Non-parametric Blur Map Regression for Depth of Field Extension

Laurent D'Andrès, Jordi Salvador, *Member, IEEE* Axel Kochale, *Member, IEEE* and Sabine Süsstrunk, *Senior Member, IEEE*

***Abstract*—Real camera systems have a limited depth of field (DOF) which may cause an image to be degraded due to visible misfocus or too shallow DOF. In this paper, we present a blind deblurring pipeline able to restore such images by slightly extending their DOF and recovering sharpness in regions slightly out-of-focus. To address this severely ill-posed problem, our algorithm relies first on the estimation of the spatially-varying defocus blur. Drawing on local frequency image features, a machine learning approach based on the recently introduced Regression Tree Fields is used to train a model able to regress a coherent defocus blur map of the image, labeling each pixel by the scale of a defocus point-spread-function. A non-blind spatially-varying deblurring algorithm is then used to properly extend the DOF of the image. The good performance of our algorithm is assessed both quantitatively, using realistic ground truth data obtained with a novel approach based on a plenoptic camera, and qualitatively with real images.**

***Index Terms*—Out-of-focus deblurring, extension of depth of field, Regression tree fields, defocus blur map.**

I. Introduction

THE usage of large sensors in compact camera designs for acquiring high resolution images and videos has direct consequences on the depth of field (DOF) that can be captured. DOF refers to the distance around the image plane for which the camera is focused and for which the objects in the scene appear acceptably sharp in the resulting image. For a given sensor size, the DOF is influenced by the focal length of the lens, the distance of the object the camera is focused on, and the aperture (f-number).

DOF plays a key role in selecting relevant scene information to be conveyed by the image. To direct viewer attention and emphasize the main subject, shallow DOF is often used by allowing the foreground and background to be blurry. While a limited DOF is desirable for aesthetic reasons, it is also an important source of image degradation. Indeed, such camera settings are more difficult to control and easily produce images for which a small underlying defocus blur affects even the main subject due to, e.g., a too shallow DOF or a misfocus. This might further affect the performance of many image processing and computer vision algorithms that do not explicitly model this degradation.

This research was conducted as L. D'Andrès was an intern at Technicolor, Hannover, Germany and a student at EPFL, Switzerland. Mail contact: laurent.dandres@alumni.epfl.ch

J. Salvador and A. Kochale are with Technicolor R&I, Hannover, Germany. {jordi.salvador,axel.kochale}@technicolor.com

S. Süsstrunk is with the School of Computer and Communication Sciences (IC), EPFL, Switzerland. sabine.susstrunk@epfl.ch

To restore such corrupted images, it is desirable to slightly extend their DOF in order to recover sharpness in slightly out-of-focus areas. The underlying inverse problem, known as blind deblurring, looks to estimate and remove the effects of the undesired defocus blur. To account for its spatially-varying nature, the estimation of defocus blur can generally be cast as a blur map estimation [1]–[4], for which the scale parameter of a priori known defocus point-spread-functions (PSF) model (disc, Gaussian) needs to be specified at each pixel. This has proven hard to solve accurately, and, to this date, most successful solutions for out-of-focus restoration are thus techniques for which the blur kernels estimation is either simplified or circumvented thanks to alterations in the optical design (coded aperture [5], chromatic aberrations [6]) or the presence of correctly focused images of the same scene [7]. This contrasts to camera shake deblurring, for which a broad range of methods [8]–[16] effectively work based on a single image recorded with a conventional camera.

In this paper, we introduce a blind deblurring pipeline, similar to the one introduced by Couzinie *et al.* [2], for the restoration of images subject to a too shallow DOF and recorded with a conventional camera. Using a disc PSF model, our algorithm first estimates a defocus blur map of the image by inferring the appropriate radius of the disc PSF at each pixel. To that end, we propose a learning-based approach based on regression tree fields (RTF) [17] and improve upon previous blur map estimations based on manually defined color constraints [1]–[4] by proposing a non-parametric alternative for which smoothness constraints are directly learned from data. Learning an effective discriminative model, such as RTF, that can provide good generalization without requiring an overly large amount of training data is, however, a very challenging task. In our first and main contribution, we show that this problem can be overcome by training a model cascade of RTFs [18] built on strong blur features extracted from local frequency image statistics [3], [19]. Once a blur map is known, a non-blind deconvolution algorithm is used to properly restore the image. Based on sparse derivative priors, our algorithm can not only deblur the entire image [2] (with similar limitations as other existing approaches as blur or noise get larger) but is also parametrizable towards slight extension of DOF, *i.e.* to deblur only the parts of the image under a certain level of blur while leaving strongly out-of-focus areas intact (see Fig. 1).

In order to carry out a meaningful quantitative evaluation of our deblurring framework, another important contribution of our work is to provide, based on a plenoptic camera, a novel approach to remedy the lack of realistic ground truth

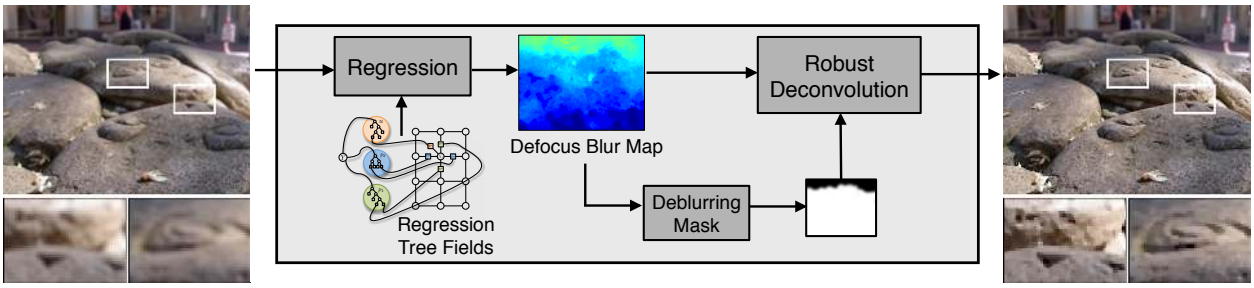


Fig. 1: Blind deblurring pipeline for depth of field extension and example of results.

data in spatially-varying out-of-focus problems (Section V). Using a synthetic yet realistic dataset acquired with a Lytro camera, along with a small set of real images, quantitative and qualitative experimental results show that our approach yields state-of-the-art performance at restoring spatially-varying defocus blur from a single image.

II. Related work

Blind deblurring has proven hard to solve [20]. Mathematically, an image subject to any type of blur is commonly modeled as

$$y[i] = (k_i * x)[i] + n[i] = \sum_m k_i[i-m]x[m] + n[i]\ , \quad (1)$$

where y[1] is the blurred image, x is a latent all-sharp image, $k_i[m]$ is the local blur kernel (or PSF) at pixel i and n is additive noise. For an ideal lens with a circular aperture, the defocus PSFs can be modeled by the disc function [21], or by a Gaussian when diffraction becomes significant for mild defocus [22]. In this work, we discard the effect of diffraction and use disc PSFs exclusively, but our approach is not limited to this setup. Because one has considerably more unknowns (x, $\{k_i\}_i$ and n) than observations (y), recovering x is a severely ill-posed problem that is generally best solved by first estimating the most likely blur kernel(s) under a given distribution of sharp natural images before applying a non-blind deconvolution to recover the sharp image x [20].

In case of defocus blur, the depth dependency of the PSFs, and hence the spatially-varying nature of the blur, makes the sub-problem of estimating the blur kernels very challenging. This problem has been extensively tackled in the context of recovering 3D from 2D, for which defocus cues are used to estimate the depth map of a scene (depth from defocus [23]–[28] or depth from focus [29]–[31]). These methods however require several input images with known focus settings and contrast with our target of estimating a defocus blur map from a single image. Methods have been proposed to estimate spatially-varying defocus PSFs at the edges from a single image, by measuring the effect of a Gaussian PSF on a step edge [1], [4], [32] or by predicting sharp edges [33]. In [1], [4], [32], full blur maps are obtained via interpolation based on color similarity constraints. Our method improves on these by making a dense estimation for the entire image and not being limited to a Gaussian PSF. Probability-based methods [2], [3] have also been proposed, in which local frequency image features are used to model the likelihood of a given PSF at any pixel. A coherent defocus blur map is then estimated via the combination of likelihood estimates and an energy minimization framework based on manually defined color constraints. By drawing on RTFs, we improve on these by discriminatively learning these constraints directly in the model. In Shi *et al.* [34], finally, a dictionary learned on sharp and slightly blur patches is used to decompose local image patches into set of atoms. Sparsity blur features are then proposed to detect small Gaussian PSFs at any pixel (or what is called just noticeable blur in [34]), based on empirical evidence that a strong correlation exists between the number of dictionary atoms and the amount of Gaussian blur.

Once the blur parameters have been estimated, the non-blind deconvolution step is also ill-posed in presence of noise. To retrieve sharp images with little artifacts, sparse derivatives [5], [35] and learning-based priors [18], [36] have been proposed as regularization on sharp natural images. While an adaptation of [18] may offer slightly superior results, the primary motivation of our work is to provide a blur map estimation accurate enough towards deblurring. A more conventional regularization based on sparse derivatives is thus used.

III. Defocus blur map estimation

A. Localized 2D frequency analysis

The analysis of the frequency spectrum of an image is a natural cue for retrieving information about the blur. As shown by Chakrabarti *et al.* [19] (motion blur) and Zhu *et al.* [3] (defocus blur), spatially-varying blur with a priori known PSF models can be well analyzed by means of local frequency component analysis. Their model allows to compute the likelihood of a small image window being blurred by a given PSF and will be the start of our approach.

Let us assume that the blur kernels $k_i[m]$ are constant in any small window η of size $W \times W$. The local frequency spectrum of an image can be approximated in such windows as the responses of the image y to a set of M filters $\{f_m\}_m$ (same size as the analysis window η) of different well-chosen

[1]Note that we remove a standard gamma correction of 2.2 before any blur-related processing (deblurring, ground truth data generation) in order to recover a physically coherent model of blur with linear intensities and take advantage of the YCbCr color space to model and deblur the luminance channel Y only, as the human visual system is not sensitive to high frequencies in color differences.

spatial frequencies. Concretely, a set of Gabor filters of the form

$$f_m[i] = w[i] \exp\left(-2\pi j \left(i_1 \omega_1^{(m)} + i_2 \omega_2^{(m)}\right)\right) \quad (2)$$

is generally used to represent those filters, where the pairs $\{(\omega_1^{(m)}, \omega_2^{(m)})\}_m$ encode the spatial frequencies, $w[i]$ is a Gaussian window aligned with the analysis window and $i = (i_1, i_2)$.

