impressive performance on several image enhancement applications. The underlying idea is to represent a low-resolution input as a sparse combination of stored example inputs. The combination coefficients are then used to synthesize the corresponding outputs based on the retrieved example outputs. In theory, all example-based approaches could also be generalized to other image enhancement approaches. However, our framework achieves much better results than previous methods including [29, 32, 33, 34] and produces the results which are comparable to [9] but with two orders of magnitude faster training time, which dramatically enhances its applicability.

### 3 The image enhancement framework

#### 3.1 Prior on natural images

For the case of Gaussian noise, the PoEdges model provides a MAP framework in the decorrelated<sup>1</sup> wavelet domain:

$$\begin{aligned}\mathbf{z}^* &= \arg \max_{\mathbf{z}} (\log p(\tilde{\mathbf{z}}|\mathbf{z}) + \log p(\mathbf{z})) \\ &= \arg \min_{\mathbf{z}} \left( \frac{1}{2} \|\tilde{\mathbf{z}} - \mathbf{z}\|^2 + \sigma_P \left( \sum_j w_j [\mathbf{z}]_j^2 \right)^{\alpha_P} \right),\end{aligned}$$

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

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

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

where  $\mathcal{I}[\cdot]$  is the degradation process of interest and  $\mathcal{W}^\#(\mathbf{z})$  is the preimage of  $\mathbf{z}$ . However, in general,  $\mathcal{I}$  is non-differentiable and is not even continuous which can lead to difficulties in the optimization.

To keep computations tractable, we bypass the optimization through the degradation process and penalize instead the below given cost functional:

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

where the *reference variable*  $\mathbf{s}_i$  is constructed as a convex combination of candidates  $\mathbf{f}_i \in \mathbb{R}^N$  based on their *confidences*  $\mathbf{c}_i$  (i.e.,  $\mathbf{s}_i = \mathbf{f}_i^\top \mathbf{c}_i, [\mathbf{c}_i]_j \geq 0, \|\mathbf{c}_i\|_{L_1} = 1$ ). The confidence vector  $\mathbf{c}_i \in \mathbb{R}^N$  is calculated based on the *predictive variances*  $\mathbf{v}_i$  of the candidates  $\mathbf{f}_i$ :

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

where the scale parameter  $\sigma_C$  is fixed at 0.2. Both  $\mathbf{f}_i \in \mathbb{R}^N$  and  $\mathbf{v}_i \in \mathbb{R}^N$  are estimated during the regression step (Sec. 3.2).

In this model, the degradation process is taken into account only indirectly by  $\mathbf{s}$  which encodes the information contained in the training examples. The model is computationally favorable. Unfortunately, in this case an intuitive probabilistic interpretation of the image enhancement process is not possible. Especially, since the conditional model is not generative, sampling a degraded image is not directly possible. However, for image enhancement applications, this is not a serious problem. A similar strategy has been exercised in [6]. In general, one could regard (1) as a regularization framework where one trades between the enforcement of regularity in the wavelet domain and the deviation from the reference variable  $\mathbf{s}$ .

#### 3.2 Regression

For each pixel location in the input degraded image  $\tilde{\mathbf{x}}$ , a Gaussian process (GP) regressor receives a patch (of size  $M \times M$ ) centered at that location, and produces estimates of a desired patch (of size  $N \times N$ ). As described later, GP prediction is obtained as a Gaussian distribution which in the current context of patch regression, is specified by the mean patch and the corresponding variance patch. When  $N > 1$ , the output patch overlaps its spatial neighbors, which for each pixel location  $i$ , constitutes a set of candidates for the pixel values  $\mathbf{f}_i$  and the corresponding predictive variances  $\mathbf{v}_i$ , respectively.

---

<sup>1</sup>For brevity, we omit the details of the decorrelation and the estimation procedure for the clusters (cf. [1] for details).

### 3.2.1 Basic Gaussian process regression

Suppose there is a given set of data points (i.e. 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  $\sigma^2 \mathbf{I}$

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

where  $\mathcal{N}(\mu, \Sigma)$  is the probability density of the Gaussian random variable with mean  $\mu$  and covariance  $\Sigma$  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 [35]:<sup>2</sup>

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

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

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

With this prior, the prediction is made by conditioning on the labels

$$p(\mathbf{f}_* | \mathcal{Y}) = \mathcal{N}(\mathbf{K}_{*, \mathbf{f}}(\mathbf{K}_{\mathbf{f}, \mathbf{f}} + \sigma^2 \mathbf{I})^{-1} \mathbf{Y}, \mathbf{K}_{*, *} - \mathbf{K}_{*, \mathbf{f}}(\mathbf{K}_{\mathbf{f}, \mathbf{f}} + \sigma^2 \mathbf{I})^{-1} \mathbf{K}_{\mathbf{f}, *}), \quad (3)$$

where  $\mathbf{Y}$  is a matrix each row of which corresponds to an element of  $\mathcal{Y}$ .

The predictions of GP regression (3) are given in the form of a Gaussian distribution where the diagonal terms of the covariance matrix (predictive variances) represent the uncertainty of regression output while the off-diagonal terms correspond to 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 (cf. [35] 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*  $\mathbf{f}_{\mathcal{U}} = \{f(\mathbf{u}_1), \dots, f(\mathbf{u}_m)\}$  (corresponding to *inducing inputs*  $\mathcal{U} = \{\mathbf{u}_1, \dots, \mathbf{u}_m\}$ ) through which the conditional independence of  $\mathbf{f}_*$  and  $\mathbf{f}$  is assumed in the approximation of the joint prior (cf. the unified framework of [10]):

$$p(\mathbf{f}_*, \mathbf{f}) \approx q(\mathbf{f}_*, \mathbf{f}) = \int q(\mathbf{f}_* | \mathbf{f}_{\mathcal{U}}) q(\mathbf{f} | \mathbf{f}_{\mathcal{U}}) p(\mathbf{f}_{\mathcal{U}}) d\mathbf{f}_{\mathcal{U}}. \quad (4)$$

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  $\mathcal{X}$  [7, 8, 36]. For instance, Seeger *et al.* [36] proposed an approximate prior

$$q(\mathbf{f}_*, \mathbf{f}) = \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} \mathbf{Q}_{\mathbf{f}, \mathbf{f}} & \mathbf{Q}_{\mathbf{f}, *} \\ \mathbf{Q}_{*, \mathbf{f}} & \mathbf{K}_{*, *} \end{bmatrix}\right), \quad (5)$$

where  $\mathbf{Q}_{\mathbf{a}, \mathbf{b}} = \mathbf{K}_{\mathbf{a}, \mathbf{u}} \mathbf{K}_{\mathbf{u}, \mathbf{u}}^{-1} \mathbf{K}_{\mathbf{u}, \mathbf{b}}$ . The corresponding predictive distribution is

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

With this prior, the predictive mean is obtained as a linear combination of evaluations of  $m$  basis functions  $\{k(\mathbf{u}_1, \cdot), \dots, k(\mathbf{u}_m, \cdot)\}$  (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.

---

<sup>2</sup>For computational convenience, each output is treated independently and identically. For notational convenience, we omit conditioning on input variables.

### 3.2.3 Semi-local approximation of Gaussian processes

In existing sparse GP algorithms, once identified, the inducing inputs  $\mathcal{U}$  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 [7, 8, 35, 36] 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 accordingly is not easy to solve.

Our approach is fundamentally different from the existing algorithms: We build a sparse GP which is specially tailored for a given 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  $\mathcal{U}_*$ .<sup>3</sup> If we introduce a spatial Markov assumption on  $\{f_*, \mathbf{f}\}$

$$p(f_* | \mathbf{f}, \mathcal{N}(f_*)) \approx q(f_* | \mathcal{N}(f_*)), \quad (6)$$

where  $\mathcal{N}(f_*)$  denotes the values of  $f$  for the inputs in the spatial neighborhoods  $\mathcal{N}(\mathbf{x}_*) \subset \mathcal{X}$  of  $\mathbf{x}_*$ , the approximation (4) becomes exact once we use  $\mathcal{N}(\mathbf{x}_*)$  for  $\mathcal{U}_*$ .

The spatial Markov assumption is fairly natural and has proven to be effective in many different applications. Although the resulting sparse model can represent only local variations at  $\mathbf{x}_*$ , the corresponding prediction takes into account the entire data set through the dependency between  $\mathbf{f}$  and  $\mathbf{f}_{\mathcal{U}_*}$  (see Eq. 4). Accordingly the corresponding joint distribution  $q(f_*, \mathbf{f})$  still fits into the approximation (5) (i.e. it is a valid approximation of the full GP). In particular, since the model is globally regularized, no overfitting occurs. This is in contrast to well-known moving least-squares algorithm which is not directly related to any global regularization.

From a theoretical perspective, a drawback of our on-line model is that it does not correspond to any globally consistent GP: Similarly to other sparse models, in our model, the prior is defined through the inducing variables (Eq. 4). Since the inducing variables are depending on each test input, it is not possible to construct a joint prior  $p(\mathbf{f}_*, \mathbf{f})$  over the entire set of testing points. A direct consequence of this inconsistency is that the predictions do not obey the marginalization rule and accordingly, no prediction can be made for off-diagonal elements of the predictive covariance which represents the dependency between predictions.<sup>4</sup> However, in the applications we foresee for this model, this is not a major concern as we focus on using the mean and predictive variance of each individual prediction. Another limitation is that there is no guarantee that the resulting localized predictions constitute a globally smooth prediction. Again, this is not a serious problem in general image enhancement applications. For applications where the global smoothness is a concern, one might have to introduce a smooth *partition of unity* [38].

This new approximation dramatically reduces the computation time during training. Actually, the only training component is building a data structure for NN-search which facilitates identifying  $\mathcal{N}(f_*)$ . During the prediction, it takes  $O(Mlm + lm^2)$  time for each test point with  $m = |\mathcal{N}(f_*)|$ , which includes the time spent for building a model. However, for large  $l$  ( $\approx 2 * 10^5$  in the current applications), this approximation might be still impractical. The second step of our approximation is obtained by introducing an additional Markov-like assumption on the observations  $\mathcal{Y}$ :

$$p(f_* | \mathcal{Y}, \mathcal{N}_1(\mathbf{y}_*)) \approx q(f_* | \mathcal{N}_1(\mathbf{y}_*)), \quad (7)$$

where  $\mathcal{N}_1(\mathbf{y}_*)$  denotes the observed training target values in the spatial neighborhood  $\mathcal{N}_1(\mathbf{x}_*)$  of  $\mathbf{x}_*$ . In general, none of (6) and (7) is stronger than the other since neither is implied by the other. Only when the noise level is zero, the two assumptions become equivalent. However, from a practical perspective, (7) can be regarded as a stronger assumption than (6) since the former implies that given  $\mathcal{N}_1(\mathbf{y}_*)$ , all the remaining training data points are irrelevant in making prediction of  $f_*$ , which is not the case for the latter. Accordingly, we set  $\mathcal{N}_1(\mathbf{x}_*)$  much wider than  $\mathcal{N}(\mathbf{x}_*)$  (i.e.  $\mathcal{N}_1(\mathbf{x}_*) \supset \mathcal{N}(\mathbf{x}_*)$ ). This guarantees that the resulting GPs are non-locally regularized.

The second step (7) is motivated by the analysis of the large-scale behavior of full GPs: For large  $l$ , the prediction  $p(f_* | \mathcal{Y})$  (Eq. 3) is not affected by the data points which are sufficiently distinct from  $\mathbf{x}_*$ . This can be shown by

---

<sup>3</sup>Still, in this framework, one could try to adopt the above-mentioned optimization strategy. However, this might result in a less attractive algorithm since in this case, the run-time complexity will be much higher than that of existing approximate GPs.

