Collaborative Blind Image Deblurring

Thomas Eboli[1] Jean-Michel Morel[2] Gabriele Facciolo[1]

[1]Université Paris-Saclay, CNRS, ENS Paris-Saclay, Centre Borelli
[2]City University of Hong Kong

Abstract

Blurry images usually exhibit similar blur at various locations across the image domain, a property barely captured in nowadays blind deblurring neural networks. We show that when extracting patches of similar underlying blur is possible, jointly processing the stack of patches yields superior accuracy than handling them separately. Our collaborative scheme is implemented in a neural architecture with a pooling layer on the stack dimension. We present three practical patch extraction strategies for image sharpening, camera shake removal and optical aberration correction, and validate the proposed approach on both synthetic and real-world benchmarks. For each blur instance, the proposed collaborative strategy yields significant quantitative and qualitative improvements.

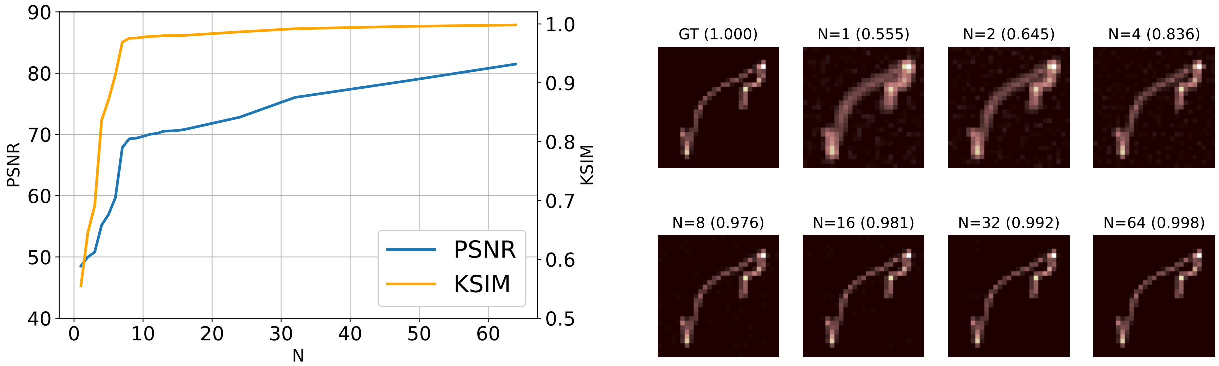
1. Introduction

Image deblurring is the problem of predicting sharp images from blurry ones. In the most common case, a single blurry image is available, and the goal is to predict its sharp version. Except in rare situations where the blur is known, *e.g.,* via embedded hardware [36] for camera shake, or via lens calibration [5] for lens blur, we have little information on the blur. This blind setting is highly ill-posed and requires priors over both the latent sharp image and the blur to be solved [29]. Classical approaches to blind deblurring leverage image and blur kernel priors to first estimate the blur kernel, and second estimate a sharp image via non-blind deblurring. The best priors for doing so, *e.g.,* [34], are based on picking the salient edges across the image that provide high-quality hints on the blur [22].

Yet, ever since the introduction of large corpora of sharp/blurry image pairs such as the GoPro dataset [33], neural networks achieve state-of-the-art deblurring results. They are trained to predict a sharp image directly from a blurry input, without requiring the intermediate estimation of the blur kernel. To do so, the network may need to extract some relevant features of the blur in order to predict correctly the missing high frequencies. The hints needed to determine the blur (the edges [22]) are usually spread across larger regions of the image than what is captured by the receptive fields of convolutional networks. This a reason why the multi-scale architectures [10, 33] are so widespread for this task. Self-attention [41] can collect such widespread hints, however they are too expensive to deploy in realistic scenarios with large images [30]. Despite impressive results, none of these approaches truly leverage the existing image processing expertise. These sparse hints are well exploited by classic methods either by exploiting the global property of the Fourier transform [20] or by analyzing the directional gradient histogram of the image [16].