If the underlying gradient distribution of the latent sharp image x is locally white Gaussian with (unknown) variance s_i in a given window, it can be shown [19] that the responses $\{y_m^{\nabla}[i]\}_m$ of the image derivative y^{∇} to the set of filters $\{f_m\}_m$ can be used to determine the likelihood of a specific blur kernel k as

$$p\left(\{|y_m^{\nabla}[i]|^2\}_m | k, s_i\right) = \prod_m \mathbf{Exp}\left(|y_m^{\nabla}[i]|^2; 1/\left(s_i \sigma_{km}^2 + \sigma_{nm}^2\right)\right), \quad (3)$$

where **Exp** is the exponential distribution, and following [19], $\{\sigma_{km}^2\}_m$ and $\{\sigma_{nm}^2\}_m$ are, respectively, called the blur spectrum and the noise spectrum, defined by

$$\sigma_{km}^2 = \sum_i |(k * f_m)[i]|^2 \,, \; \sigma_{nm}^2 = \sigma_n^2 \sum_i |(\nabla * f_m)[i]|^2 \,. \quad (4)$$

Using this model, a defocus blur map b can be estimated by performing a maximum likelihood estimation over a set of different defocus PSFs for each pixel[2] (ML blur map). A ML decision is, however, prone to errors (Fig. 3 (c)) because: (i) conventional camera PSFs (*e.g.* disc function) partially share similar frequency responses at different scales [5] (ii) the inherent trade-off between spatial and spectral resolution limits the ability of the model to get localized boundaries (especially when W gets large). To estimate a coherent defocus blur map, a model which incorporates appropriate constraints between neighboring pixels is therefore needed. Existing approaches [2], [3] relied on color constraints to get better blur boundaries estimation (blur discontinuities are caused by depth discontinuities which generally align with color discontinuities), but this typically fails in absence of colors difference or with gradually changing blur. In this paper, a learning-based alternative drawing on regression tree fields (RTF) is therefore proposed to alleviate their need.

B. Regression tree fields

Introduced by Jancsary *et al.* [17], RTF belong to the family of graphical models designed to solve image labeling problems, where one is given an observed image y and wishes to predict a labeled image b in a globally consistent way (the labeled image is denoted by b to purposely match the notation of a defocus blur map). From a high-level perspective, RTFs consist of a simple Gaussian Conditional Random Field (CRF) whose corresponding density $p(b|y) \propto e^{-E(b|y)}$ is completely specified by a quadratic energy of the form

$$E(b|y) = \frac{1}{2} b^T \Theta(y) b - b^T \theta(y) \,, \quad (5)$$

[2]This in particular requires the estimation of s_i (besides σ_n^2 for which several methods exist). An heuristic approach is to independently select the optimal s^* that maximizes (3) for each window. See [3], [19] for details.

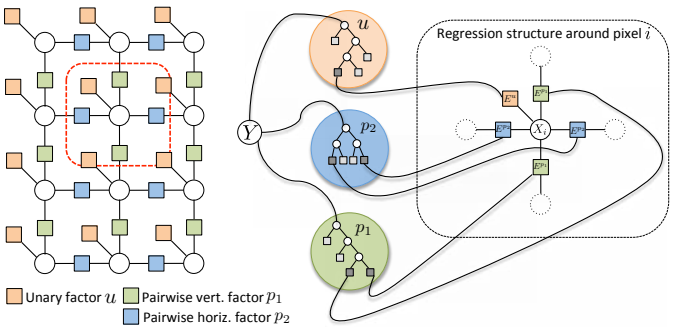


Fig. 2: Parametrization of a RTF model. Left, the representation of a Gaussian CRF via a factor graph, for which local interactions have been grouped into three factor types (one unary and two pairwise). Right, an illustration of how regression trees are associated to the CRF to assign local model parameters based on the local image content.

and for which the model parameters $\Theta(y)$ and $\theta(y)$ are regressed from the observed image y. The power of the model lies in its parametrization via regression trees, which assign local model parameters depending on the image content and render the approach non-parametric. Concretely, the global energy $E(b|y)$ is decomposed into local potentials relating only one or two pixels, which are then regrouped into common factors type $f \in F$ sharing the same local parameters Θ^f and θ^f

$$E(b|y) = \sum_f \sum_{p \in P^f} E^f(b_p|y)$$
$$E^f(b_p|y) = \frac{1}{2} b_p^T \Theta_p^f(y) b_p - b_p^T \theta_p^f(y) \,, \quad (6)$$

where P^f is the set of pixels or pairs of pixels related to a given factor type.

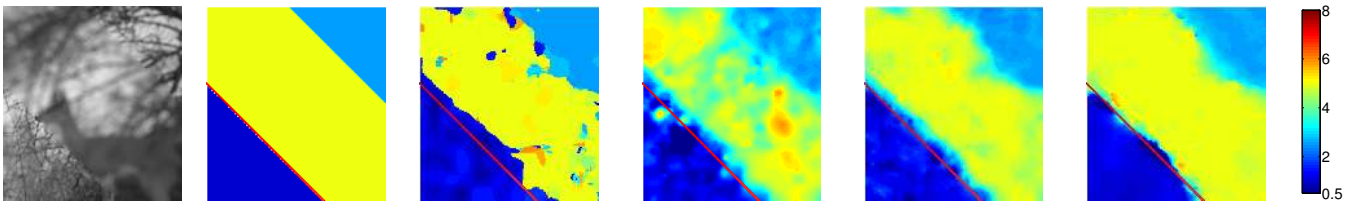
As shown in Fig. 2, small connected neighborhoods around each pixel are used to instantiate in a repetitive manner the different factor types considered in the model, *e.g.* $|F| = 3$ for a 4-connected neighborhood (one unary factor and, due to spatial symmetries, 2 pairwise factors). Each factor type is associated to a regression tree, whose leaves store different parametrizations of the local potentials. The image content around each factor instantiation determines the actual leaf that is attained and hence the local Gaussian model in effect. Once the global energy has been shaped by the sum of all the local potentials over the entire image, the prediction b^* is given by the mode of the Gaussian density $p(b|y)$

$$b^* = [\Theta(y)]^{-1} \theta(y) \,, \quad (7)$$

which can be obtained by solving a system of linear equations for which efficient methods exist.

C. RTF-based blur map estimation

We want to use RTF to regress a coherent defocus blur map b, whose entry $b[i]$ contains the radius parameter r_i of a disc PSF that matches the local PSF $k_i[m]$ at pixel i in (1). While the RTF parametrization is powerful, the success of its application relies, however, on the ability to learn effective regression tree structures (*e.g.* , split functions, linear regressors in the leaves) specific to our given image problem.



(a) Blurred (b) Ground Truth (c) ML Map MAE/MSE: 0.44/1.28 (d) RTF_0 MAE/MSE: 0.60/0.83 (e) RTF_1 MAE/MSE: 0.39/0.52 (f) RTF_3 MAE/MSE: 0.22/0.29

Fig. 3: Comparison of blur map inference on a particular example of our validation set. RTF_0 is trained with (normalized) likelihoods and Gabor filters as features, RTF_1 with sorted radii (see main text) and RTF_3 extends RTF_1 to the cascade model with three layers.

This not only requires a sufficient amount of training data for which the ground truth labeled image is available, but also the capacity of extracting meaningful image features for which effective split functions can be learned.

Training data. We generate ground truth training data by synthetically blurring a set of sharp natural images with various blur patterns, orientations and blur scales r. Concretely, we have created synthetic random blur patterns composed by a combination of two different types of regions: (i) gradually changing blur regions, consisting in a gradient between two radii; or (ii) uniform blur regions (constant radius) delimited by sharp transitions that simulate depth discontinuities (*e.g.* see Fig. 3 (b)). Parametrized by radii of our synthetic disc PSF model in the interval $[0.5, 5]$ pixels, these blur patterns serve as ground truth blur maps for training. We use the Berkeley segmentation dataset [37] to extract crops of all-sharp natural images which are then synthetically blurred with our ground truth blur maps. This approach is simplistic because our blur patterns do not model the natural variation of depth in the images (*e.g.* blur discontinuities do not align with color discontinuities). It has, however, several advantages towards generalization, because it allows to more easily control the number of blur discontinuities, the orientations of the blur patterns, the diversity of the blur scales, and finally the content of the image patches (only regions with enough texture are considered). This results in a carefully designed training dataset consisting of 100 images of size 240×240. We additionally gather 20 more images for validation.

Model selection. The high computational cost of RTF training renders the use of validation methods robust against over-fitting prohibitive and parameters have been therefore empirically set. Specifically, we model connections in a 5×5 neighborhood (*i.e.* 12 different pairwise factors, as well as one unary factor) and use regression trees of depth 10 for both unary and pairwise factors. We moreover use probabilistic training based on the maximization of the pseudolikelihood [17] to learn the tree structures.

Features design and cascading. Features are N-dimensional vectors where each dimension represents a different type of information about a given pixel (*e.g.* , filter responses). In order to build effective regression trees which can moreover be learned on a limited amount of training data, it is of prime importance to extract features expressive enough towards blur radius discrimination. To that end, our approach is based on the likelihood model developed above (III-A) to extract local spectral blur cues. Specifically, we estimate the likelihoods of a set of defocus disc PSFs with blur radii in $R = \{0.5, 0.75, 1, 1.25, ..., 6\}$ at each pixel i, based on Eq. (3) (using a sliding window $W = 41$)[3]. We then propose to compute feature vectors Φ_i by sorting the blur radii based on their likelihood

$$\Phi_i = \begin{bmatrix} r_i^{(1)} \\ r_i^{(2)} \\ \vdots \\ r_i^{(j)} \end{bmatrix}, j = 1, 2, \ldots, 23 \tag{8}$$

where $r_i^{(j)}$ is the radius with the j-th highest likelihood computed at pixel i.

We briefly justify this proposed scheme, RTF_1, by comparing it to a more straightforward model trained with plain likelihoods and 60 Gabor filter responses as features (RTF_0). In Table I, we report training and validation errors for each model as long as the ML decision. While both schemes RTF_0 and RTF_1 have almost equivalent training errors (both in MAE and MSE), we can however observe that RTF_0 generalize considerably less to unseen data than RTF_1. The explanation for this result is that the dimensionality of the feature space for RTF_0 (83-dimensional) is simply too high for the relatively small training set at disposition. Regression trees in RTF work by thresholding one-dimensional feature responses (possibly also taking the difference of two dimensions of the feature space) so that deep trees and a large number of training data are needed to learn on such a high dimensional complex feature space. In comparison, it has been observed in [3] that

	Training Set		Validation Set	
	MAE	MSE	MAE	MSE
ML	0.41	0.84	0.44	0.82
RTF_0	0.19	0.10	0.51	0.87
RTF_1	0.19	0.12	0.30	0.28
RTF_3	**0.10**	**0.04**	**0.23**	**0.21**

TABLE I: Training and validation errors of different RTF models.

the right blur radius estimate can generally be found among the first few local maxima of Eq. (3). By sorting the radii in the feature vector for RTF_1, and gathering the relevant information in the same dimensions of the feature space,

[3]Note that we intentionally compute features for some radii larger than the upper bound of the radius used to create our training data. The idea is that it may help the training algorithm to more easily detect patches for which the features are ambiguous (e.g. uniform areas in a sharp image) and therefore allow the algorithm to make better decisions.

we considerably ease the task of the training algorithm to learn good splitting decisions by thresholding one dimensional feature responses. In short, the sorting scheme in Eq. (8) embeds the features into a lower dimensional feature space with a higher predictive power, therefore greatly improving generalization with a limited amount of training data.