<sup>4</sup>Snelson and Ghahramani have proposed a similar inconsistent GP approximation [37] where the training data are pre-partitioned (during training) into a set of clusters which constitute input-dependent inducing inputs.

firstly noting that  $p(f_*|\mathcal{Y})$  is represented based on kernel evaluations of  $\mathcal{X}$ :

$$\begin{aligned}\mathbb{E}[f(\mathbf{x}_*)]_j &= \sum_{i=1}^l \alpha_{ij} k(\mathbf{x}_*, \mathbf{x}_i) \\ \mathbb{V}[f(\mathbf{x}_*)] &= k(\mathbf{x}_*, \mathbf{x}_*) - \sum_{i=1}^l \beta_i k(\mathbf{x}_*, \mathbf{x}_i),\end{aligned}\quad (8)$$

where  $\alpha_{ij} = [(\mathbf{K}_{\mathbf{f},\mathbf{f}} + \sigma^2 \mathbf{I})^{-1} \mathbf{Y}]_{i,j}$  and  $\beta_i = [(\mathbf{K}_{\mathbf{f},\mathbf{f}} + \sigma^2 \mathbf{I})^{-1} \mathbf{K}_{\mathbf{f},*}]_i$ . Rearranging the multiplicands inside the summands one can restate (8)

$$\begin{aligned}[\mathbb{E}[f(\mathbf{x}_*)]]_j &= \sum_{i=1,\dots,l} \kappa(\|\mathbf{x}_* - \mathbf{x}_i\|) [\mathbf{Y}]_{i,j}, \\ \mathbb{V}[f(\mathbf{x}_*)] &= k(\mathbf{x}_*, \mathbf{x}_*) - \sum_{i=1,\dots,l} \kappa(\|\mathbf{x}_* - \mathbf{x}_i\|) [\mathbf{K}_{\mathbf{f},*}]_i,\end{aligned}\quad (9)$$

where  $\kappa$  is the *equivalent kernel* corresponding to  $k$

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

This shows that the predictive distribution 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  $k$  [39].

In the context of spline smoothing, Silverman [40] showed that there is an asymptotically exact approximation  $\tilde{m}_j(\cdot)$  of  $[\mathbb{E}[f(\mathbf{x}_*)]]_j$ ,

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

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 [41]:

$$\tilde{\kappa}(\|\mathbf{x} - \mathbf{y}\|) = \left( \frac{s_c}{\|\mathbf{x} - \mathbf{y}\|} \right)^{M^2/2} J_{M^2/2}(2\pi s_c \|\mathbf{x} - \mathbf{y}\|), \quad (10)$$

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,  $\kappa$  approaches to  $\tilde{\kappa}$ . Furthermore, the support of  $\tilde{\kappa}(\|\mathbf{x}_* - \cdot\|)$  shrinks down and eventually it converges to a single point  $\mathbf{x}_*$  as  $l \rightarrow \infty$ . Accordingly, the localization behavior of  $\kappa$  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, which with the locality of  $\kappa$  shows that the *weight functions*  $\{\kappa(\|\mathbf{x}_i - \cdot\|)\}$  corresponding to data points  $\mathbf{x}_i$  which are distinct from  $\mathbf{x}_*$  do not contribute significantly to the expansion (9) and equivalently to (8). For large  $l$ , the expansions become mainly influenced by  $\mathcal{N}_1(\mathbf{y}_*)$ . In this case, Eq. 7 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 (10) is only asymptotically exact. However, Fig. 1 shows that even for a relatively small  $l (= 20,000)$ , the qualitative behavior of  $\kappa$  is already in accordance with its analytic approximation  $\tilde{\kappa}$  (i.e.,  $\kappa$  oscillates locally and decays globally) and indeed,  $\kappa$  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. It should be noted that this choice also maximizes the *differential entropy score* [42].

Since we do not know the proper values of  $\mathcal{N}(f_*)$  and  $\mathcal{N}_1(\mathbf{y}_*)$  in advance, in practice, we simply choose  $m$  and  $n$ -nearest neighbors (NNs)  $\mathcal{C}_n(\mathbf{x}_*)$  ( $\mathcal{C}_m(\mathbf{x}_*) \subset \mathcal{C}_n(\mathbf{x}_*) \subset \mathcal{X}$ ) of  $\mathbf{x}_*$  such that the prediction is performed based on  $n$  data points which are summarized by  $m$  inducing inputs:

$$q(f_* | \mathcal{Y}(\mathcal{C}_n(\mathbf{x}_*))) = \mathcal{N}(\sigma^{-2} \mathbf{K}_{*,\mathbf{u}} \Sigma \mathbf{K}_{\mathbf{u},\mathbf{c}} \mathbf{Y}(\mathcal{C}_n(\mathbf{x}_*)), \mathbf{K}_{*,*} - \mathbf{Q}_{*,*} + \mathbf{K}_{*,\mathbf{u}} \Sigma \mathbf{K}_{\mathbf{u},*}), \quad (11)$$

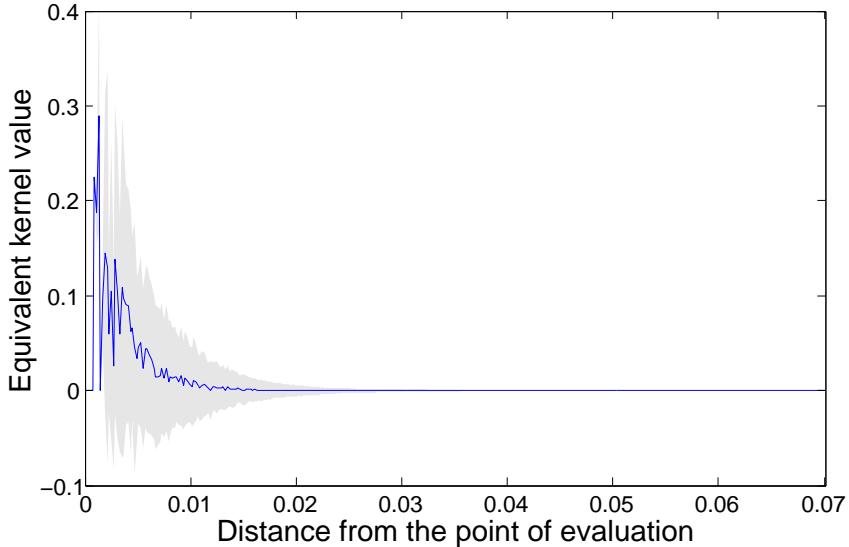


Figure 1: Plot of  $\kappa(r)$  as a function of distance  $r(\cdot) = \|\mathbf{x}_* - \cdot\|^2$  for the data points used in super-resolution experiments: 20,000 training data points are used in calculating  $\mathbf{K}_{f,f}$  while a distinct set of 20,000 data points are used as the evaluation points  $\{\mathbf{x}_*\}$ . The gray area corresponds to twice the standard deviations.

where in a compact notation  $\mathcal{Y}(A)(\mathbf{Y}(A))$  represents the subset of  $\mathcal{Y}$  (rows of  $\mathbf{Y}$ ) corresponding to the elements of  $A \subset \mathcal{X}$ ,  $\Sigma = (\sigma^{-2} \mathbf{K}_{\mathbf{u},\mathbf{c}} \mathbf{K}_{\mathbf{c},\mathbf{u}} + \mathbf{K}_{\mathbf{u},\mathbf{u}})^{-1}$ , 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 (see Sec. 4 for details).

This new approximation will henceforth be referred to as a *semi-local GP* approximation. With the same number of inducing inputs, this input-dependent selection should in general provide more flexibility than the standard sparse methods. Furthermore, as exemplified by Fig. 2, even with a fairly small number of inducing inputs (such that the time complexity of testing becomes similar to that of a standard sparse regression), semi-local GPs perform comparably to sparse methods which use relatively large numbers of carefully chosen inducing inputs.<sup>5</sup>

There is another advantage to the semi-local GP: given hyper-parameters, the only training component is building a data structure for NN-search, and so the off-line processing is very fast. Therefore, the framework is very flexible as the system can be easily adapted to the distribution of a specific (non-generic) class of images (cf. Sec. 4.2).

### 3.2.4 Adaptive noise model

There are two sources of uncertainty in making predictions with GPs [43]. 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, these two types of uncertainties ( $U1$  and  $U2$ ) are independently modeled with the noise parameter ( $\sigma^2$ ) and the kernel parameter ( $b$ ), respectively. This clear separation is due to the use of an *i.i.d.* Gaussian noise model (Eq. 2) which has an important advantage of leading to an analytical model for predictive distribution (Eq. 3).

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

In this section, we present a simple noise model which exploits the dependency between the two sources of uncertainty. Our scheme is based on the empirically observed correlation between two quantities which are related to  $U1$  and  $U2$ , respectively. The first quantity ( $Q1$ ) is the average distance from a test input to its nearest training inputs, which represents  $U1$ . The second quantity ( $Q2$ ) is the average distance among the corresponding retrieved training outputs. This is not directly related to  $U2$ . However, when  $Q1$  is zero,  $Q2$  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,  $Q1$  and  $Q2$  are (mildly) correlated: If  $Q1$  is small, the corresponding training inputs

<sup>5</sup>In the experiments in Fig. 2, the inducing inputs were optimized by maximizing the *marginal likelihood*  $p(\mathbf{f}_* | \mathbf{u})$  (cf., [7]).

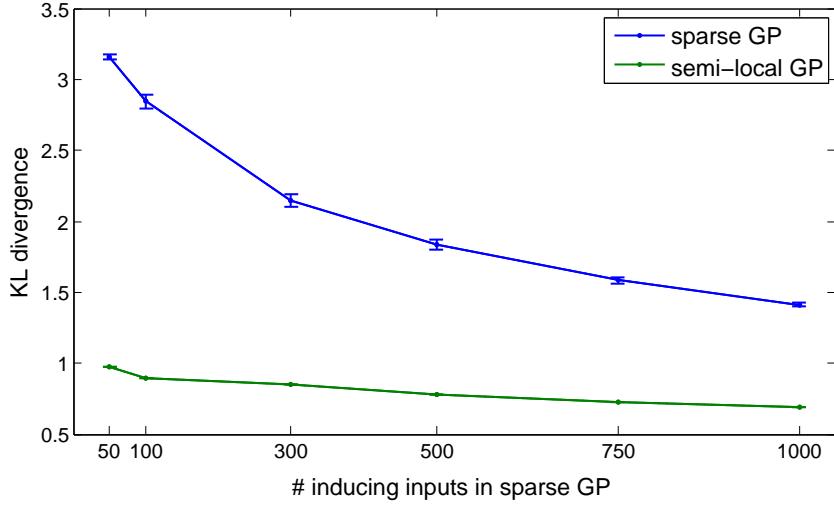


Figure 2: Approximation accuracies of sparse GPs and semi-local GPs with respect to a full GP for the data points used in super-resolution experiments: to facilitate the comparison with full GP, only 20,000 data points are used in training all models, which are sampled from a large training set of size 200,000. The Kullback-Leibler divergences of predictive distributions of approximate GPs (from those of the full PGs) are plotted against different numbers  $m'$  of inducing inputs for a sparse GP. For each  $m'$ , a semi-local GP was trained with the number of inducing inputs  $m$  and local training data points  $n$  such that the time complexity of a prediction for a test point roughly matches  $((m')^2 \approx m^2 n)$ . The lengths of error bars correspond to twice the standard deviations.