In this work we propose to feed the network the relevant hints needed to improve the ability of the model to extract the blur from the representation of the blurry image. We propose a collaborative scheme where the network jointly processes several patches with the same underlying blur as a way to disambiguate the blur. Increasing the number of patches increases the probability of collecting all the useful information needed to deblur. Collaboration happens by a sort of “attention-without-attention” module, which amounts to selecting relevant patches for the task at hand. For instance, in the case of motion deblurring, patches coming from the same moving object. Note that the patches can come from locations thousands of pixels apart, which is in practice hardly achievable by convolutions or attention. We show that collecting such sets of patches sharing similar underlying blur is straightforward for three practical instances of blur: camera shake [44], optical aberrations [39] and mild blurs [16]. Within the network, this collaboration is achieved by processing the patches in parallel and by inserting pooling layers that foster the collaboration between the encoded features. Our experiments show that this collaborative processing boosts the deblurring accuracy of the network. This strategy can be applied to a variety of architectures. A practical application of our technique illustrated in this paper is designing lightweight yet efficient blind deblurring networks.

Our contributions are summarized as follows: (1) We



(a) Average of the 8 kernels from [29].

(b) The fourth kernel from [29].

Figure 1. Illustration of the collaborative scheme with the kernels from [29]. We report the PSNR and the kernel similarity (KSIM) as defined in [45] on the left and the KSIM for each qualitative kernel on the right. Combining more sharp/blur pairs dramatically improves the accuracy of the kernel support, with saturation occuring at $N = 8$.

propose a collaborative strategy that consists in gathering a stack of patches with similar underlying blur in an image, and jointly processing them in a neural network upgraded with a layer pooling along the stack dimension. (2) We show its practicality for three instances of blur: camera shake, optical aberrations and mild blurs. (3) We provide theoretical elements to connect the proposed approach to existing classical blind deblurring methods. (4) We show on both real-world and synthetic data the efficiency of the approach for the three sorts of blur listed above, and validate two elements of design: how many patches should collaborate and which pooling function to use.

2. Related work

Classic blind image deblurring algorithms alternate between a kernel-estimation step, and a non-blind deblurring step [29]. At each step the estimated kernel is used to recover a sharper image by the non-blind deblurring, which in turn is used to refine the kernel prediction. The kernel prediction makes use of domain knowledge such as smoothness and sparsity of camera shake [44], approximate symmetry of optical aberration [39], the Gaussian shape of defocus blur [21] or translational motions in street photography [12]. The work of Nah *et al.* [33] adopts instead a black-box paradigm by learning from a large dataset of aligned blurry and sharp image pairs a multi-scale CNN that predicts from a single blurry image a restored variant, without the need for traditional image priors or any explicit structure on the family of blur to remove. The properties on the blur are now determined by the training dataset, for instance camera shake [38], defocus [1] or dynamic motions [33]. Subsequent architectures follow this trend by introducing recurrent layers [40], additional skip connections [19, 35], attention modules [10, 42], adversarial losses [24], patch-aware normalization layers [11], and more recently diffusion models [14, 43]. Similar deep learning-based strategies have been since proposed for defocus [1] and optical aberration [9] correction. These strategies involve both collecting large real-world supervisory datasets and designing ad-hoc models. In this work, we combine knowledge of a certain kind of blur, *e.g.,* camera shake, out-of-focus blur, optical aberration, and blind deblurring networks by grouping patches from an input blurry image. We explicitly make them interact within an architecture whose design is inspired by the burst deblurring approach of [3], yet for restoring a *single* image.