While the features proposed for RTF_1 carry meaningful information to discriminate between blur radii, the shifts introduced around blur boundaries due to the width of the analysis window ($W = 41$) make it however difficult to directly regress good local model parameters in these areas. To address this issue, we train a cascade of three RTF models as introduced in [18]. Each model stage of the final model, RTF_3, uses the output of all previous models as additional features and can considerably increase accuracy at blur boundaries.

In Fig. 3, we illustrate the suitability of the proposed RTF_3 by comparing to the more straight-forward training models RTF_0 and RTF_1. We observe that only our guided approach with cascading offers good generalization to unseen data, dramatically improving upon a ML decision by regressing a smoother blur map which is also better aligned with blur discontinuities. In the rest of this paper, RTF_3 is used whenever we refer to the estimation of a blur map b.

IV. Extension of DOF

We now address the parametrization of the deblurring algorithm towards slight extension of DOF. Following [38], we first reformulate the blur model (1) into the matrix-vector form

$$\mathbf{y} = (\hat{\mathbf{K}} - \epsilon_{\mathbf{k}})\mathbf{x} + \mathbf{n} = \hat{\mathbf{K}}\mathbf{x} - \mathbf{e} + \mathbf{n}\,, \tag{9}$$

where $\hat{\mathbf{K}}$ is the estimated blur matrix built from the blur map estimation b (each row of $\hat{\mathbf{K}}$ contains the coefficients of a local disc PSF) and $\epsilon_{\mathbf{k}}$ models the residual blur matrix to compensate for the errors in the estimate $\hat{\mathbf{K}}$. To achieve slight extension of DOF, we construct a deblurring mask $\mathbf{M}$

$$\mathbf{M}[i] = \begin{cases} 1 & b[i] \leq R \\ 0 & b[i] > R \end{cases}\,, \tag{10}$$

where R is the maximum blur radius one wants to remove ($R = \infty$ means complete deblurring). Whenever $\mathbf{M}$ has a zero entry at a given pixel i, we trick the imaging system not to deblur the image at this pixel by assigning a delta blur kernel $\delta[m]$ in the corresponding row of $\hat{\mathbf{K}}$, instead of the estimated blur kernel.

By imposing sparse derivative priors to account for the statistics of sharp natural images [5], [35] and assuming that the error term $\mathbf{e}$ due to kernel estimation errors is sparsely distributed (See section VI-A for justification), we extend the DOF of the image by optimizing

$$\begin{array}{lll} \min_{\mathbf{x},\mathbf{e}} & & \|\hat{\mathbf{K}}\mathbf{x} - \mathbf{e} - \mathbf{y}\|^2 \\ + & \lambda_1 & (||\mathbf{M}\mathbf{D}_{\mathbf{h}}\mathbf{x}||_\alpha^\alpha + ||\mathbf{M}\mathbf{D}_{\mathbf{v}}\mathbf{x}||_\alpha^\alpha) \\ + & \lambda_2 & ||\mathbf{e}||_1\,, \end{array} \tag{11}$$

where $\mathbf{D_h}$ and $\mathbf{D_v}$ denote horizontal and vertical discrete derivative operators, $||\mathbf{M}\mathbf{D_h}\mathbf{x}||_\alpha^\alpha$ with $\alpha \leq 1$ enforces sparse derivatives exclusively on the parts of the image that are actually deblurred and λ_1 , λ_2 are regularization parameters. Using $\mathbf{e} = 0$ as initialization, we minimize this energy function by alternatively solving for $\mathbf{x}$ and $\mathbf{e}$ while keeping the other variable fixed. Concretely, we use an Iterative Reweighted Least Squares (IRLS) approach similar to [5] in order to solve the non-convex subproblem of optimizing $\mathbf{x}$ under α-norms with $\alpha \leq 1$. We solve for $\mathbf{e}$ by minimizing $\|\mathbf{e} - (\hat{\mathbf{K}}\mathbf{x} - \mathbf{y})\|^2 + \lambda_2||\mathbf{e}||_1$, using soft-thresholding.

In order to avoid introducing unrealistic discontinuities in the output image due to the thresholding operation towards slight extension of DOF, a final stage of our algorithm combines both the blurred image $\mathbf{y}$ and the deblurring output $\hat{\mathbf{x}}$ using a smooth version $\mathbf{M}_s$ of the original mask (*e.g.* , by convolving it with a lowpass filter):

$$\mathbf{x}_{\text{final}} = \mathbf{M_s}\hat{\mathbf{x}} + (\mathbf{I} - \mathbf{M_s})\,\mathbf{y} \tag{12}$$

V. Ground truth data

Studies investigating spatially-varying defocus blur generally only resort to qualitative evaluation of their methods on natural images [1], [3], [4] because of the lack of ground truth (GT) data. While quantitative evaluation is performed in [2], it is limited to a rather simplistic dataset based on manually-defined blurred images.

In this paper, we aim to fill this gap and propose a procedure to create realistic GT data (x, y, b) based on a plenoptic camera. These cameras allow to compute, in one single exposure, different images in various configurations of focus distance and depth of field [39]. Images subject to spatially-varying out-of-focus blur can thus be generated along with their corresponding all-sharp images (by approximating a pinhole camera). Such camera provides us thus with a baseline to generate realistic GT image pairs (x_p, y_p). In the following, we introduce a procedure to remedy the lack of GT blur map b explaining the blurring process between x_p and y_p and obtain realistic synthetic GT data (x, y, b).

A. Local estimation of a defocus PSF

To compute a realistic GT blur map b, we propose to blur the all-sharp image x_p with various disc PSFs $k(r)$ and use the image y_p with a shallow DOF only as a template to locally select the best radius explanation. Concretely, for each radius r, we measure a patch-based mean squared error (MSE) distance at each pixel i

$$d(i, r) = \frac{1}{l^2} \sum_{j \in N_i} |y_p[j] - y_r[j]|^2\,,\, y_r = k(r) * x_p\,, \tag{13}$$

where N_i is a very small neighborhood of size $l \times l$ around a given pixel i. The GT blur map b of a realistic synthetic blurred image y_s can then be computed using a minimum distance criterion

$$b[i] = r_i^* = \underset{r}{\operatorname{argmin}}\, d(i, r)\ \ ,\ \ y_s[i] = y_{r_i^*}[i]\,. \tag{14}$$

While y_s optimally fits y_p in a MMSE sense for a given set of PSFs, the GT blur map created as such is, however, prone to errors due to an over-fitting of the criterion to the template-based image y_p in various cases: (i) image content too uniform;

(ii) rendering artifacts in y_p or disc PSF not a good model of the reality; or (iii) chromatic aberrations in y_p. To work around these limitations, we impose a regularization step that detects pixels for which the distance criterion is of high-confidence and propagates their estimation to the non-confident pixels.

B. Detection of high-confidence PSF estimation

Thresholds T_1 and T_2 are used to respectively detect: case (i) when a clear minimum is visible; case (ii) when the minimum distance is sufficiently small.

$$C_1 = \{i| \left(\frac{1}{|R|} \sum_{r \in R} d(i,r) \right) - \min_r d(i,r) > T_1\} \qquad C_2 = \{i| \min_r d(i,r) < T_2\} \tag{15}$$

To prevent wrong estimations due to chromatic aberrations, the local estimation of a PSF is moreover performed over the three color channels R, G and B separately and all pixels for which the estimation is the same are confident.

$$C_3 = \{i| \min_r d_{\mathrm{R}}(i,r) = \min_r d_{\mathrm{G}}(i,r) = \min_r d_{\mathrm{B}}(i,r)\} \ . \tag{16}$$

We obtain a final set of confident pixels by intersecting the three sets: $C = C_1 \cap C_2 \cap C_3$.

C. Regularization

The regularization of the ground truth blur map b is finally achieved by minimizing an energy function for multi-label classification

$$E(b) = \sum_i D_i(r_i) + \sum_{(i,j) \in N} \lambda_{i,j} V_{i,j}(r_i, r_j), \tag{17}$$

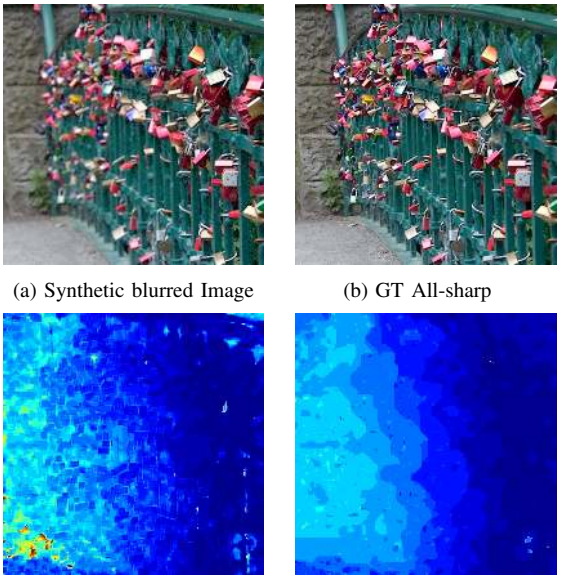
where $D_i(r_i)$ is a data-term encoding the cost of assigning a particular PSF $k(r)$ at pixel i, and $V_{i,j}(r_i, r_j)$ is a smoothness term used for regularization and parametrized by its strength $\lambda_{i,j}$ and the set of neighboring pixels N. To enforce that the PSF estimations of the set of confident pixels C remain the same, we set the data-term cost as follows

$$D_i(r_i) = \begin{cases} 0 & \text{if } i \in C \text{ and } r_i = r_i^* \\ T & \text{if } i \in C \text{ and } r_i \neq r_i^* \\ 1 & \text{if } i \notin C \end{cases} , \tag{18}$$

where T is a large cost especially enforcing that after regularization, $r_i = r_i^*, \forall i \in C$. For non-confident pixels, the data cost is set to the same value for each radius so that the output is completely driven by the regularization. We subsequently model the smoothness term by setting $V_{i,j}(r_i, r_j) = |r_i - r_j|$ with $\lambda_{i,j} = 1$, in order to favor neighboring pixels to have the same blur radius.