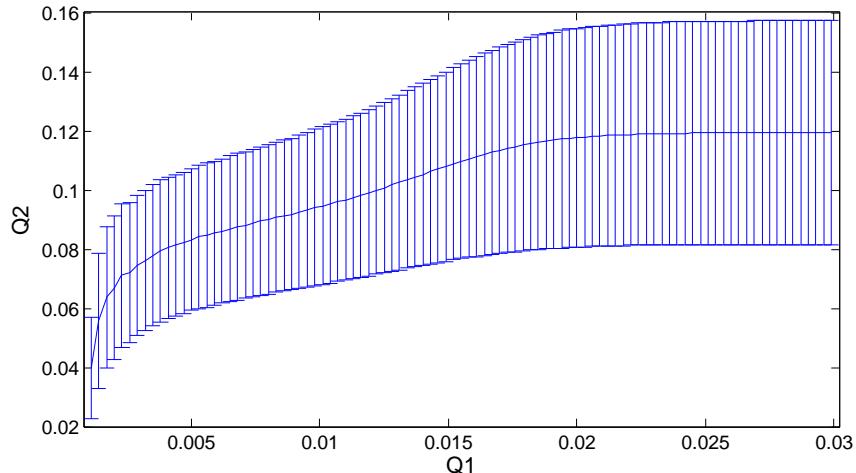


Figure 3: Variation of  $Q_2$  as a function  $Q_1$  in super-resolution experiments (see text for the descriptions of  $Q_1$  and  $Q_2$ ): For each test input, 10 nearest neighbors are selected and the corresponding  $Q_1$  and  $Q_2$  values are calculated. The length of error bars correspond to twice the standard deviation in repeated experiments.

should be close to each other and so  $Q_2$  tends to be small. However, Fig. 3 shows there are cases where the correlation is much stronger than what is expected. Specifically, when  $Q_1$  is very small (e.g.  $Q_1 < 0.005$ ; i.e., the test input is very close to some training inputs),  $Q_1$  and  $Q_2$  are especially strongly correlated. As  $Q_1$  increases, the correlation becomes milder (as expected) and eventually it disappears. This observation leaded us to conjecture that  $U_2$  is indeed correlated to  $U_1$  especially when  $U_1$  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  $\sigma^2$  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:

$$q(f_* | \mathcal{Y}(\mathcal{C}_n(\mathbf{x}_*))) = \mathcal{N}(\mathbf{K}_{*,\mathbf{u}} \Sigma' \mathbf{K}_{\mathbf{u},\mathbf{c}} \Gamma \mathbf{Y}(\mathcal{C}_n(\mathbf{x}_*)), \mathbf{K}_{*,*} - \mathbf{Q}_{*,*} + \mathbf{K}_{*,\mathbf{u}} \Sigma' \mathbf{K}_{\mathbf{u},*}),$$



Figure 4: Color map of *easy* data points in super-resolution experiments (see Sec. 4.2 for details). For each pixel in the image, the corresponding image patch which is centered at the pixel of interest is compared with training patches and the corresponding  $Q_1$  value is calculated. The pixels marked with green color corresponds to the data points whose  $Q_1$  value is smaller than 0.005 (cf. Fig. 3). Flat image patches are pre-excluded.

where

$$\begin{aligned}\Sigma' &= (\mathbf{K}_{\mathbf{u}, \mathbf{c}} \Gamma \mathbf{K}_{\mathbf{c}, \mathbf{u}} + \mathbf{K}_{\mathbf{u}, \mathbf{u}})^{-1}, \\ \Gamma &= \text{diag}[N_c \exp(-N_d \mathbf{b} \mathbf{d})] + \sigma^{-2} \mathbf{I},\end{aligned}\quad (12)$$

$\mathbf{d}$  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  $\text{diag}[\cdot]$  is an operator which converts a vector into a diagonal matrix.

From the regularization perspective, the noise variance  $\sigma^2$  is the parameter trading the contribution of training error and the global regularization term. From this perspective, an intuitive explanation of our noise model is that when the given input is sufficiently close to the training data we rely more on the data than the regularization term. 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 (11). However, this model resulted in on average 0.08 improvement of PSNR values in our super-resolution experiments.

An interesting observation concerning the quantity  $Q_1$  is that most of the *easy* test points that have small  $Q_1$  values are those lying at the major edges as shown in Fig. 4. 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  $Q_1$  and  $Q_2$  for small  $Q_1$  values. The role of our adaptive noise model (12) is then to regularize less for those patterns lying at the major edges. A visually noticeable consequence is that ringing artifacts are significantly reduced in the results. As discussed in [9], 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 putting more emphasize on observed data than the regularizer, we can effectively suppress these regularization artifacts which, in the case of images, appear as ringing artifacts.

Kim and Kwon [9] had to adopt a post-processing step to explicitly remove the ringing artifact, which is not necessarily when we use an adaptive noise model. Figure 5 demonstrates this with an example.

### 3.2.5 Comparison with the algorithm of Kim and Kwon [9]

Perhaps the most strongly related to the proposed algorithm is the super-resolution algorithm of Kim and Kwon [9] in which a prior on natural images is combined with a learning-based conditional model similarly to our method. In principle, this algorithm can also be applied to various different applications. However, there is an important limitation in the algorithm of [9], which we explicitly overcome: to achieve the high-level of accuracy and reasonable execution time, it requires a day and half of training time.

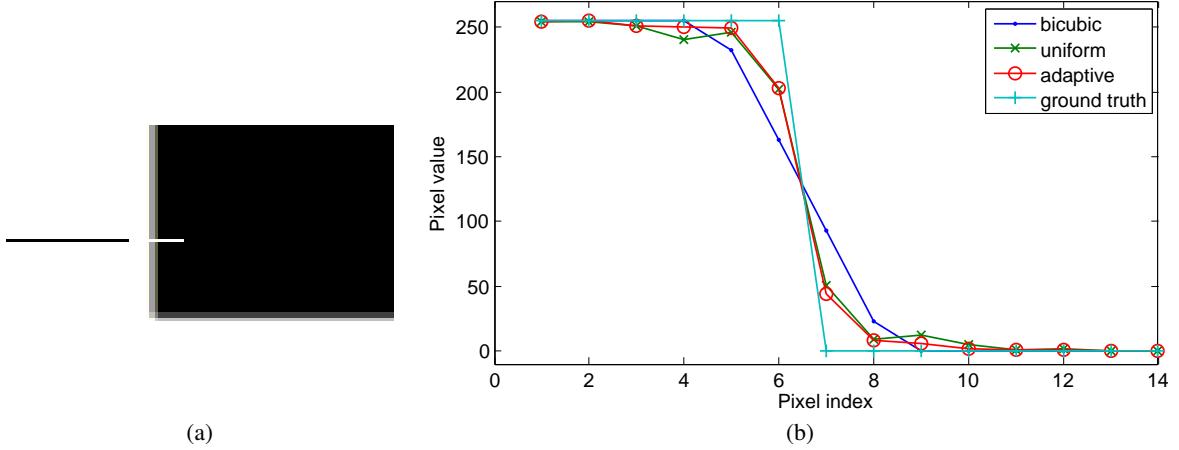


Figure 5: An example of a major edge: (a) an image showing a strong edge (bicubic-resampled image is shown) and (b) cross-sections of the super-resolution results at the locations marked with a white horizontal bar in (a). In the result of our algorithm that uses the homogeneous noise model (denoted as ‘uniform’) a fluctuation occurred at the vicinity of the edge (pixel indices 4 and 9). This is suppressed when the adaptive noise model is used instead.

In terms of prediction accuracy, our semi-local GP gives slightly better results than the sparse algorithm of [9]. However, for similar execution time, our algorithm only takes 5 minutes for training. This can result in a significant difference in the utilities of the two algorithms: Our algorithm can be quickly adapted to new application domains, which is not possible for [9]. Furthermore, we have 4 hyper-parameters to be tuned for each application ( $M$ ,  $\sigma_P$ ,  $\sigma^2$ , and  $b$ )<sup>6</sup> while in [9], 8 hyper-parameters need to be tuned (for each application). Since the kernel ridge regression used in [9] corresponds to the mean estimation of a GP, the two algorithms share the same set of hyper-parameters in the regression stage. The difference in the number of hyper-parameters stems from the fact that in [9] an explicit post-processing step is used to suppresses the ringing artifacts, which has additional hyper-parameters (to be tuned for each application). On the other hand, in our algorithm, the ringing artifact is suppressed by adaptively controlling the degree of regularization where the hyper-parameters can be shared in different applications. The time required for training and the number of hyper-parameters of [9] restricts the application domain significantly. For instance, the algorithm of [9] cannot be applied in interactive setting, i.e., the user tunes the parameters by visually inspecting results, which is feasible in our algorithm. Experimental results supporting this claim are provided in Sec. 4.2.

## 4 Results

This section demonstrates three different applications of the proposed framework: enhancement of JPEG and JPEG 2000 images (Sec. 4.1) and single-image super-resolution (Sec. 4.2). To evaluate the performance in each case, we used sixteen images (of size  $512 \times 512$  or  $256 \times 256$ ) shown in Fig. 6, which are disjoint from the training images where 200,000 training data points are sampled for GP regressions. To facilitate quantitative evaluation, for each clean image a degraded image was generated according to the corresponding task (e.g., JPEG encoding for the enhancement of JPEG images and blurring plus sub-sampling for super-resolution, respectively). The subsequently enhanced images were then compared with the original images in terms of the increases of peak signal-to-noise ratio (PSNR)<sup>7</sup> and the perceptually-based structural similarity (SSIM) index [25] with respect to the input degraded images. On average, (in *on-line scenario* (cf., Sec. 4.1)) processing a single image took around two minutes plus five minutes of training (degrading images, sampling training data, and building an NN-search tree) which was done once for all 16 test images for each task. The run-times of the other algorithms compared in the current paper are summarized in Table 1. For the processing of color images, the input image is represented in

<sup>6</sup>The remaining hyper-parameters are fixed throughout the entire set of experiments in the current paper.

<sup>7</sup>The PSNR value of an (vectorized) image  $O$  with respect to the reference image  $T$  is defined as

$$\text{PSNR}(O|T) = 10 \log_{10} \left( \frac{255^2}{\|O - T\|_2^2} \right).$$

Table 1: Average execution times of algorithms compared in the experiments; for a  $512 \times 512$ -sized image on a 3.6GHz machine.

Algorithm	Execution time (sec.)
Nosratinia [20] (JPEG)	3
Gehler and Welling [1] (JPEG)	14
Roth and Black [3] (JPEG)	2,600
Foi <i>et al.</i> [44] (JPEG)	27
Laparra <i>et al.</i> [24] (JPEG) ( $256 \times 256$ -sized images $\times 4$ )	280
Freeman <i>et al.</i> [29] (Super-resol.; Mag. factor 2)	21
Chang <i>et al.</i> [32] (Super-resol.; Mag. factor 3)	28
Yang <i>et al.</i> [34] (Super-resol.; Mag. factor 3)	151
Kim and Kwon [9] (Super-resol.; Mag. factor 3)	60

Table 2: Parameters used in the experiments

Expr. idx.		$M$	$\sigma^2$	$b$	$\sigma_P$	$N_c$	$N_d$	$N$
JPEG	Q2	7	$5 * 10^{-3}$	10	2.0			
	Q3	9	$5 * 10^{-3}$	8.3				
JPEG 2000	0.15BPP	7	$5 * 10^{-2}$	34	3.0	10	20	5
	0.1BPP	7	$5 * 10^{-3}$	28				
Super-resolution	Mag. factor 2	7	$5 * 10^{-8}$	20	0.5			
	Mag. factor 3	9	$5 * 10^{-8}$	90				

YIQ color space where the rest of enhancement steps are performed only for the Y component. The final result is then obtained by combining the resulting Y component with the I and Q components of the input image.