Using multiple images or patches from a single image to disambiguate restoration is common in image processing. Image collaboration is crucial for system-specific degradation such as vignetting correction [27], camera calibration [46] or fixed pattern noise estimation [8]. Nevertheless the most notable example of collaborative image processing is the BM3D [13] denoising algorithm that gathers patches with similar aspect to denoise them together. In the same spirit burst deblurring methods [3, 15] combine multiple frames depicting the same underlying sharp image but different blurs. Our approach is a single-image method that can be seen as the *dual* problem of burst deblurring: we select patches of different underlying sharp contents but sharing identical or similar blurs. This aims at obtaining information on the blur in as many directions as possible thanks to a greater variety of directional gradients from the different images/patches to better predict a blur kernel, a general idea considered in previous works [16, 17, 20, 32]. To our knowledge this philosophy is explicitly applied to single-image blind deblurring via neural networks for the first time in this paper.

3. Theoretical background

To motivate our approach we review the problem of kernel estimation from several samples. Let $x_1, x_2, \ldots, x_N$ be N sharp images, $y_1, y_2, \ldots, y_N$ corresponding blurry images where $y_n = k * x_n + \varepsilon_n$ for n in $\{1, \ldots, N\}$, k a blur kernel, and the ε_n's instances of noise. From the pairs (x_n, y_n) ($n = 1, \ldots, N$), we estimate the blur kernel k by minimizing a regularized ℓ_2 energy function:

$$\min_k \frac{1}{N} \sum_{n=1}^{N} \|y_n - k * x_n\|_2^2 + \lambda \|k\|_2^2. \tag{1}$$

Its unique minimizer is obtained in the Fourier domain by

$$\widehat{K} = \sum_{n=1}^{N} X_n^* Y_n / \left(\sum_{n=1}^{N} X_n^* X_n + \lambda N \right), \tag{2}$$

where the capital letters X_n, Y_n and K denote the Fourier transforms of x_n, y_n and k, the $*$ superscript denotes the complex conjugate, and λ is a positive regularization weight. In the Fourier domain, the multiplication and division are entrywise. In this setting, even if one image X_n has a zero at a given frequency, the average of several images would likely not be zero, unless the frequency is removed by the blur kernel. When $N = 1$ instead, if there is ambiguity at some frequencies (the blur may be oriented in the same direction as an edge for instance), then it is nearly impossible to recover the correct frequency of the blur kernel. On the spatial domain, considering more images thus boils down to gathering as much information on the oriented gradients as possible to disambiguate the blur from the signal, a strategy at the core of certain blind deblurring techniques [16, 20].

Let us illustrate how important it is to use N image pairs *together* using Eq. (2). We select the 512×512 central crops of the 64 first RGB test images from the DIV2K dataset [2] as sharp images x_n ($n = 1, \ldots, 64$) in order to have all the images at the same format. We blur them with the 8 kernels from the Levin dataset [29], and add 1% white Gaussian noise. Figure 1 illustrates the increase of the accuracy of the kernel support with respect to N for the 8 kernels from [29]. The results confirm that as the number of images in the stack increases, the performance improves significantly, leading to a nearly perfect reconstruction, even for a challenging motion blur kernel. Notably, the performance saturates at a relatively small stack size of $N = 8$. This analysis highlights the dramatic improvement that collaboration among images with the same blur can bring towards better deblurring.

4. Proposed method

In this section we address the more realistic blind deblurring case, where only the blurry images $y_1, \cdots y_N$ are available.

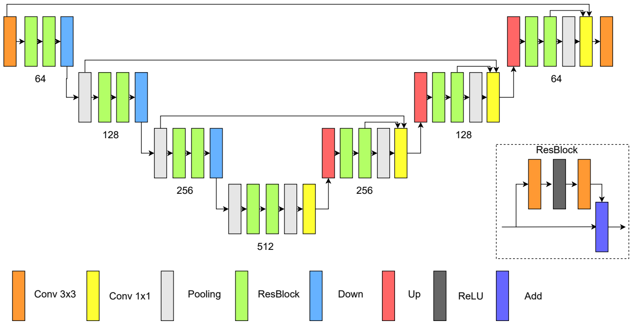


Figure 2. Upgraded UNet we use for our experiments. At each stage in the encoder and decoder we add a pooling layer following [3]. The pooling strategy is implemented with either averaging, a lambda layer [6] or a self-attention layer [41].