D. Dataset Generation

Using a Lytro camera, a dataset of 22 blurred images of size 360×360 was collected following this procedure[4]. In our implementation, the patch-based distance criterion of Eq. 13 is computed with patches of size 7×7 and disc PSFs in the set of radii $R = \{0.5, 0.75, 1, 1.25, ..., 10\}$. Please note that as we cannot know beforehand the largest blur in a given image, the rationale is to choose the upper bound in R large enough to be certain to capture all the potential blur scales. Here it is set to 10 for safety, even if in practice the blur scales in this dataset never exceed disc radius of size 5. For regularization, we set $T_1 = 0.1$, $T_2 = 0.03$, $T = 1000$ and use α-expansion [41] to minimize the energy of Eq. 17 in an 8-neighborhood. The generated GT data are exploited in the rest of our experiments to perform a quantitative evaluation of our algorithm in realistic scenarios, for which a Gaussian noise (variance σ_n=1 and σ_n=2.55) will also be superimposed on the blurred images. One sample of our dataset is shown in Fig. 4, along with the intermediate blur map computed using the MMSE criterion of Eq. 14. We can observe how the MMSE blur map is grainy due to over-fitting, yielding locally various blur scales despite of being at the same distance from the camera. To that extent, our regularized ground truth blur map more plausibly reproduce the physical reality than the MMSE one. Please refer to the supplementary material for more examples.



(a) Synthetic blurred Image (b) GT All-sharp

(c) Intermediate blur map (d) Regularized GT blur map

Fig. 4: Example of ground truth data generation, with superimposition of noise (σ_n=2.55) for the blurred image. (Best viewed zoomed-in in PDF)

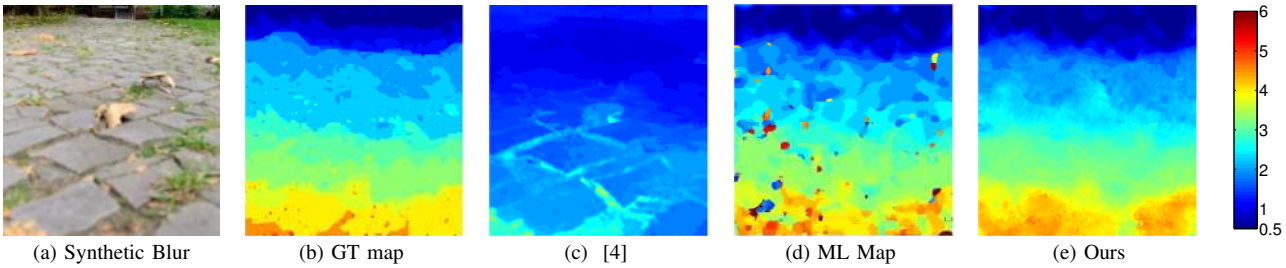
VI. Experiments

In this section, a number of experiments are carried out to test and evaluate the proposed deblurring framework.

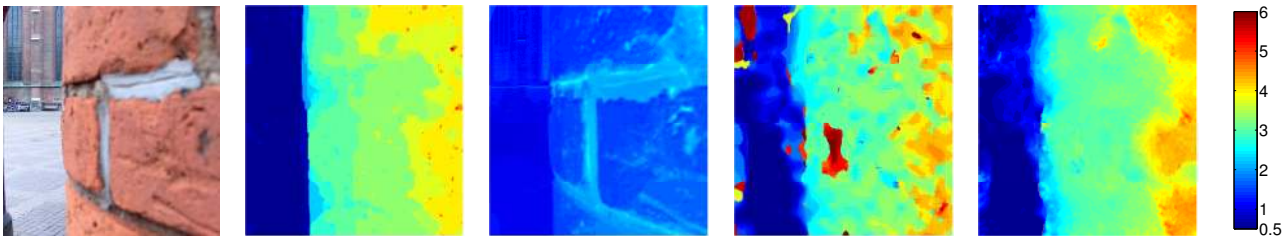
A. Evaluation with realistic dataset and ground truth

Blur map estimation. We first quantitatively evaluate our approach for blur map estimation on our dataset, using the mean absolute error (MAE) and mean squared error (MSE) as baseline quality metrics for blur maps. The complete list of results for each image of our dataset is reported in Table II

[4]Please note that while images of size 1080×1080 pixels are natively output by a (first generation) Lytro camera, studies [40] have pointed out that the sensor can in reality only output an effective spatial resolution of 380×380. To yield sharp enough ground truth data in focused regions, we have therefore downscaled the images by a factor 3, namely to 360×360 pixels.



(a) Synthetic Blur (b) GT map (c) [4] (d) ML Map (e) Ours



(f) Synthetic Blur (g) GT map (h) [4] (i) ML Map (j) Ours

Fig. 5: Examples of blur map estimation in realistic synthetic scenarios. Top: $\sigma_n = 1$, Bottom: $\sigma_n = 2.55$

and compared against the ML blur map (ML) as well as the publicly available edge-based method of Zhuo and Sim [4].[5]

TABLE II: Quantitative evaluation of blur map estimation in realistic scenarios. For each image of our dataset, we report the errors of the estimated blur map compared to our ground truth blur map. For each entry x/y in the table, x is the mean absolute error (MAE) and y the mean squared error (MSE). Std. Dev. stands for the standard deviation per image.

	Noise σ_n=1			Noise σ_n=2.55		
	[4]	ML	RTF	[4]	ML	RTF
Im. 1	0.48/0.33	0.22/0.19	0.18/0.06	0.54/0.41	0.26/0.26	0.20/0.07
Im. 2	0.81/0.87	0.32/0.43	0.24/0.10	0.82/0.91	0.35/0.47	0.27/0.13
Im. 3	0.49/0.31	0.30/0.32	0.24/0.14	0.51/0.34	0.35/0.40	0.27/0.16
Im. 4	0.53/0.41	0.32/0.60	0.17/0.06	0.56/0.44	0.35/0.61	0.19/0.07
Im. 5	0.39/0.18	0.42/0.80	0.19/0.10	0.40/0.19	0.54/1.22	0.29/0.25
Im. 6	0.67/0.59	0.26/0.44	0.17/0.06	0.71/0.67	0.33/0.50	0.20/0.07
Im. 7	0.45/0.26	0.28/0.38	0.19/0.08	0.48/0.28	0.34/0.47	0.22/0.10
Im. 8	1.21/2.33	0.32/0.44	0.22/0.11	1.33/2.86	0.41/0.57	0.25/0.14
Im. 9	0.72/0.58	0.20/0.32	0.13/0.04	0.76/0.64	0.27/0.48	0.17/0.09
Im. 10	0.52/0.37	0.21/0.18	0.19/0.06	0.56/0.43	0.26/0.25	0.21/0.06
Im. 11	0.77/0.75	0.25/0.51	0.20/0.20	0.80/0.81	0.27/0.50	0.22/0.22
Im. 12	1.07/1.42	0.20/0.13	0.17/0.05	1.17/1.77	0.25/0.18	0.19/0.06
Im. 13	0.50/0.35	0.35/0.61	0.17/0.05	0.55/0.42	0.42/0.68	0.25/0.10
Im. 14	0.64/0.53	0.61/1.15	0.49/0.44	0.64/0.54	0.60/1.10	0.50/0.49
Im. 15	1.00/1.17	0.18/0.19	0.16/0.08	1.03/1.26	0.18/0.17	0.17/0.08
Im. 16	0.58/0.56	0.18/0.08	0.18/0.06	0.60/0.64	0.19/0.08	0.18/0.06
Im. 17	1.12/1.48	0.36/0.52	0.24/0.12	1.22/1.81	0.41/0.58	0.27/0.14
Im. 18	0.64/0.53	0.24/0.25	0.17/0.07	0.67/0.58	0.25/0.25	0.18/0.07
Im. 19	0.69/0.62	0.46/0.77	0.30/0.18	0.70/0.68	0.53/0.88	0.32/0.21
Im. 20	0.71/0.68	0.27/0.30	0.20/0.11	0.78/0.84	0.28/0.29	0.21/0.10
Im. 21	0.69/0.58	0.37/0.87	0.18/0.06	0.72/0.63	0.41/0.91	0.21/0.09
Im. 22	1.21/1.77	0.43/0.87	0.20/0.09	1.33/2.16	0.46/0.83	0.23/0.11
Average	0.72/0.76	0.31/0.47	**0.21/0.11**	0.77/0.88	0.35/0.53	**0.24/0.13**
Std. Dev.	0.25/0.55	0.11/0.28	**0.07/0.09**	0.28/0.68	0.11/0.30	**0.07/0.10**

Quantitatively, our approach clearly outperforms [4] both in MAE and MSE at each noise level. While the subpar performance of [4] may be a bit overestimated due to the PSF conversion, the qualitative comparison in Fig. 5 clearly shows the greater ability of our approach to catch the blur trend, without being dependent on the image content. This can be explained by the fact that our method does not primarily rely on edges for blur estimation, but instead use average spectral statistics that are not dependent to the image structure, hence regressing more coherent blur maps. By efficiently removing large artifacts, our approach makes moreover significant quantitative improvements upon the ML estimation (especially in MSE) and results in low-error blur maps for which the blur scales are overall estimated accurately. This is corroborated qualitatively (Fig. 5) by the great ability of our method to catch blur trends, such as gradually changing blur (Fig. 5 (e)) and to handle several levels of blur.

[5]Please note that we map the Gaussian PSF in [4] to a disc PSF by measuring the closest fit when blurring a step edge and that we use the true noise level for extracting likelihoods of the ML map or the features of RTF.

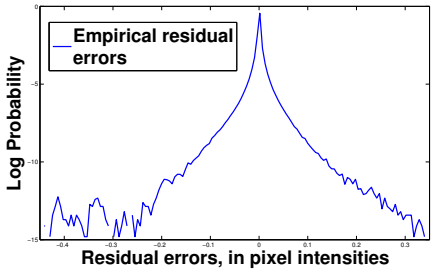
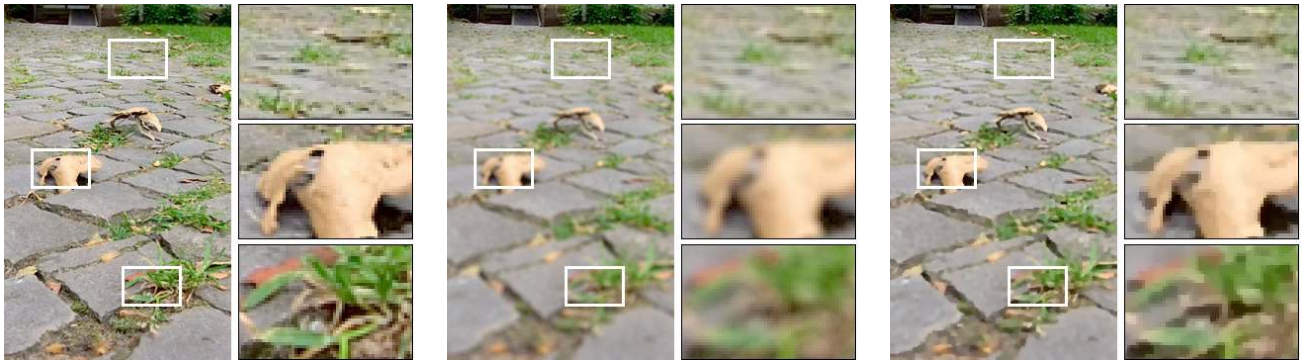


Fig. 6: Empirical log-distribution of the residual term $\mathbf{e} = \epsilon_{\mathbf{k}}\mathbf{x}$ (Eq. 9) computed over our whole dataset.