There are several parameters to be tuned, which include the input and output patch sizes ( $M$ ) and ( $N$ ), the regularization parameter ( $\sigma_P$ ), and the (hyper-)parameters for regression (the noise parameters  $\sigma^2$ ,  $N_c$ , and  $N_d$ , the kernel parameter  $b$ , and the numbers of training and inducing inputs  $n$  and  $m$ , respectively). The parameters  $m$  and  $n$  were fixed at 50 and 200, respectively. These values are determined by trading the performance against the computational complexity. We observed steady performance increases as  $m$  and  $n$  increase. At the same time, the run-time grew roughly quadratically and linearly with respect to  $m$  and  $n$ , respectively. Our preliminary experiments suggested that the optimal values of the other parameters vary significantly depending on the application except for  $N_c$ ,  $N_d$  (see Eq. 12), and  $N$  with respect to which, the performance variations are not so rapid. We fixed the parameters  $N_c$ ,  $N_d$ ,  $N$  at 10, 20, and 5, respectively throughout the entire set of experiments. These values were decided based on a set of validation images which are disjoint from the images used in training and testing in JPEG artifact removal experiments. The remaining parameters were optimized for each application (based on a set of validation images). Table 2 summarizes the corresponding parameters.

#### 4.1 Enhancement of JPEG- and JPEG 2000-encoded images

Here, our primary goal is to remove DCT-coding block artifacts in the JPEG case, and remove ringing artifacts that are typical for JPEG 2000 images. For each case, we build a specific enhancement model with our framework. 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 specialized to each compression factor. For practical applications, we propose two different scenarios. In the *off-line scenario*, a model specialized to each small interval of compression factors is trained such that the whole range of compression factors is covered by several models. Then, the enhancement of a given encoded image can be performed by choosing the closest model

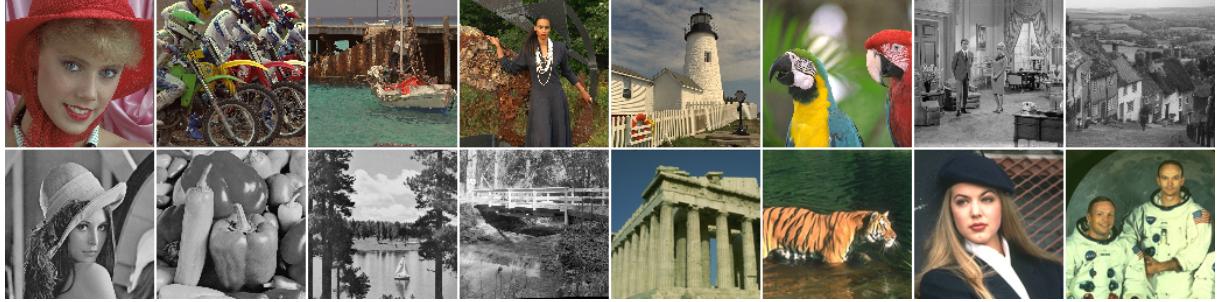


Figure 6: Gallery of test images: the images are referred to in the text by their positions in raster order.

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 the compression ratio at hand.<sup>8</sup>

While the proposed framework can afford both application scenarios, in general, the second scenario is preferable since the resulting system is more flexible: The system is specifically tailored to each input image. It should be noted that only by our semi-local approximation this actually becomes feasible. Basic GP regression for 200,000 data points is entirely infeasible and training the sparse Gaussian process model [9] took around a day and a half in our preliminary experiments.

In the experiments, we will be focusing on the on-line scenario and on specific compression ratios: the quantization tables (which determine the compression ratios) Q2 and Q3 in Table 2 of [20] for JPEG encoding and for 0.15 and 0.1 bits per pixel (BPPs) for JPEG 2000. For the encoding of JPEG 2000 images, we used the Kakadu software.<sup>9</sup>

As a preprocessing step, similarly to [29], the input images are first band-frequency filtered based on the Laplacian of Gaussian (LOG) filter. Given a patch of a LOG filtered image, the regressor estimates a patch corresponding to the difference between the input and the underlying ground truth such that the output candidates  $\{f_i\}$  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 based on the  $L^1$  norm of the input patch and the regression output is inverse normalized. To process pixels lying at the image boundary, the input images were extended by symmetrically replicating pixel values across the image boundary. For the case of JPEG image enhancement, the input JPEG images are first preprocessed by re-application of JPEG [20] which already suppresses block-artifacts in an efficient way, but tends to leave ringing artifacts.

The compression ratios used in the current paper as well as our test set of images, Fig. 6, are also employed in many published JPEG and JPEG 2000 image enhancement works (e.g., [2, 15, 16, 17, 23]), which allows us to compare our method to these approaches. Especially the enhancement results corresponding to the classical standard images (e.g., ‘Goldhill’, ‘Lena’, and ‘pepper’ images: from 8-th to 10-th images in Fig. 6) reported in these publications indicate that the proposed JPEG image enhancement method is significantly better in terms of PSNRs. To compare against most closely related state-of-the-art JPEG enhancement approaches, the re-application of JPEG [20] and shape-adaptive DCT [44] are evaluated (using publicly available source code).<sup>10</sup> We also compare with Laparra *et al.*’s support vector regression (SVR)-based algorithm [24] (Figs. 10 and 14). To enable a fair comparison with this algorithm, only six gray level images (from the 7-th to 12-th images in Fig. 6) were used. These images were cropped to  $256 \times 256$ -pixels around the center and were encoded using Matlab’s ‘imwrite’ function with compression quality 7 (as was done in [24]).<sup>11</sup> In this set of experiments, the hyper-parameters of the proposed method optimized for Q2 were re-used as suggested by our on-line scenario.

It should be noted that all three methods already outperformed many existing algorithms (cf., comparison with the other algorithms reported in [20, 24, 44]). We also compare to two generic image prior-based image enhance-

<sup>8</sup>In this case, the time consuming parameter optimization can be avoided by optimizing them off-line in a manner similar to the off-line scenario.

<sup>9</sup><http://www.kakadusoftware.com/>.

<sup>10</sup><http://www.cs.tut.fi/~foi/SA-DCT/>.

<sup>11</sup>This experimental setting is not the fundamental limitation of Laparra *et al.*’s algorithm. This setting enables us to directly use the code kindly shared by the authors at [http://www.uv.es/vista/vistavalencia/denoising\\_SVR/index.html](http://www.uv.es/vista/vistavalencia/denoising_SVR/index.html).

Table 3: Performances of different JPEG 2000 enhancement algorithms for Lena, pepper, and bridge images ( $\Delta$ PSNR(dB))

Image	[16]	[15]	Our method
Lena (0.1BPP)	0.03	0.49	<b>0.50</b>
pepper (0.1BPP)	0.31	0.11	<b>0.47</b>
bridge (0.1BPP)	-0.05	N.A.	<b>0.04</b>
Lena (0.15BPP)	-0.34	N.A.	<b>0.54</b>
pepper (0.15BPP)	0.04	N.A.	<b>0.62</b>
bridge (0.15BPP)	<b>0.13</b>	N.A.	0.10

ment algorithms with Gaussian noise models: Fields of Experts (FOEs) and the Product of Edge-perts (PoEdges). The latter is used in our algorithm as a prior. This comparison should demonstrate how much the performance gain can be achieved by adopting the degradation-specific conditional models (i.e., Gaussian process regression). For these models, the hyper-parameters (filter size and the noise variance for FOE and the noise variance for PoEdges) were manually set at those values providing the best average PSNR values, respectively.

For comparison with existing JPEG 2000 image enhancement algorithms, Nosratinia’s JPEG 2000 enhancement algorithm [21], FOEs, and PoEdges were evaluated and the reported results of Zhai *et al.*’s block-shift filtering-based approach [16] and Wang and Zhai’s trilateral filters [15] are compared in Table 3.<sup>12</sup>

Visual inspection of our result to the results obtained with the other methods (Figs. 7-12), reveals that our method produces the least number of artifacts and preserves actual image features best. This is also numerically confirmed through the PSNR and SSIM values, Figs. 13-14 and Table 3.

Specific observations include: re-application of JPEG significantly reduced block artifacts which improved both the visual quality and PSNR values. However, averaging differently encoded images resulted in slightly blurred edges and texture details. Furthermore, as shown in the result on the Lena image (Fig. 7) it is not completely free of block artifacts for JPEG images. FOEs and PoEdges successfully removed block artifacts which overall resulted in improvements of both PSNR and SSIM index (cf. Fig. 13) over re-application of JPEG. As regularization methods, however, they did not show any noticeable enhancement of edge and texture details. Furthermore, similarly to re-application of JPEG, they still showed a considerable amount of ringing artifacts, especially in JPEG 2000 images. The SVR-based method [24] successfully removed block artifacts and produced sharp edges (cf. the last two columns of Fig. 10). However, it tended to leave ringing artifacts and accordingly, made the results more noisy than other methods. The results of the proposed method are almost as sharp as the results of SVR at the edges. Furthermore, our results show fewer ringing artifacts as our method coherently reconstructs sharp edges and texture details. This is clearly visible in the visor of Lena (Fig. 10) and in the helmet of the biker in Fig. 9. The shape-adaptive DCT (indexed as ‘SADCT’ in each figure) also successfully removed block artifacts and produced sharp edges. However, detailed visual inspection (e.g. in the stripe pattern of the tiger and at the eyebrow of the woman of Fig. 8) reveals that the results of our method are much cleaner (with less ringing artifacts) and more detailed. It is noteworthy that even with a Gaussian noise assumption, two generic image prior-based algorithms (FOEs and PoEdges) produced better results than re-application of JPEG which is one of the best JPEG and JPEG 2000 image enhancement algorithms. This result supports the use of a generic image prior for natural image enhancement. The improvement achieved by the proposed method over these algorithms further demonstrates the utility of the combination of a generic image prior and an application-specific conditional model.

## 4.2 Single-image super-resolution

Adopting the framework of [9, 29], the input low-resolution image is first enlarged to the target scale by bicubic resampling. Then, we use the same preprocessing steps as for JPEG and JPEG 2000 enhancement, and apply our framework. We will focus on magnifications by 2 and 3 along each dimension (i.e., the number of pixels in the super-resolved image is  $2^2$  or  $3^2$ -times larger than that of the low-resolution image).

<sup>12</sup>The readers are advised to be cautious regarding the absolute values of PSNR: even with the same bit rates and using the same encoding software, the PSNRs of the JPEG 2000-encoded images used in the current paper are slightly different ( $\pm 0.05$ dB) from those reported in [15, 16].



Figure 7: Examples of artifact suppression for *Lena* image (Q2): (a) input JPEG image, (b) re-application of JPEG [20], (c) SADCT [44], (d) FOE [3], (e) PoEdges [1], and (f) our method. Increases of PSNRs (in dB) and SSIMs with respect to the input JPEG images (displayed below each column) were calculated based on the complete images. For the input image (a), the original PSNR and SSIM values are shown. The best results are marked with bold letters. Please refer to the electronic version of the current paper for better visualization.