We thus cannot rely directly on Eq (2) to improve the kernel estimation. We instead leverage the collaboration of N blurry images, as highlighted in the previous section, in order to better capture the structure of the blur while at the same time deblurring. This collaborative processing results in an efficient single-image blind deblurring neural network.

4.1. Collaborative architecture

We propose a neural network f_θ with parameter θ that features two notable changes compared to typical single-image blind deblurring networks, *e.g.,* [33]: (i) N inputs and outputs instead of a single one, and (ii) an inner collaboration layer combining the N feature maps. When N is 1, the proposed framework boils down to the classical single-image blind deblurring approach.

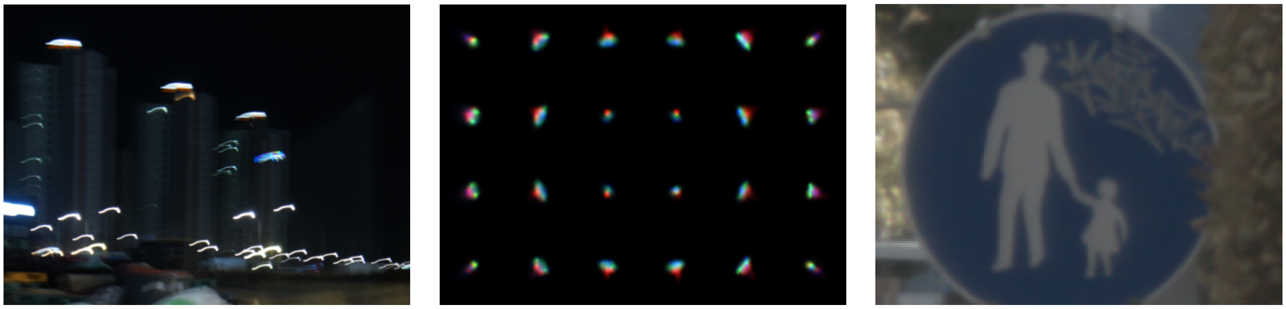
Input and output. We propose a neural network that processes in a single forward pass N images $\mathcal{Y}_N = \{y_1, \ldots, y_N\}$ containing similar blurs (not necessarily the exact same for each image), and predicts N sharp versions $\widehat{\mathcal{X}}_N = \{\widehat{x}_1, \ldots, \widehat{x}_N\}$. In practice these sets are implemented as 4D tensors concatenating all the RGB images of the stacks, the first dimension being of size N. Let f be such a network and θ its parameter, then the inference reads

$$\widehat{\mathcal{X}}_N = f_\theta(\mathcal{Y}_N). \tag{3}$$

During training, the N restored patches have collaborated and can thus be supervised individually since what matters in the end is the quality of each individual image $\widehat{x}_n$ ($n = 1, \ldots, N$). Provided training pairs of sets $\mathcal{X}_N^{(m)}$ and $\mathcal{Y}_N^{(m)}$ ($m = 1, \ldots, M$), we learn the parameter θ by minimizing:

$$\min_\theta \frac{1}{MN} \sum_{m=1}^{M} \sum_{n=1}^{N} \ell(\widehat{x}_n^{(m)}, x_n^{(m)}), \tag{4}$$

where ℓ is a pixelwise loss. In this work, we adopt the ℓ_1 distance as supervising loss function. The images $\widehat{x}_n^{(m)}$ and



(a) Camera shake [38]. (b) Optical aberration [5]. (c) Fusion blur [25].