Statistical characterization of RTF-based blur map estimation errors. It is well known that even small errors in the estimation of blur kernel(s) can easily have dramatic effects on the quality of deblurring algorithms (*e.g.* ringing artifacts). By exploiting our realistic synthetic dataset, we additionally provide a meaningful statistical evaluation of the errors made by our RTF-based blur map estimation towards deblurring. Concretely, based on Eq. 9, we want to estimate the distribution of the term e due to kernel errors. Using the GT data of our dataset, we therefore compute, for each image, the term $\epsilon_{\mathbf{k}} = \mathbf{K} - \hat{\mathbf{K}}$ using our GT blur map and then derive $\mathbf{e} = \epsilon_{\mathbf{k}}\mathbf{x}$ using our latent GT sharp image. In Fig. 6, we plot the empirical error distribution of the term $\mathbf{e}$, evaluated over our whole dataset. We can observe that the distribution is typically sparse around its mode (0). This precious prior



(a) Ground Truth Sharp (b) Blurred (PSNR=24.92 dB) (c) Deblurred (PSNR=27.73 dB)

Fig. 7: Example of deblurring result in a realistic synthetic scenario ($\sigma_n = 1$). (Best viewed zoomed-in in PDF) More examples can be found in supplementary material.

knowledge about kernel errors justifies the model used for deblurring, as it indeed invites to use regularizers of the form $||e||_\beta^\beta$ with $\beta < 1$ in the deconvolution, or an approximation with $\beta = 1$ for computational efficiency.

Spatially-varying deblurring. We now quantitatively evaluate the performance of our deblurring algorithm by estimating the all-sharp images using our blur map estimations. In the following, we set α=0.8 to enforce sparse derivatives, λ_1=2·10^{-4} (σ_n=1) or λ_1=5·10^{-4} (σ_n=2.55) and λ_2=2·10^{-3} as parameters of the deblurring algorithm (Eq. 11). In order to estimate the all-sharp images, *i.e.* to deblur the images entirely, we do not use any deblurring mask here and set $R = \infty$ in Eq. 10.

TABLE III: Quantitative evaluation of spatially-varying deblurring in realistic scenarios. For each image of our dataset, we report the PSNR value when deblurring with various blur map estimations by comparing to our ground truth all-sharp image. Only our blur map estimation allows significant deblurring.

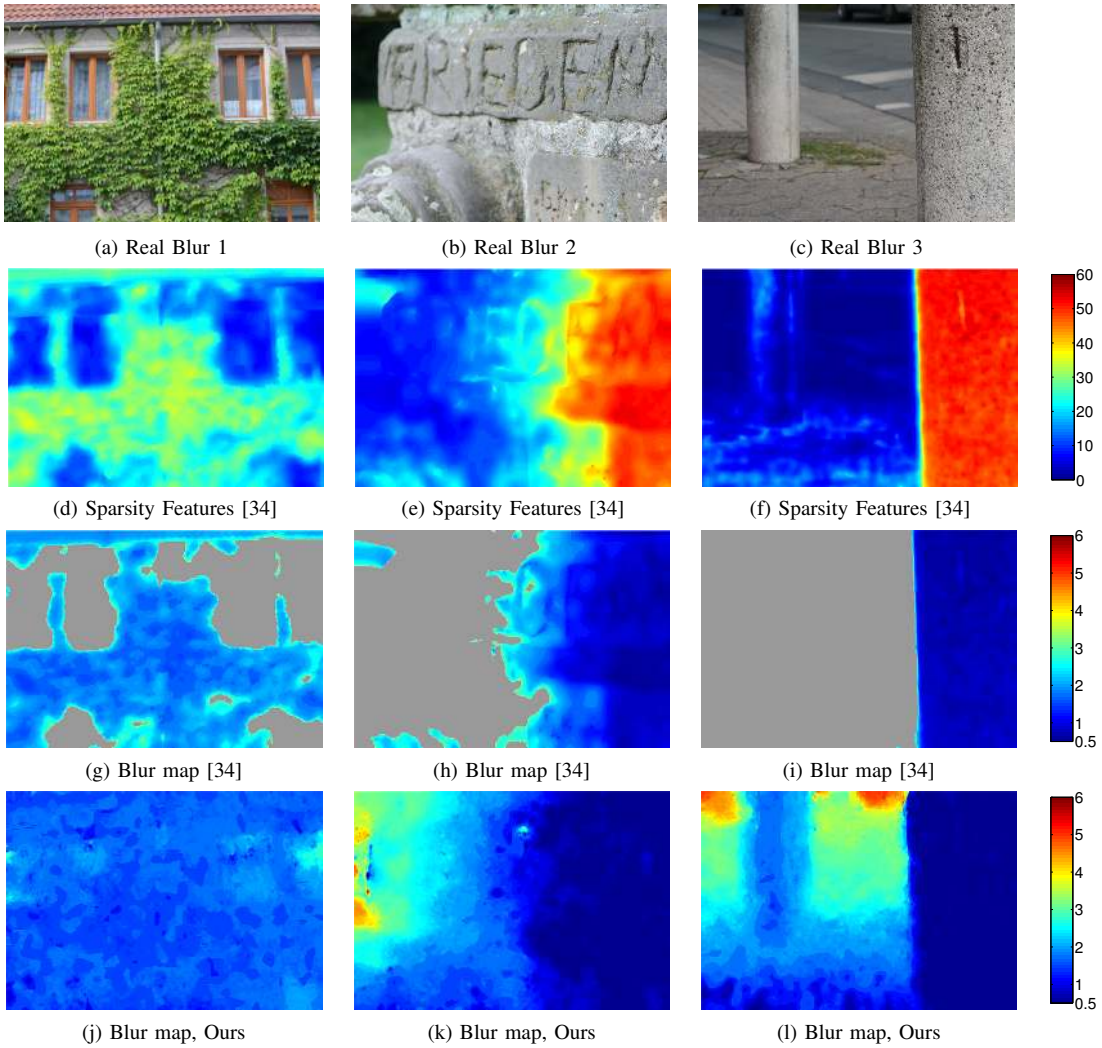
	σ_n=1				σ_n=2.55			
	Blurred	[4]	ML	RTF	Blurred	[4]	ML	RTF
Im. 1	26.70	25.67	28.83	29.32	26.12	26.30	27.86	28.07
Im. 2	25.76	25.49	26.38	27.31	25.28	25.75	26.14	26.53
Im. 3	25.09	25.22	26.64	27.32	24.56	25.44	26.07	26.49
Im. 4	24.87	26.59	26.40	28.42	24.64	26.50	26.60	27.61
Im. 5	27.69	25.93	27.01	30.31	26.88	27.09	27.43	29.04
Im. 6	25.30	24.13	25.93	27.15	25.07	24.83	25.86	26.70
Im. 7	28.43	28.11	29.53	30.67	28.14	28.64	29.10	29.80
Im. 8	24.90	25.81	26.53	27.10	24.72	25.50	25.80	26.03
Im. 9	23.93	22.99	25.82	26.40	23.80	23.61	25.10	25.41
Im. 10	25.34	24.60	26.71	27.14	25.05	25.22	25.96	26.21
Im. 11	24.86	24.59	26.08	26.21	24.65	24.98	25.59	25.73
Im. 12	25.22	25.25	27.93	27.93	25.04	25.42	26.82	26.83
Im. 13	29.57	26.05	27.62	30.75	29.21	27.58	28.30	29.77
Im. 14	26.00	25.69	26.58	28.25	25.82	26.13	26.89	27.44
Im. 15	21.72	21.03	23.87	24.04	21.58	21.51	23.09	23.19
Im. 16	25.24	24.68	27.46	27.61	24.98	25.27	26.55	26.63
Im. 17	25.60	23.58	22.65	25.19	25.31	24.38	23.48	25.18
Im. 18	23.97	23.73	25.33	27.14	23.78	24.12	25.38	26.34
Im. 19	25.61	25.57	26.97	28.43	25.41	25.93	27.22	27.75
Im. 20	23.58	22.88	23.93	24.75	23.39	23.24	23.76	24.30
Im. 21	27.08	26.57	25.76	28.72	26.85	26.96	26.37	28.01
Im. 22	25.36	24.48	24.92	27.06	25.22	25.02	25.29	26.44
Average	25.54	24.94	26.31	**27.60**	25.25	25.43	26.12	**26.80**
Gain	n/a	-0.6	+0.77	**+2.06**	n/a	+0.18	+0.87	**+1.55**

In Table III, we compare the PSNR value of each original blurred image of our dataset to the ones obtained after deconvolution with the different blur map estimations (Edge-based [4], ML, Ours). We observe that only our RTF-based blur map estimation allows to significantly and consistently improve the PSNR, with average gains of 2.06 ± 0.83 dB (σ_n=1) and 1.55 ± 0.67 dB (σ_n=2.55). To put the relative small margin obtained in perspective, one has to consider the complexity of the scene and that typical disc defocus PSFs cannot be properly inverted (zeros in its Fourier Transform). Although a direct comparison with [2] was not possible, we can compare their gains (+0.76 dB with σ_n=1) on their dataset with our gains (+2.06 dB (σ_n=1), +1.55 dB (σ_n=2.55)) on a more complex and realistic dataset. Without being able to draw further conclusions, these figures confirm the good performance of our method in comparison to the state of the art. Fig. 7 illustrates the kind of deblurring results obtained in synthetic scenarios of our dataset. Substantially more examples can be found in the supplementary material.

B. Application to real data

Defocus blur map estimation. In order to show the applicability of our method to real images, a small dataset subject to real out-of-focus blur has been acquired with a Nikon D7100. A subset of blur map estimation results is shown in Fig. 8. It is qualitatively compared to the recent dictionary-based approach of Shi *et al.* [34], which tackles "just noticeable blur" detection. We can see that our method has the greatest advantage of handling the estimation of larger blur scales than [34]. Indeed, even if the sparsity features of [34] in Fig. 8 (d)-(f) may catch the blur trend for the whole image, the method however breaks down for standard deviation larger than $\sigma = 2$ (Gaussian PSF) when converting to an exact blur strength. This is in particular critical towards our deblurring target scenario, for which an exact estimation of the blur scales is needed. Qualitatively, our results are also more coherent with the natural depth variation in the images, in particular for the almost uniformly blurred image in Fig. 8 (a) or the detection of the left post in Fig. 8 (c).