Figures 15-18 show the results of super-resolution. For comparison, the results of Freeman *et al.*'s algorithm [29], Chang *et al.*'s algorithm [32], and Kim and Kwon's algorithm [9]<sup>13</sup> are also displayed. For a magnification factor of 3, we additionally present a comparison with Yang *et al.*'s sparse coding-based algorithm [34].<sup>14</sup>

All tested super-resolution algorithms outperformed the simple baseline method, namely, bicubic resampling. Freeman *et al.*'s algorithm [29] and Yang *et al.*'s algorithm [34] produced sharper but partially noisy images. The results of Chang *et al.*'s algorithm are less noisy but more blurry than the results of [29] and [34]. For general images, the results of our approach and [9] are equally good (Figs. 15 and 16) except for the tiger image in the magnification factor 3 case where slight ringing artifacts observed in the result of [9] are disappeared in our results (the first column of Fig. 16). Overall, the results of both algorithms are as sharp as but a lot less noisy than [29] and [34]. The latter observation is also confirmed through the PSNR and SSIM measures.

It should be noted that Kim and Kwon's algorithm already outperformed state-of-the-art algorithms, including Tappen *et al.*'s algorithm utilizing a prior of natural images [33] and Fattal's algorithm based on the statistical analysis of edges [45]. However, for the same level of testing time and performance, the proposed method is around 300 times faster in training than [9]. This is important especially when *a priori* knowledge is available in terms of a class-specific set of example images. For instance, if it is known that the image of interest to be magnified is representing a document or a face (whose statistical properties might be distinct from those of general images), one could quickly generate examples from these specific class of images on which the system is trained. While this leads to much better results as shown in Figs. 17 and 18, it is infeasible in [9] due to its high complexity

<sup>13</sup><http://www.mpi-inf.mpg.de/~kkim/supres/supres.htm>.

<sup>14</sup>We used the code and the corresponding dictionary (i.e., a set of basis vectors used in representing the images, trained for a magnification factor of 3, cf. [34] for details) kindly made available by the authors at <http://www.ifp.illinois.edu/~jyang29/>.

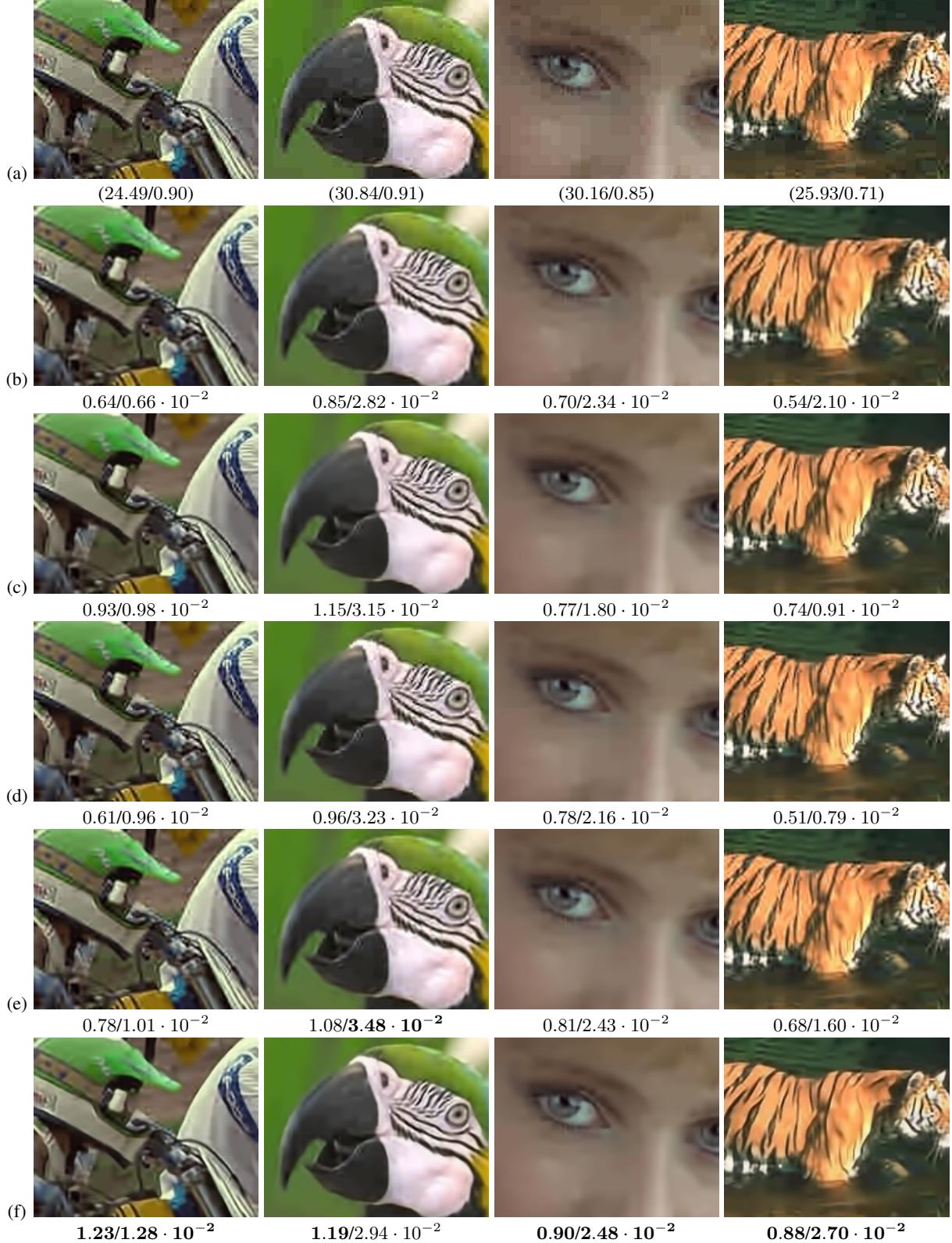


Figure 8: Examples of JPEG artifact suppression (Q2): (a) input JPEG images, (b) re-application of JPEG [20], (c) SADCT [44], (d) FOE [3], (e) PoEdges [1], and (f) our method.

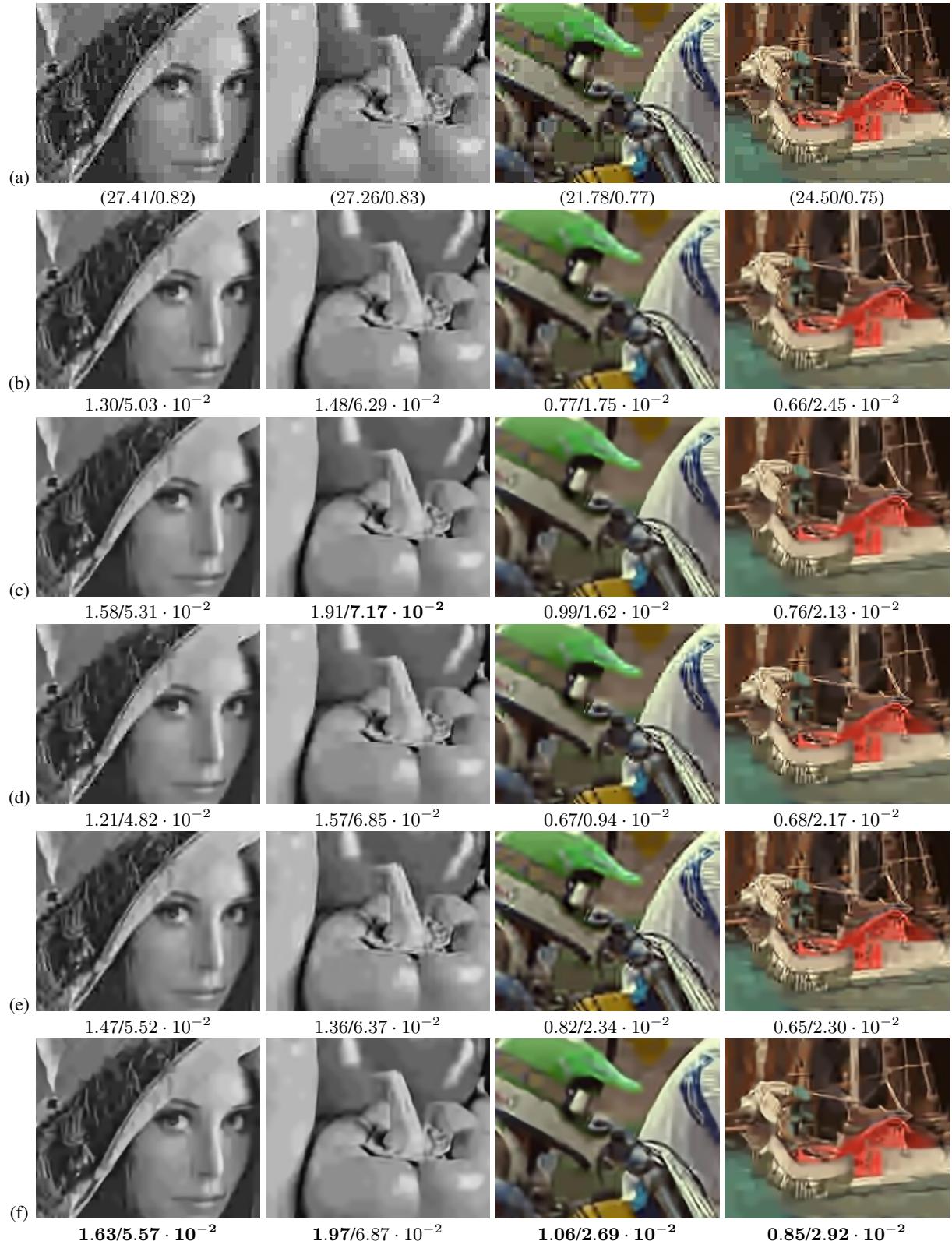


Figure 9: Examples of JPEG artifact suppression (Q3): (a) input JPEG images, (b) re-application of JPEG [20], (c) SADCT [44], (d) FOE [3], (e) PoEdges [1], and (f) our method.



Figure 10: Examples of JPEG artifact suppression on  $(256 \times 256)$ -size sub-images (compression quality 7): (a) input JPEG images, (b) SVR-based method [24], (c) SADCT [44], and (d) our method.



Figure 11: Examples of JPEG 2000 artifact suppression (0.15BPP): (a) input JPEG 2000 images, (b) re-application of JPEG [20], (c) FOE [3], (d) PoEdges [1], and (e) our method.



Figure 12: Examples of JPEG 2000 artifact suppression (0.1BPP): (a) input JPEG 2000 images, (b) re-application of JPEG [20], (c) FOE [3], (d) PoEdges [1], and (e) our method.

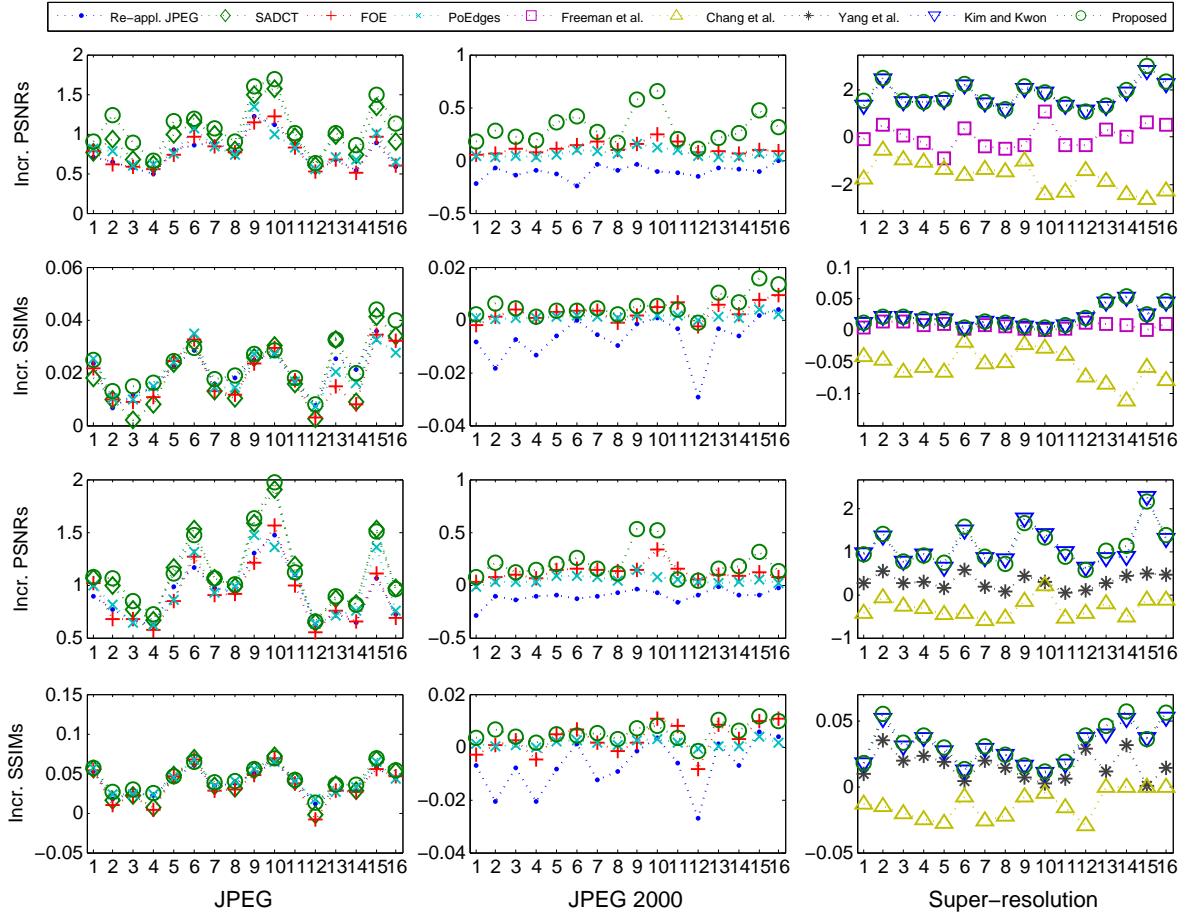


Figure 13: Performance of different image enhancement algorithms for three example applications: from top to bottom (JPEG) Q2 and Q3, (JPEG 2000) 0.15BPP and 0.1BPP, and (super-resolution) mag. factors 2 and 3, respectively.

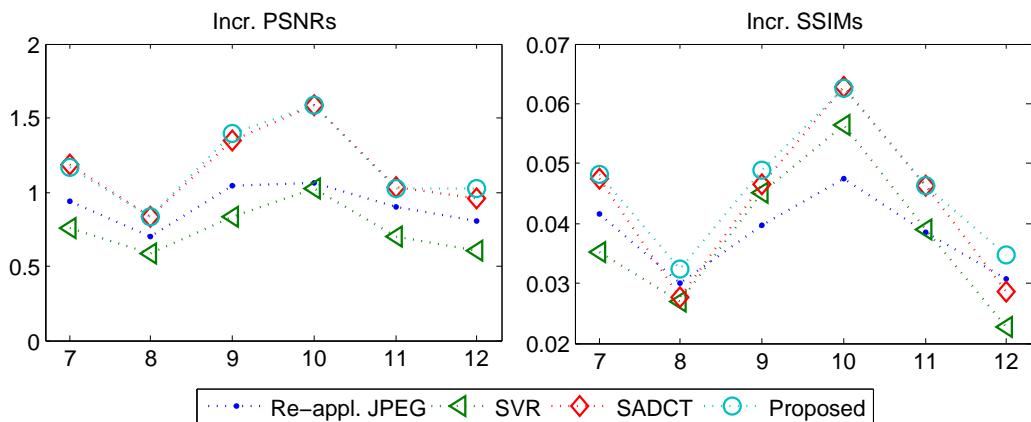


Figure 14: Performance of different JPEG enhancement algorithms for  $(256 \times 256)$ -size sub-images.

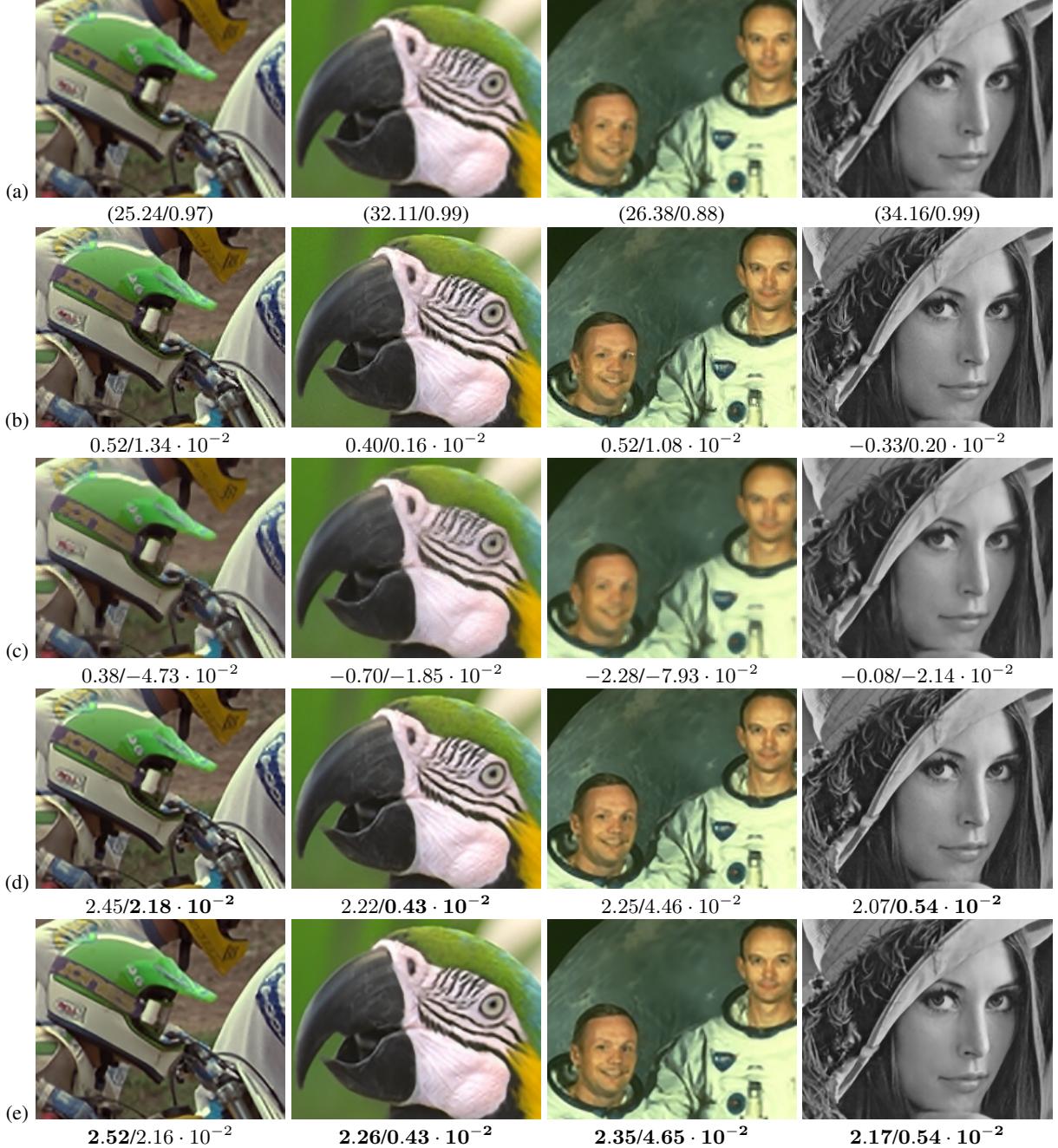


Figure 15: Examples of image super-resolution (mag. factor 2): (a) bicubic resampling, (b) Freeman *et al.* [29], (c) Chang *et al.* [32], (d) Kim and Kwon [9], and (e) our method.

in training.<sup>15</sup> The hyper-parameters for our each task-specific system were adopted from the same systems trained on the generic DBs so that the time-consuming parameter optimization stage was avoided.

---

<sup>15</sup>Experimental setup for face image super-resolution experiment is different from the case of generic images. Details can be found in the appendix.



Figure 16: Examples of image super-resolution (mag. factor 3): (a) bicubic resampling, (b) Chang *et al.* [32], (c) Yang *et al.* [34], (d) Kim and Kwon [9], and (e) our method.

for each  $v_{i=1..k}$ . We have for each  $v_{i=1..k}$ . We have for each  $v_{i=1..k}$ . We have of them, one per visual w of them, one per visual w of them, one per visual w of them, a component may b tors, a component may b tors, a component may b difference in Equation 1. difference in Equation 1. difference in Equation 1.

One can observe that tl One can observe that tl One can observe that tl (few values have a signif (few values have a signif (few values have a signif most high descriptor valu most high descriptor valu most high descriptor valu and the geometrical struc and the geometrical struc and the geometrical struc

for each  $v_{i=1..k}$ . We have for each  $v_{i=1..k}$ . We have for each  $v_{i=1..k}$ . We have of them, one per visual w of them, one per visual w of them, one per visual w of them, a component may b tors, a component may b tors, a component may b difference in Equation 1. difference in Equation 1. difference in Equation 1.

One can observe that tl One can observe that tl One can observe that tl (few values have a signif (few values have a signif (few values have a signif most high descriptor valu most high descriptor valu most high descriptor valu and the geometrical struc and the geometrical struc and the geometrical struc

Figure 17: Example of super-resolution for document images: from top to bottom, magnification factors 2 and 3, respectively. (left) bicubic resampling, (middle) Kim and Kwon [9], and (right) our method trained based on a document image DB (with different sizes, fonts, resolutions, etc.). There is no ground truths for the document images.

## 5 Conclusion and open questions

There is a trade-off between designing either a general framework or an application-specific approach. Our general 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. In addition, as suggested by the results in three example applications, our generic approach can outperform or on par with domain-specific algorithms. In this paper, we have focused on designing a general framework and have not tried to maximize the contribution of application-specific components (except for the face super-resolution experiments). Although the resulting systems already demonstrated state-of-the-art performance in a variety of different enhancement tasks, we expect that the enhancement quality could be even further improved by resorting to more advanced image representations or preprocessing. Future work should explore this direction.

Nonetheless, our framework provides interesting conceptual insights, allows for high-quality image enhancement in different scenarios, and allows us to customize the degradation models very efficiently since training times are orders of magnitude faster than related previous methods from the literature. Our framework is equally applicable to other computer vision problems such as video super-resolution, and image and video enhancement under Gaussian and non-Gaussian noise.

In the future, we plan to investigate the enhancement of compressed video (e.g. H.264/MPEG-4) and the enhancement of images in various (non-Gaussian) noise conditions.

## A Details of the face image super-resolution experiments

For face image super-resolution experiments, we used FERET database [46]. Experimental scenario is that each input image contains a face which is roughly of size  $25 \times 25$ . Then, the objective is to super-resolve the facial part of the image 4 times along each dimension such that the resulting super-resolved face image is of size  $100 \times 100$ .