In Fig. 9, an additional comparison with Zhu *et al.* [3] on real images shows that our method performs similarly well or slightly better in case of weak color boundaries and is typically better in gradually changing blur. See for example how the smooth blur progression is more accurately captured



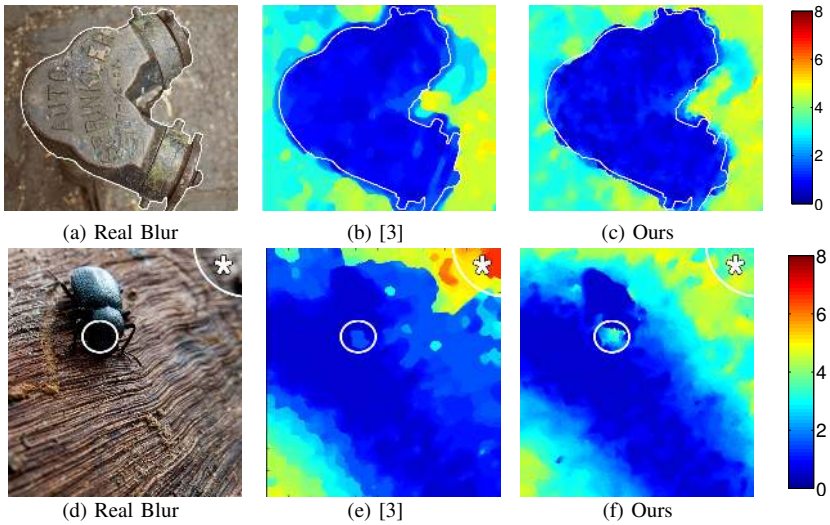
(a) Real Blur 1 (b) Real Blur 2 (c) Real Blur 3

(d) Sparsity Features [34] (e) Sparsity Features [34] (f) Sparsity Features [34]

(g) Blur map [34] (h) Blur map [34] (i) Blur map [34]

(j) Blur map, Ours (k) Blur map, Ours (l) Blur map, Ours

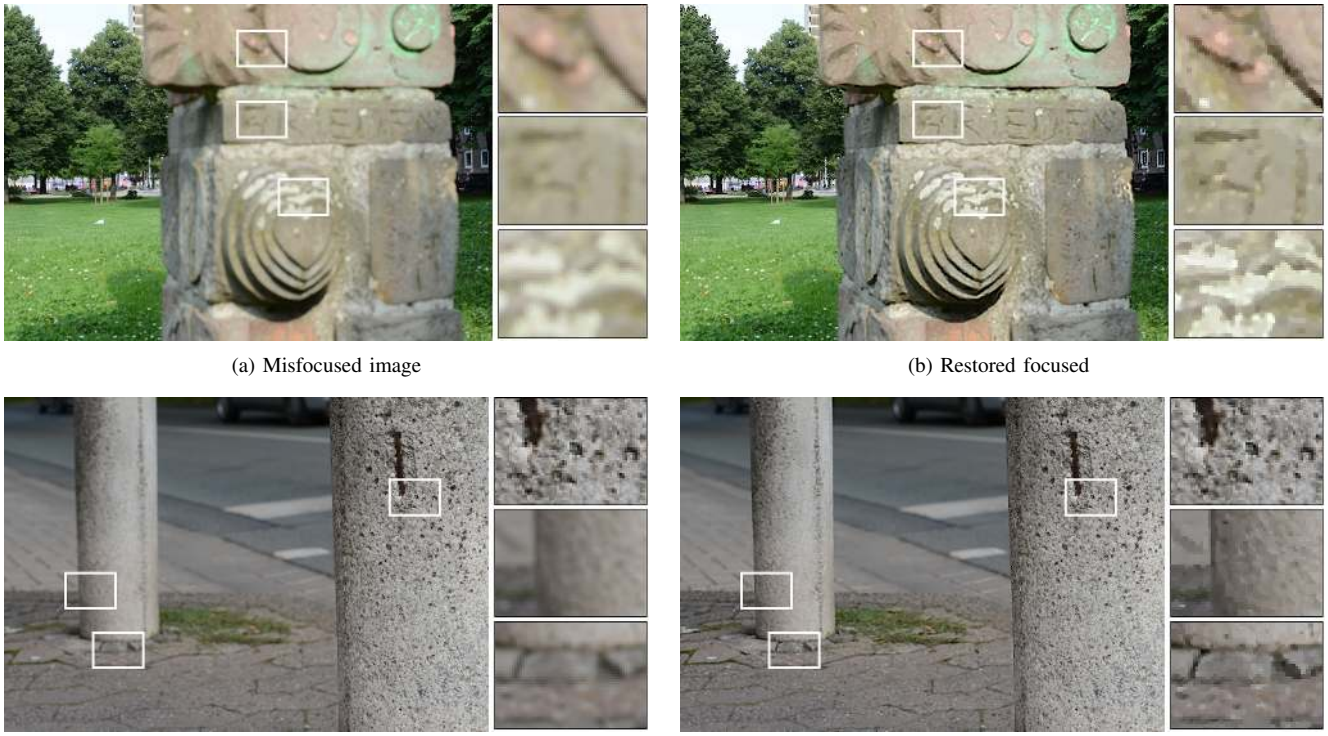
Fig. 8: Blur map estimation examples. Comparison with Shi *et al.* [34] on real images. The sparsity features of [34] ((d)-(f)) are converted to a Gaussian PSF following the formula provided in their paper, and subsequently to a disc PSF for visualization comparison with our approach (by measuring the closest fit when blurring a step edge). Please note that the conversion of sparsity features is only valid up to a Gaussian PSF with standard deviation $\sigma = 2$. Regions with larger blur strengths (or not enough texture) which do not produce an estimation of the blur scale with [34] are marked in uniform gray in (g)-(i).



(a) Real Blur (b) [3] (c) Ours

(d) Real Blur (e) [3] (f) Ours

Fig. 9: Blur map estimation examples. Comparison with Zhu *et al.* [3] on real images. Please note that our method has not been trained to handle the large blur in the region marked by (*).



(a) Misfocused image

(b) Restored focused

(c) Very Shallow DOF

(d) Slight Extension of DOF

Fig. 10: Slight extension of DOF with real images. Top: restoration of a misfocused image. Bottom: restoration of an image subject to a too shallow DOF. (Best view zoomed-in in PDF) More examples can be found in supplementary material.

by our method, in the top left corner and right part of Fig. 9 (e) versus (f). Our simple learning-based approach remains however limited around blur boundaries aligned with good image contrast, which may not be as sharp as models based on strong color constraints [2], [3].

Depth of field extension on real images. Based on the same images gathered for blur map estimation, we now target our final deblurring application. A subset of DOF extension results is first reported in Fig. 10. We can see that, despite the lack of calibration of our PSF model to real cameras, our method is able to significantly deblur the images locally. For example, Fig. 10 (a) shows an image wrongly focused on the background of the scene instead of the central stone statue in the foreground. By applying our spatially-varying deblurring algorithm with $R = \infty$ in Eq. 10 (*i.e.* complete deblurring or infinite DOF), the stone statue is brought back to focus by recovering a decent level of sharpness in the restored image of Fig. 10 (b), while the background, already focused, is correctly left unchanged. In Fig. 10 (c), the image is focused on the post on the right part of the image and shot with a really shallow DOF so that the left post is already out-of-focus. Note how our method is able to slightly extend the DOF in Fig. 10 (d) (using $R = 2.5$ in Eq. 10), bringing the left post to focus, without over-sharpening in-focus areas and leaving the background out-of-focus.

In Fig. 11, we provide additional results that are compared to the ones obtained using the blur maps of Shi *et al.* [34] shown in Fig. 8 (but using the original Gaussian PSF for deconvolution). For the misfocused image in (a) (almost uniform blur), only our method is able to recover sharpness over the whole image, while [34] misestimates the regions around the windows and leaves them out of focus. In the second image in (b), subject to a very shallow DOF and gradually changing blur, our restoration algorithm ($R = 3$) is able to slightly extend the DOF over the whole monument. In comparison, the result of [34] is limited to one half of the monument and is prone to over-sharpening (middle part). Overall, it is clear that towards deblurring, our method seems to be more stable while having the advantage of being able to handle larger blurs than [34].

Video Deblurring. We finally perform a real experiment in the context of video deblurring. It is based on the HEVC DASH dataset [42], which is a real-life professional edit of several sequences shot for the 4Ever project [43]. Two (cropped) frames of the subsequence we have used are shown in Fig. 12 (a) and (g). This scene is particularly interesting because it displays a so shallow DOF that the only focused region is the middle of the interviewee's face. His ear and shirt are already slightly blurred, the background is completely out of focus and some frames also display some slight motion blur, such as the face moving on frame 125 (Fig. 12 (g)).

In this experiment, we have tried to slightly extend the DOF of the images in order to recover sharpness around the ear and shirt. To showcase the strengths and weaknesses of our approach, we compare our results to the recent and state-of-the-art deblurring method of Kim and Lee [44], which handles various sources of spatially-varying blur in videos (camera shake, motion blur, depth variation). Deblurring results for both approaches are shown in Fig. 12, along with a subset of the pixel-wise varying kernels of [44] and our defocus blur

map overlaid with the original image.

Despite minor errors (mainly around uniform regions of the face for which no deblurring is needed), our blur map accurately estimates the different blur levels: (i) absence of blur around the middle of the face; (ii) slight defocus blur for the ear and shirt collar; and (iii) strong defocus blur for the background. As a result, as shown in Fig. 12 (c) and (i), our method is able to deblur the slightly defocused areas, recovering image details and sharpness around the ear and shirt collar as desired. One can however notice the sensitivity of our defocus blur map to motion blur, on the face part of Fig. 12 (l). This is due to the fact that small disc PSFs more closely match the slight motion blur PSFs than a delta kernel (no blur) in presence of motion, and tends to degrade the quality of the deblurring result because the estimated kernel does not perfectly correspond to the real motion blur kernel. In comparison, standard spatially-varying deblurring methods addressing motion blur, such as [44], are not tailored to handle defocus blur. As we can see on the pixel-wise blur kernels of Fig. 12 (e) and (k), [44] is indeed not able to detect defocus blur and estimates delta kernels (no blur) in absence of motion blur. While their method brightly deblurs the slight motion blur that our approach does not handle, we also observe that it cannot be applied for our target scenarios since it has no effect on defocus blur.

In short, this experiment showcases the contribution of our work towards handling spatially-varying defocus blur and DOF extension in videos. We also note that the frame-by-frame deblurring approach followed in this experiment is overly simplistic as no mechanism to ensure temporal coherency is considered, which has the undesired effect of introducing flickering artifacts in the final video. To apply our approach for serious video processing applications, future work should therefore be focused on addressing both the problem of motion blur and temporal consistency.

C. Limitations

In this work, we have shown how even a naive training of RTF, based on randomly synthetically generated blur data, can be useful to solve the task of deblurring spatially-varying out-of-focus images. One current limitation of our approach is the limited performance of our learned model around blur kernel boundaries (depth boundaries). This can in particular be observed when the surrounding region of a sharp or slightly blurred region is completely out-of-focus (e.g., near the interviewee's ear in Fig. 12). In this work, we have relied on the smart regularization of our deblurring algorithm, especially the modeling of blur kernel estimations errors inspired by [38], to avoid unpleasant ringing artifacts.