For this task, we exploited the fact that when the face images are well-aligned, each pixel of interest mostly shows (part of) specific objects. For instance, at the left-top corner of the face image, the regressor can focus on reconstructing the left eye and disregard e.g. mouth. To facilitate this scenario, we performed automatic face detection and facial component detection on the database using the algorithms of [47] and [48], respectively. This produced 705 face images (see [48] for details). Based on the detected eye corner locations, the input image is scaled such that the  $x$ -coordinate value difference between the outer corners of two eyes becomes 50 (Fig. 19).

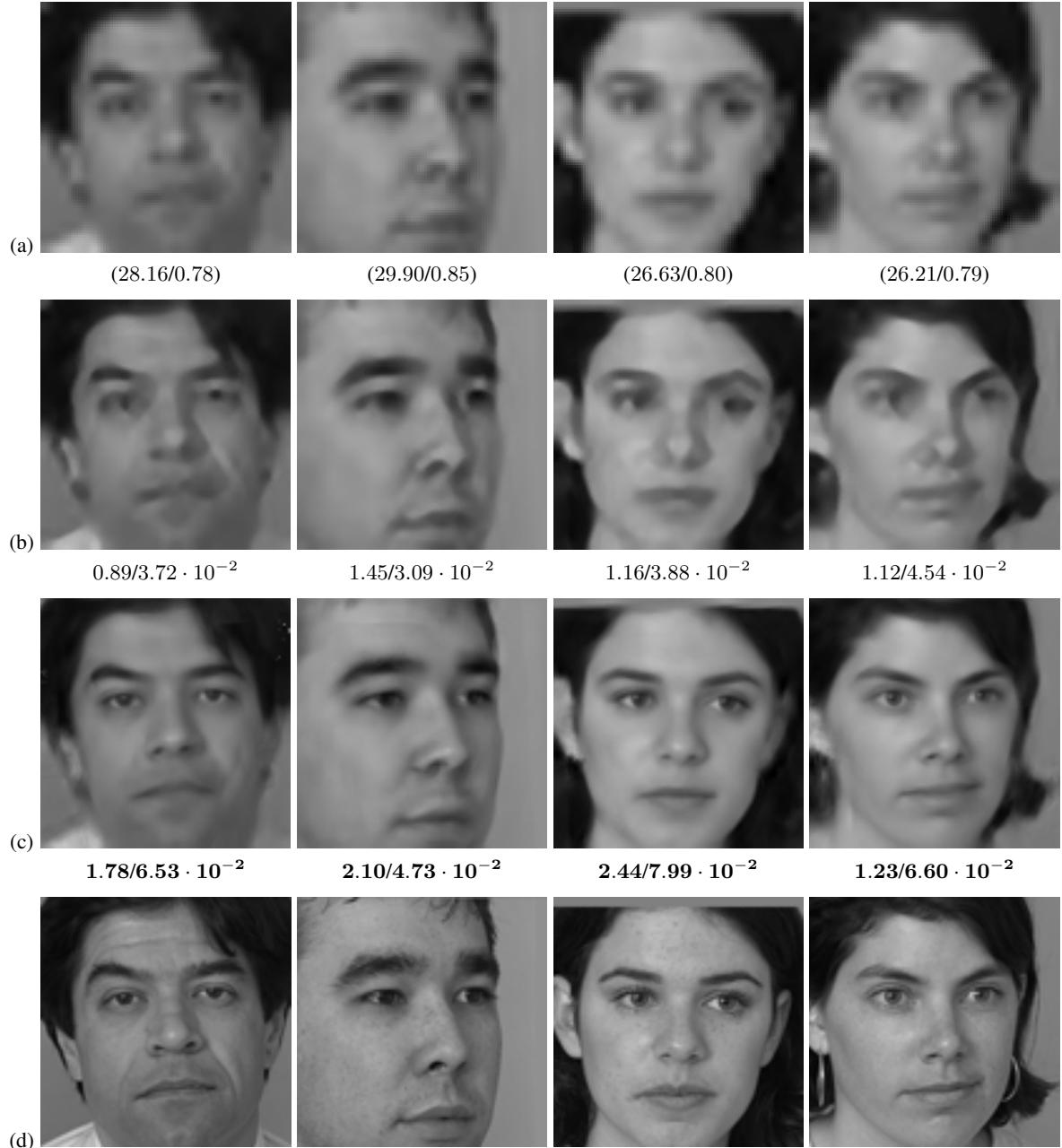


Figure 18: Examples of super-resolution for face images (mag. factor 4). (a) bicubic resampling, (b) Kim and Kwon [9], (c) our method trained based on a face image DB, and (d) ground truth high-resolution images.

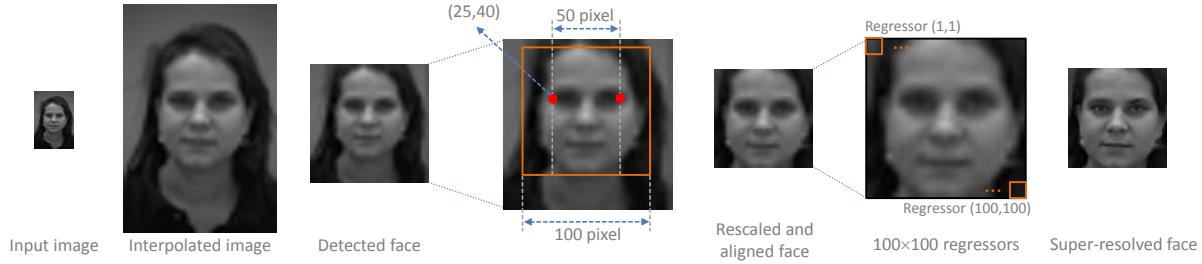


Figure 19: Schematic diagram of face image super-resolution.

Then, the detected face image is aligned into a  $100 \times 100$ -sized grid such that the outer corner of the left eye is located at  $(25, 40)$  in this grid. In this way, the face images are roughly scale-normalized and aligned.

For each pixel location in a face image, we construct a GP regressor which is trained for reconstructing the images at that specific location. Since in general the facial component detection is erroneous (especially for low-resolution input faces), we explicitly made the regressor tolerant against small detection misplacements by sampling the training patch for the  $(i, j)$ -th regressor not only at  $(i, j)$  but also at its eight spatial neighbors. The parameters for regressors are adopted from the GP which was trained for the generic image database.

The experiments were performed in leave-one-person-out manner: For each face-containing test image, the images containing the same person of interest are removed from the training set and the regressors are subsequently trained. The low-resolution image is firstly 4-times-magnified along each dimension based on bicubic resampling on which the face detector and facial component detector is applied and the resulting face location is cropped, re-scaled, and aligned in the same way as for the training data. The detection of facial component in low-resolution interpolated image is facilitated by exploiting a strong prior on the joint location of several facial components [48]. Accordingly, even though we are using only the outer corner locations of eyes, we detect jointly the inner corners of eyes, mouth corners, and the nose. For quantitative evaluation, we prepared high-resolution ground truth images which are re-sized and aligned based on the object locations identified at low-resolution input images.

## References

- [1] P. V. Gehler and M. Welling. Product of “edge-perts”. In *Advances in Neural Information Processing Systems*, Cambridge, MA, 2005. MIT Press.
- [2] D. Sun and W.-K. Cham. Postprocessing of low bit-rate block DCT coded images based on a fields of experts prior. *IEEE Trans. Image Processing*, 16(11):2743–2751, 2007.
- [3] S. Roth and M. J. Black. Fields of experts. *International Journal of Computer Vision*, 82(2):205–229, 2009.
- [4] U. Schmidt, Q. Gao, and S. Roth. A generative perspective on MRFs in low-level vision. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 1751–1758, Cambridge, MA, 2010.
- [5] J. Portilla, V. Strela, M. Wainwright, and E. P. Simoncelli. Image denoising using scale mixtures of Gaussians in the wavelet domain. *IEEE Trans. Image Processing*, 12(11):1338–1351, 2003.
- [6] W. T. Freeman, E. C. Pasztor, and O. T. Carmichael. Learning low-level vision. *International Journal of Computer Vision*, 40(1):25–47, 2000.
- [7] E. Snelson and Z. Ghahramani. Sparse Gaussian processes using pseudo-inputs. In *Advances in Neural Information Processing Systems*, Cambridge, MA, 2006. MIT Press.
- [8] C. Walder, K. I. Kim, and B. Schölkopf. Sparse multiscale Gaussian process regression. In *Proc. International Conference on Machine Learning*, pages 1112–1119, 2008.
- [9] K. I. Kim and Y. Kwon. Single-image super-resolution using sparse regression and natural image prior. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 32(6):1127–1133, 2010.
- [10] J. Quiñonero-Candela and C. E. Rasmussen. A unifying view of sparse approximate Gaussian process regression. *Journal of Machine Learning Research*, 6:1939–1959, 2005.
- [11] A. Skodras, C. Christopoulos, and T. Ebrahimi. The JPEG 2000 still image compression standard. *IEEE Signal Processing Magazine*, 18(5):36–58, 2001.

- [12] M. Rabbani and R. Joshi. An overview of the JPEG 2000 still image compression standard. *Signal Processing: Image Communication*, 17(1):3–48, 2002.
- [13] M. A. Robertson and R. L. Stevenson. DCT quantization noise in compressed images. *IEEE Trans. Circuits and Systems for Video Technology*, 15(1):27–38, 2005.
- [14] D. Tschumperlé and R. Deriche. Vector-valued image regularization with PDEs: a common framework for different applications. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 27(4):506–517, 2005.
- [15] T. Wang and G. Zhai. JPEG2000 image postprocessing with novel trilateral deringing filter. *Optical Engineering*, 47(2):027005–1–027005–6, 2008.
- [16] 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*, 20(8):595–607, 2009.
- [17] Y. Yang, N. P. Galatsanos, and A. K. Katsaggelos. Projection-based spatially adaptive reconstruction of block-transform compressed images. *IEEE Trans. Image Processing*, 4(7):896–908, 1995.
- [18] X. Li. Improved wavelet decoding via set theoretic estimation. *IEEE Trans. Circuits and Systems for Video Technology*, 15(1):108–112, 2005.
- [19] F. Alter, S. Durand, and J. Froment. Adapted total variation for artifact free decompression of JPEG images. *Journal of Mathematical Imaging and Vision*, 23:199–211, 2005.
- [20] A. Nosratinia. Denoising of JPEG images by re-application of JPEG. *Journal of VLSI Signal Processing*, 27(1):69–79, 2001.
- [21] A. Nosratinia. Postprocessing of JPEG-2000 images to remove compression artifacts. *IEEE Signal Processing Letters*, 10(10):296–299, 2003.
- [22] G. Qiu. MLP for adaptive postprocessing block-coded images. *IEEE Trans. Circuits and Systems for Video Technology*, 10(8):1450–1454, 2000.
- [23] K. Lee, D. S. Kim, and T. Kim. Regression-based prediction for blocking artifact reduction in JPEG-compressed images. *IEEE Trans. Image Processing*, 14(1):36–49, 2005.
- [24] V. Laparra, J. Gutiérrez, G. Camps-Valls, and J. Malo. Image denoising with kernels based on natural image relations. *Journal of Machine Learning Research*, 11:873–903, 2010.
- [25] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Trans. Image Processing*, 13(4):600–612, 2004.
- [26] Z. Lin and H.-Y. Shum. Fundamental limits of reconstruction-based superresolution algorithms under local translation. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 26(1):83–97, 2004.
- [27] S. Baker and T. Kanade. Limits on super-resolution and how to break them. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 24(9):1167–1183, 2002.
- [28] S. Chaudhuri. *Super-Resolution Imaging*. Springer, 2001.
- [29] W. T. Freeman, T. R. Jones, and E. C. Pasztor. Example-based super-resolution. *IEEE Computer Graphics and Applications*, 22(2):56–65, 2002.
- [30] A. Hertzmann, C. E. Jacobs, N. Oliver, B. Curless, and D. H. Salesin. Image analogies. In *Computer Graphics (Proc. SIGGRAPH)*, pages 327–340, NY, 2001. ACM Press.
- [31] L. C. Pickup, S. J. Roberts, and A. Zissermann. A sampled texture prior for image super-resolution. In S. Thrun, L. Saul, and B. Schölkopf, editors, *Advances in Neural Information Processing Systems*, Cambridge, MA, 2004. MIT Press.
- [32] H. Chang, D.-Y. Yeung, and Y. Xiong. Super-resolution through neighbor embedding. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 275–282, 2004.
- [33] 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.
- [34] J. Yang, J. Wright, T. S. Huang, and Y. Ma. Image super-resolution via sparse representation. *IEEE Trans. Image Processing*, 19(11):2861–2873, 2010.
- [35] C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006.

- [36] 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.
- [37] E. Snelson and Z. Ghahramani. Local and global sparse Gaussian process approximations. In *Proc. Artificial Intelligence and Statistics*, 2007.
- [38] J. M. Lee. *Introduction to Smooth Manifolds*. Springer, 2002.
- [39] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.
- [40] B. W. Silverman. Spline smoothing: the equivalent variable kernel method. *The Annals of Statistics*, 12(3):898–916, 1984.
- [41] P. Sollich and C. K. I. Williams. Using the equivalent kernel to understand Gaussian process regression. In *Advances in Neural Information Processing Systems*, Cambridge, MA, 2005. MIT Press.
- [42] N. Lawrence, M. Seeger, and R. Herbrich. Fast sparse Gaussian process methods: the informative vector machine. In *Advances in Neural Information Processing Systems*, pages 625–632, Cambridge, MA, 2003. MIT Press.
- [43] 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*, pages 56–77, 2006.
- [44] A. Foi, V. Katkovnik, and K. Egiazarian. Pointwise shape-adaptive DCT for high-quality denoising and deblocking of grayscale and color images. *IEEE Trans. Image Processing*, 16(5):1395–1411, 2007.
- [45] R. Fattal. Image upsampling via imposed edge statistics. *ACM Trans. Graphics (Proc. SIGGRAPH)*, 26(3):95:1–95:8, 2007.
- [46] P. J. Phillips, H. Moon, P. J. Rauss, and S. Rizvi. The FERET evaluation methodology for face recognition algorithms. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 22(10):1090–1104, 2000.
- [47] W. Kienzle, G. Bakir, M. Franz, and B. Schölkopf. Face detection - efficient and rank deficient. In *Advances in Neural Information Processing Systems*, pages 673–680, Cambridge, MA, 2005. MIT Press.
- [48] P. Breuer, K. I. Kim, W. Kienzle, B. Schölkopf, and V. Blanz. Automatic 3D face reconstruction from single images or video. In *Proc. International Conference on Automatic Face and Gesture Recognition*, pages 1–8, 2008.

Below you find a list of the most recent research reports of the Max-Planck-Institut für Informatik. Most of them are accessible via WWW using the URL <http://www.mpi-inf.mpg.de/reports>. Paper copies (which are not necessarily free of charge) can be ordered either by regular mail or by e-mail at the address below.

Max-Planck-Institut für Informatik  
– Library and Publications –  
Campus E 1 4

D-66123 Saarbrücken

E-mail: [library@mpi-inf.mpg.de](mailto:library@mpi-inf.mpg.de)

---

MPI-I-2009-RG1-002	P. Wischnewski, C. Weidenbach	Contextual rewriting
MPI-I-2009-5-006	S. Bedathur, K. Berberich, J. Dittrich, N. Mamoulis, G. Weikum	Scalable phrase mining for ad-hoc text analytics
MPI-I-2009-5-004	N. Preda, F.M. Suchanek, G. Kasneci, T. Neumann, G. Weikum	Coupling knowledge bases and web services for active knowledge
MPI-I-2009-5-003	T. Neumann, G. Weikum	The RDF-3X engine for scalable management of RDF data
MPI-I-2008-RG1-001	A. Fietzke, C. Weidenbach	Labelled splitting
MPI-I-2008-5-004	F. Suchanek, M. Sozio, G. Weikum	SOFI: a self-organizing framework for information extraction
MPI-I-2008-5-003	F.M. Suchanek, G. de Melo, A. Pease	Integrating Yago into the suggested upper merged ontology
MPI-I-2008-5-002	T. Neumann, G. Moerkotte	Single phase construction of optimal DAG-structured QEPs
MPI-I-2008-5-001	F. Suchanek, G. Kasneci, M. Ramanath, M. Sozio, G. Weikum	STAR: Steiner tree approximation in relationship-graphs
MPI-I-2008-4-003	T. Schultz, H. Theisel, H. Seidel	Crease surfaces: from theory to extraction and application to diffusion tensor MRI
MPI-I-2008-4-002	W. Saleem, D. Wang, A. Belyaev, H. Seidel	Estimating complexity of 3D shapes using view similarity
MPI-I-2008-1-001	D. Ajwani, I. Malinger, U. Meyer, S. Toledo	Characterizing the performance of Flash memory storage devices and its impact on algorithm design
MPI-I-2007-RG1-002	T. Hillenbrand, C. Weidenbach	Superposition for finite domains
MPI-I-2007-5-003	F.M. Suchanek, G. Kasneci, G. Weikum	Yago : a large ontology from Wikipedia and WordNet
MPI-I-2007-5-002	K. Berberich, S. Bedathur, T. Neumann, G. Weikum	A time machine for text search
MPI-I-2007-5-001	G. Kasneci, F.M. Suchanek, G. Ifrim, M. Ramanath, G. Weikum	NAGA: searching and ranking knowledge
MPI-I-2007-4-008	J. Gall, T. Brox, B. Rosenhahn, H. Seidel	Global stochastic optimization for robust and accurate human motion capture
MPI-I-2007-4-007	R. Herzog, V. Havran, K. Myszkowski, H. Seidel	Global illumination using photon ray splatting
MPI-I-2007-4-006	C. Dyken, G. Ziegler, C. Theobalt, H. Seidel	GPU marching cubes on shader model 3.0 and 4.0
MPI-I-2007-4-005	T. Schultz, J. Weickert, H. Seidel	A higher-order structure tensor
MPI-I-2007-4-004	C. Stoll, E. de Aguiar, C. Theobalt, H. Seidel	A volumetric approach to interactive shape editing
MPI-I-2007-4-003	R. Bargmann, V. Blanz, H. Seidel	A nonlinear viseme model for triphone-based speech synthesis
MPI-I-2007-4-002	T. Langer, H. Seidel	Construction of smooth maps with mean value coordinates
MPI-I-2007-4-001	J. Gall, B. Rosenhahn, H. Seidel	Clustered stochastic optimization for object recognition and pose estimation

MPI-I-2007-2-001	A. Podelski, S. Wagner	A method and a tool for automatic verification of region stability for hybrid systems
MPI-I-2007-1-003	A. Gidenstam, M. Papatriantafilou	LFthreads: a lock-free thread library
MPI-I-2007-1-002	E. Althaus, S. Canzar	A Lagrangian relaxation approach for the multiple sequence alignment problem
MPI-I-2007-1-001	E. Berberich, L. Kettner	Linear-time reordering in a sweep-line algorithm for algebraic curves intersecting in a common point
MPI-I-2006-5-006	G. Kasneci, F.M. Suchanek, G. Weikum	Yago - a core of semantic knowledge
MPI-I-2006-5-005	R. Angelova, S. Siersdorfer	A neighborhood-based approach for clustering of linked document collections
MPI-I-2006-5-004	F. Suchanek, G. Ifrim, G. Weikum	Combining linguistic and statistical analysis to extract relations from web documents
MPI-I-2006-5-003	V. Scholz, M. Magnor	Garment texture editing in monocular video sequences based on color-coded printing patterns
MPI-I-2006-5-002	H. Bast, D. Majumdar, R. Schenkel, M. Theobald, G. Weikum	IO-Top-k: index-access optimized top-k query processing
MPI-I-2006-5-001	M. Bender, S. Michel, G. Weikum, P. Triantafilou	Overlap-aware global df estimation in distributed information retrieval systems
MPI-I-2006-4-010	A. Belyaev, T. Langer, H. Seidel	Mean value coordinates for arbitrary spherical polygons and polyhedra in $\mathbb{R}^3$
MPI-I-2006-4-009	J. Gall, J. Potthoff, B. Rosenhahn, C. Schnoerr, H. Seidel	Interacting and annealing particle filters: mathematics and a recipe for applications
MPI-I-2006-4-008	I. Albrecht, M. Kipp, M. Neff, H. Seidel	Gesture modeling and animation by imitation
MPI-I-2006-4-007	O. Schall, A. Belyaev, H. Seidel	Feature-preserving non-local denoising of static and time-varying range data
MPI-I-2006-4-006	C. Theobalt, N. Ahmed, H. Lensch, M. Magnor, H. Seidel	Enhanced dynamic reflectometry for relightable free-viewpoint video
MPI-I-2006-4-005	A. Belyaev, H. Seidel, S. Yoshizawa	Skeleton-driven laplacian mesh deformations
MPI-I-2006-4-004	V. Havran, R. Herzog, H. Seidel	On fast construction of spatial hierarchies for ray tracing
MPI-I-2006-4-003	E. de Aguiar, R. Zayer, C. Theobalt, M. Magnor, H. Seidel	A framework for natural animation of digitized models
MPI-I-2006-4-002	G. Ziegler, A. Tevs, C. Theobalt, H. Seidel	GPU point list generation through histogram pyramids
MPI-I-2006-4-001	A. Efremov, R. Mantiuk, K. Myszkowski, H. Seidel	Design and evaluation of backward compatible high dynamic range video compression
MPI-I-2006-2-001	T. Wies, V. Kuncak, K. Zee, A. Podelski, M. Rinard	On verifying complex properties using symbolic shape analysis
MPI-I-2006-1-007	H. Bast, I. Weber, C.W. Mortensen	Output-sensitive autocomplete search
MPI-I-2006-1-006	M. Kerber	Division-free computation of subresultants using bezout matrices
MPI-I-2006-1-005	A. Eigenwillig, L. Kettner, N. Wolpert	Snap rounding of Bézier curves
MPI-I-2006-1-004	S. Funke, S. Laue, R. Naujoks, L. Zvi	Power assignment problems in wireless communication
MPI-I-2005-5-002	S. Siersdorfer, G. Weikum	Automated retraining methods for document classification and their parameter tuning
MPI-I-2005-4-006	C. Fuchs, M. Goesele, T. Chen, H. Seidel	An empirical model for heterogeneous translucent objects
MPI-I-2005-4-005	G. Krawczyk, M. Goesele, H. Seidel	Photometric calibration of high dynamic range cameras
MPI-I-2005-4-004	C. Theobalt, N. Ahmed, E. De Aguiar, G. Ziegler, H. Lensch, M.A. Magnor, H. Seidel	Joint motion and reflectance capture for creating relightable 3D videos
MPI-I-2005-4-003	T. Langer, A.G. Belyaev, H. Seidel	Analysis and design of discrete normals and curvatures
MPI-I-2005-4-002	O. Schall, A. Belyaev, H. Seidel	Sparse meshing of uncertain and noisy surface scattered data