Better training models could however be built to estimate kernel boundaries more accurately. In particular, we currently do not model the natural variation of depth in our naive training data (e.g., blur discontinuities do not align well with color discontinuities), as it was more straightforward to use randomly generated blur data to obtain a dataset large enough for training. This forces RTF to make blur boundary decisions based on gradient and spectral characteristics only (mainly to correct the shifts incorporated in the input features). While this has shown to work in some cases (for example in Fig. 3 (f)), this remains suboptimal and prone to errors. Further work should therefore focus on using more properly formed training data, where in particular blur boundaries correspond to depth/color boundaries. This will obviously require to incorporate color informations as features in Eq. 8 to let the new model learn changes in defocus blur beyond the spectral features. One possible direction is definitely to use the method presented in Section V to create spatially-varying ground truth training data which are realistic. The use of the first generation of the Lytro camera we have used in this work makes it difficult to easily build a suitable large training dataset due to the limited image quality and the small sensor size, which does not allow to easily capture images with the required very shallow DOF or sufficient blur.

VII. Conclusion

We introduced a blind deblurring pipeline for the restoration of images with too shallow Depth-of-Field (DOF). At its core is the estimation of a defocus blur map, based on a model cascade of Regression Tree Fields (RTF). We showed how even simple training data, manually generated by synthetic blur map patterns, can be combined with local spectral blur cues to train a discriminative model able to regress accurate defocus blur maps. Their successful application for restoring spatially-varying out-of-focus blurred images is demonstrated through various experiments with both synthetic and real images. To that end, based on a plenoptic camera, a novel approach to remedy the lack of realistic ground truth data in spatially-varying out-of-focus problems has also been proposed in order to provide a quantitative evaluation of our framework. We believe that a promising area for further research will be their use as training data for learning-based approaches. The success of such algorithms is intrinsically linked to the ability of reproducing real conditions during training, and such realistic spatially-varying ground truth data may considerably help learning better models.

Acknowledgment

We would like to thank the authors of [3] for kindly sharing their dataset of images to us, and, therefore, making a direct comparison to their approach possible.

References

[1] S. Bae and F. Durand, "Defocus magnification." *Comput. Graph. Forum*, vol. 26, pp. 571–579, 2007.

[2] F. Couzinié-Devy, J. Sun, K. Alahari, and J. Ponce, "Learning to estimate and remove non-uniform image blur," in *CVPR*, 2013.

[3] X. Zhu, S. Cohen, S. Schiller, and P. Milanfar, "Estimating spatially varying defocus blur from a single image," *IEEE, Transactions on Image Processing*, 2013.

[4] S. Zhuo and T. Sim, "Defocus map estimation from a single image," *Pattern Recogn.*, vol. 44, no. 9, pp. 1852–1858, Sep. 2011.

[5] A. Levin, R. Fergus, F. Durand, and W. T. Freeman, "Image and depth from a conventional camera with a coded aperture," *ACM Trans. Graph.*, vol. 26, no. 3, p. 70, 2007.

[6] F. Guichard, H. Nguyen, R. Tessières, M. Pyanet, I. Tarchouna, and F. Cao, "Extended depth-of-field using sharpness transport across color channels," in *IS&T-SPIE Electronic Imaging*, 2009, p. 72500.

[7] M. W. Tao, J. Malik, and R. Ramamoorthi, "Sharpening out of focus images using high-frequency transfer," *Computer Graphics Forum*, 2013.

[8] J.-F. Cai, H. Ji, C. Liu, and Z. Shen, "Blind motion deblurring from a single image using sparse approximation," in *CVPR*, 2009.

[9] R. Fergus, B. Singh, A. Hertzmann, S. T. Roweis, and W. T. Freeman, "Removing camera shake from a single photograph," *ACM Trans. Graph.*, vol. 25, no. 3, pp. 787–794, Jul. 2006.

[10] M. Hirsch, C. J. Schuler, S. Harmeling, and B. Schlkopf, "Fast removal of non-uniform camera shake." in *ICCV*, 2011.

[11] H. Ji and K. Wang, "A two-stage approach to blind spatially-varying motion deblurring," in *CVPR*, 2012.

[12] A. Levin, Y. Weiss, F. Durand, and W. T. Freeman, "Efficient marginal likelihood optimization in blind deconvolution," in *CVPR*, 2011.

[13] T. Michaeli and M. Irani, "Blind deblurring using internal patch recurrence," in *ECCV*, 2014.

[14] L. Sun, S. Cho, J. Wang, and J. Hays, "Edge-based blur kernel estimation using patch priors," in *Proc. IEEE International Conference on Computational Photography*, 2013.

[15] O. Whyte, J. Sivic, A. Zisserman, and J. Ponce, "Non-uniform deblurring for shaken images," in *CVPR*, 2010.

[16] L. Xu and J. Jia, "Two-phase kernel estimation for robust motion deblurring," in *ECCV*, 2010.

[17] J. Jancsary, S. Nowozin, T. Sharp, and C. Rother, "Regression tree fields - an efficient, non-parametric approach to image labeling problems," *CVPR*, 2012.

[18] U. Schmidt, C. Rother, S. Nowozin, J. Jancsary, and S. Roth, "Discriminative non-blind deblurring," in *CVPR*, 2013.

[19] A. Chakrabarti, T. Zickler, and W. T. Freeman, "Analyzing spatially-varying blur," in *CVPR*, 2010.

[20] A. Levin, Y. Weiss, F. Durand, and W. T. Freeman, "Understanding blind deconvolution algorithms," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 12, pp. 2354–2367, Dec. 2011.

[21] M. Potmesil and I. Chakravarty, "Synthetic image generation with a lens and aperture camera model," *ACM Trans. Graph.*, vol. 1, no. 2, pp. 85–108, Apr. 1982.

[22] E. Kee, S. Paris, S. Chen, and J. Wang, "Modeling and removing spatially-varying optical blur," in *ICCP*, 2011.

[23] M. Subbarao and G. Surya, "Depth from defocus: A spatial domain approach," *International Journal of Computer Vision*, vol. 13, pp. 271–294, 1994.

[24] A. P. Pentland, "A new sense for depth of field," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 9, no. 4, pp. 523–531, Apr. 1987.

[25] M. Watanabe and S. K. Nayar, "Rational filters for passive depth from defocus," *International Journal of Computer Vision*, vol. 27, no. 3, pp. 203–225, 1998.

[26] P. Favaro and S. Soatto, "Learning shape from defocus," in *Proceedings of the 7th European Conference on Computer Vision-Part II*, ser. ECCV '02, 2002, pp. 735–745.

[27] H. Jin and P. Favaro, "A variational approach to shape from defocus," in *Proceedings of the 7th European Conference on Computer Vision-Part II*, ser. ECCV '02, 2002, pp. 18–30.

[28] P. Favaro and S. Soatto, "A geometric approach to shape from defocus," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 3, pp. 406–417, 2005.

[29] J. Ens and P. Lawrence, "An investigation of methods for determining depth from focus," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 15, no. 2, pp. 97–108, Feb. 1993.

[30] S. Nayar and Y. Nakagawa, "Shape from Focus," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, no. 8, pp. 824–831, Aug 1994.

[31] S. W. Hasinoff and K. N. Kutulakos, "Confocal stereo," in *Proceedings of the 9th European Conference on Computer Vision - Volume Part I*, ser. ECCV'06, 2006, pp. 620–634.

[32] Y.-W. Tai and M. S. Brown, "Single image defocus map estimation using local contrast prior," in *ICIP*, 2009.

[33] N. Joshi, R. Szeliski, and D. Kriegman, "Psf estimation using sharp edge prediction," in *CVPR*, 2008.

[34] J. Shi, L. Xu, and J. Jia, "Just noticeable defocus blur detection and estimation," in *CVPR*, 2015.

[35] D. Krishnan and R. Fergus, "Fast image deconvolution using hyper-laplacian priors," in *NIPS*, 2009.

[36] U. Schmidt, K. Schelten, and S. Roth, "Bayesian deblurring with integrated noise estimation." in *CVPR*. IEEE, 2011, pp. 2625–2632.

[37] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 5, pp. 898–916, May 2011.

[38] H. Ji and K. Wang, "Robust image deblurring with an inaccurate blur kernel." *IEEE Transactions on Image Processing*, 2012.

[39] R. Ng, "Digital light field photography," Ph.D. dissertation, Stanford, CA, USA, 2006, aAI3219345.

[40] V. Boominathan, K. Mitra, and A. Veeraraghavan, "Improving resolution and depth-of-field of light field cameras using a hybrid imaging system." in *ICCP*, 2014.

[41] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2001.

[42] J. Le Feuvre, J.-M. Thiesse, M. Parmentier, M. Raulet, and C. Daguet, "Ultra high definition hevc dash data set," in *Proceedings of the 5th ACM Multimedia Systems Conference*, ser. MMSys '14. New York, NY, USA: ACM, 2014, pp. 7–12. [Online]. Available: http://doi.acm.org/10.1145/2557642.2563672

[43] "4ever project," http://www.4ever-project.com/, last checked: July 2015.

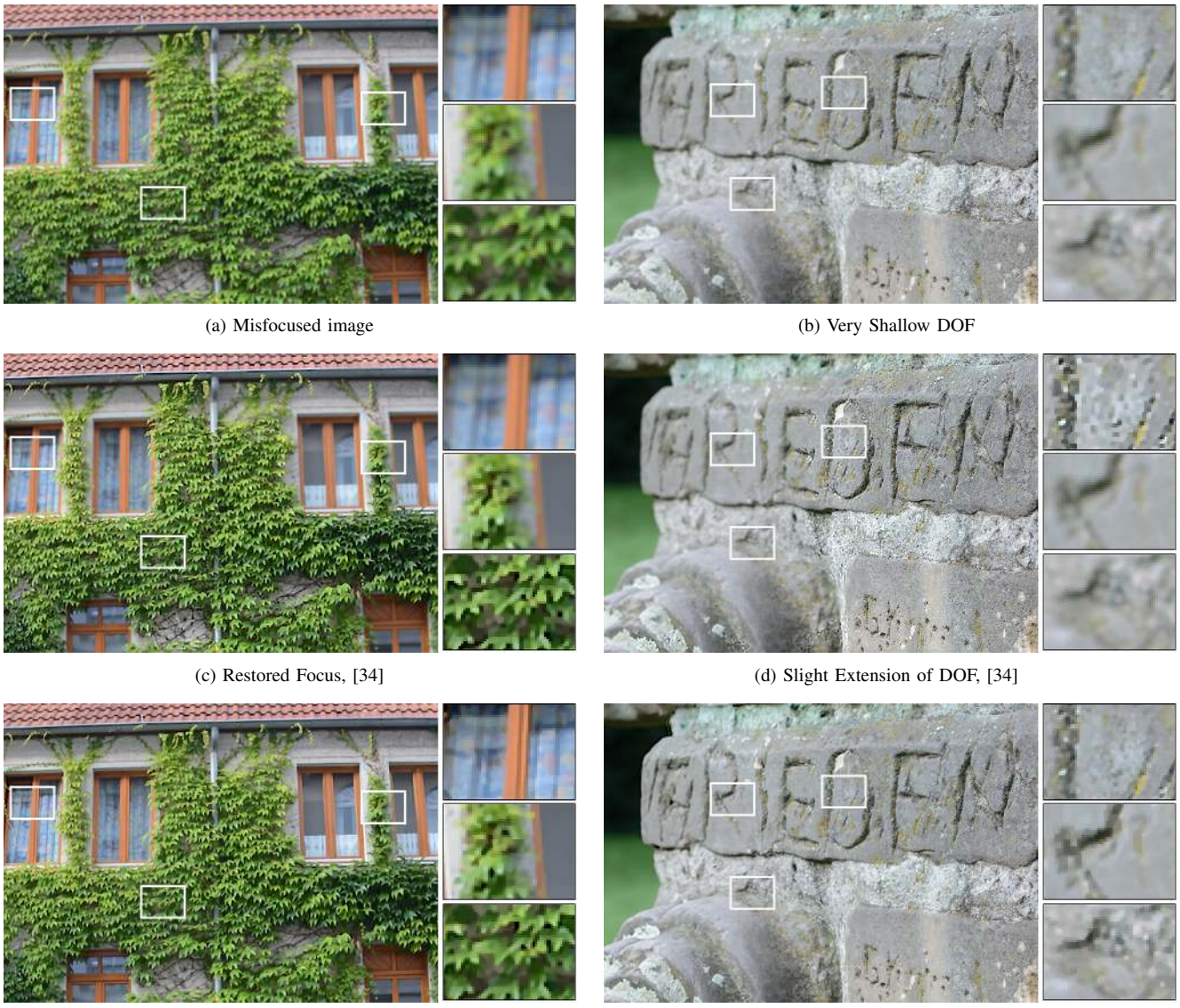
[44] T. H. Kim and K. M. Lee, "Generalized video deblurring for dynamic scenes," in *CVPR*, 2015.

Laurent D'Andrès received the B.Sc. and M.Sc. degrees in Communication Systems from the Swiss Federal Institute of Technology in Lausanne (EPFL), Switzerland, in 2012 and 2014, respectively. His fields of interest concern digital signal processing, image processing and machine learning. He accomplished his master's thesis as a Research Intern at Technicolor R&I labs in Hannover and worked on designing new machine learning based algorithms towards out-of-focus deblurring and depth of field extension.

Jordi Salvador is project leader at Technicolor R&I in Hannover and member of the Technicolor's Fellowship Network since 2014. His main research focus is on machine learning for image super resolution and restoration. Formerly, he obtained the Ph.D. degree in 2011 from the Universitat Politècnica de Catalunya (UPC), where he contributed to projects of the Spanish Science and Technology System (VISION, PROVEC) and also to a European FP6 project (CHIL) as research assistant on multiview 3D reconstruction. He has also served as reviewer in several conferences and journals. His research interests include 3D reconstruction, real-time and parallel algorithms, image and video restoration, inverse problems and machine learning.

Axel Kochale (IEEE Member since 1993) is managing a team working on resolution enhancement as technology area leader at Technicolor R&I Hannover/Germany. He obtained is diploma (Dipl.Ing.FH) in telecommunication from the University of Applied Science and Arts (Hannover/Germany) in 1993. From 1993 he started working as R&D engineer in the company now named Technicolor to work on video processing for consumer products, professional video broadcast equipment and solutions for media production. He currently holds 24 patents in this field and is member of the Technicolor Fellowship network. His current research interest include super resolution upscaling, image deblurring and denoising, and its application to media production and interactive TV.

Sabine Süsstrunk. leads the Images and Visual Representation Lab (IVRL) in the School of Computer and Communication Sciences (IC) at EPFL since 1999. Her main research areas are in computational photography, color imaging, multimedia, and image quality. She has authored and co-authored over 150 publications and holds 8 patents. In 2013, she received a Best Paper Award at the IEEE International Conference on Image Processing (ICIP) and the IS&T/SPIE 2013 Electronic Imaging Scientist of the Year Award. Sabine is currently Associate Editor for the IEEE Transactions on Computational Imaging.



(a) Misfocused image

(b) Very Shallow DOF

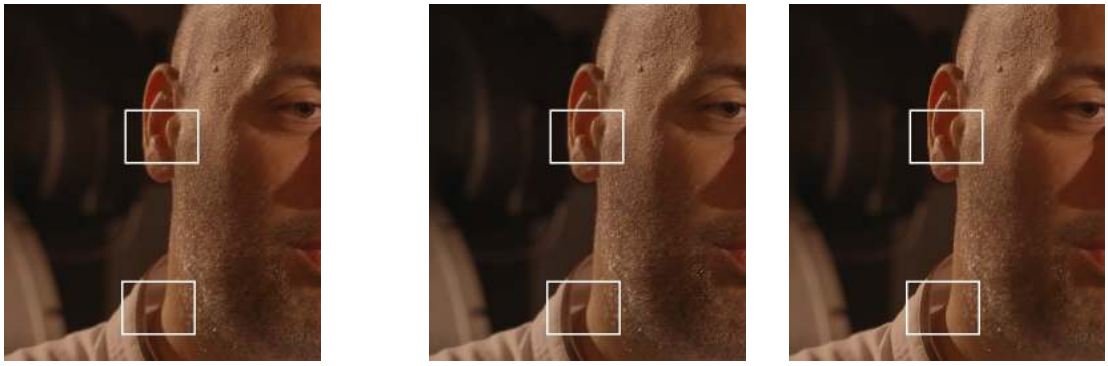
(c) Restored Focus, [34]

(d) Slight Extension of DOF, [34]

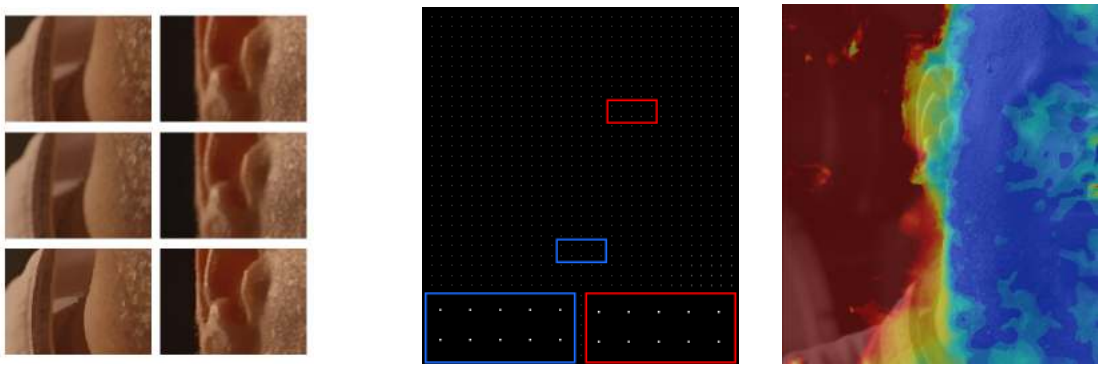
(e) Restored Focus, Ours

(f) Slight Extension of DOF, Ours.

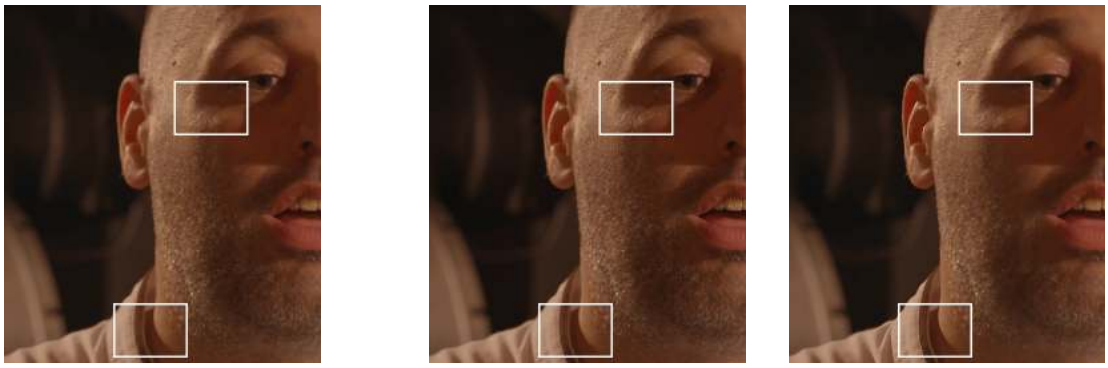
Fig. 11: Deblurring examples. Comparison with Shi *et al.* [34] on real images. From top to bottom: Blurred image, [34], Ours. Left: restoration of a misfocused image. Right : restoration of an image subject to a too shallow DOF. See how our method is more stable in order to to restore focus over the whole misfocused image (e) or to extend the DOF over the whole monument shot with a shallow DOF (f). (Best viewed zoomed-in in PDF)



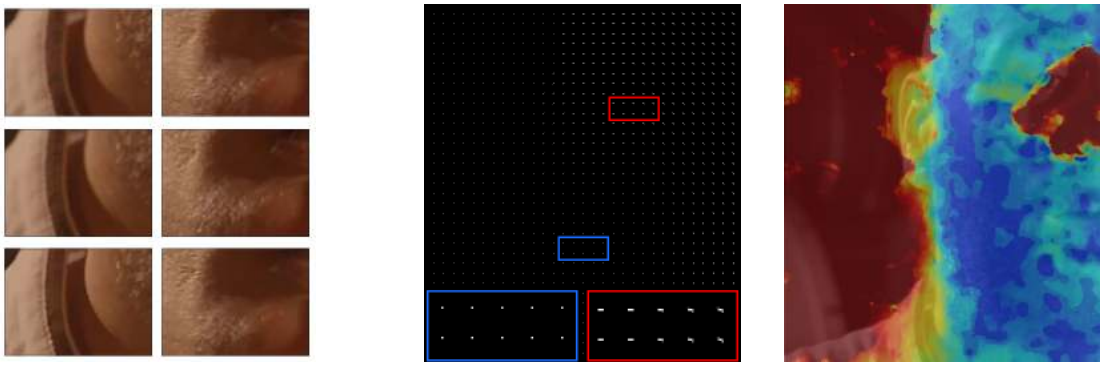
(a) Frame 224, shallow DOF (b) Kim and Lee [44] (c) Ours



(d) Close-ups (e) Pixel-wise varying kernels of [44] (f) Our defocus blur map



(g) Frame 125, shallow DOF + slight motion blur (h) Kim and Lee [44] (i) Ours



(j) Close-ups (k) Pixel-wise varying kernels of [44] (l) Our defocus blur map

Fig. 12: Sample video deblurring and comparison with Kim and Lee [44], based on 238 frames extracted from the HEVC DASH dataset [42]. The frames 125 and 224 of our subsequence are shown. For each frame, from left to right, top to bottom: image with shallow DOF (original frame), deblurred image by [44], slight extension of DOF (Ours), close-ups corresponding to the boxes shown in red (from top to bottom: Original frame, [44], Ours), subset of spatially-varying kernels estimated by [44], our defocus blur map overlaid with the image. Only our method is suitable towards our DOF extension scenario.