Figure 3. Three real instances of blur: camera shake from the RealBlur dataset [38] (scene 118, image 2), a subset of the local calibrated optical aberrations from the Canon EF24mm f/1.4L USM opened at f/2.8, calibrated in [5], and the result of the ×2 multi-frame super-resolution online demo of [25]. For shake the light streaks suggest the blur is roughly the same everywhere, thus patches can be sampled uniformly. The same holds for fusion blur where the fused image globally lacks of sharpness. The aberrations are all unique but roughly follow a central symmetry, thus easy to sample.

$y_n^{(m)}$ are the n-th elements of respectively $\widehat{\mathcal{X}}_N^{(m)}$ and $\mathcal{Y}_N^{(m)}$.

Inner collaboration. We follow the approach of Aittala and Durand [3]. This approach applies the same convolution layers to each one of the N images of the stack and combine the knowledge from different images in the same stack via pooling of the individual blurry images' representations after a given convolutional or attention layer in a network. Let e_n be the representation of the n-th blurry image y_n $(n = 1, \ldots, N)$ after this given layer of f_θ. We implement the inner collaboration layer by a pooling function p operating on the e_n along the stack dimension, and that returns a global representation g of the stack:

$$g = p(\{e_1, \ldots, e_N\}). \tag{5}$$

The feature g has the same dimension as an individual local feature e_n $(n = 1, \ldots, N)$. In [3], in the context of burst deblurring, g is supposed to extract a representation of the underlying sharp signal since all the blurry frames in the burst have different blurs but share the same underlying sharp content. In our case the roles of the images and blurs are reversed: we have images of different contents sharing similar blurs in our stack. Consequently, we expect the global information compiled in g via the pooling function p to be related to the blur. Lastly, the local features e_n $(n = 1, \ldots, N)$ are updated through the merge with the global representation g via a 1×1 convolution layer:

$$e_n \leftarrow \mathtt{conv}_{1\times 1}(\{e_n, g\}), \quad \forall n \in \{1, \ldots, N\}. \tag{6}$$

That way, we expect the upgraded feature map e_n to be guided by some information on the blur, ultimately improving the deblurring ability of f_θ.

The function p has not been explicitly defined so far on purpose. Such a function should take a stack of spatial feature maps and return a single spatial feature map. Since no assumption is made on p, it can be either learning-free or learnable. Classical pooling strategies such as the mean or max functions are used in [3] and are drop-in candidates to extract g in our context. We also explore in this paper how to learn p by implementing it with the lambda layer from [6] or the self-attention (SA) module [41]. The learning-free approach is important to build small models that may be deployed on devices like smartphones [31], whereas the learnable modules aim at better performance at the cost of additional computations. We benchmark the different candidates for p in Section 5. Figure 2 illustrates the upgraded architecture with the pooling strategy.

4.2. Gathering images with similar underlying blurs

The theoretical background of the previous section and its proposed integration within a neural network assume that N images y_n $(n = 1, \ldots, N)$ degraded with similar blurs are given. For camera shake [44], mild blurs [16] (covering out-of-focus and sharpening), and optical aberrations [18], such patches may be handpicked from a *single* image. For camera shake the blur smoothly varies across the field of view [44], and for "reasonable" shake like small translations featured in the Realblur dataset [38], it boils down to a uniform blur kernel applied to the whole image. Rough uniformity of the blur is also a typical assumption for modeling lack of sharpness in the image, *e.g.,* because of a failing autofocus or the interpolation blur of a preceding multi-frame algorithm [28].

For instance, inevitable blur may come from optical aberrations [17, 23], for this radial blur very similar blurs may be found at symmetric locations on the field of view [39]. Sharpening is another instance of important brick in most ISP pipelines nowadays [16], and consists in removing small Gaussian-like blurs caused by slight out-of-focus or multi-frame fusion algorithms. The blur may vary but it may be considered roughly similar across the

Table 1. Average PSNR over 400 images blurred with isotropic Gaussian blur. UNet-T achieves the same performance as UNet for $N \geq 4$ despite having 4 times more parameters.

N	UNet	UNet-T
1	32.75	32.46 (-0.29)
2	32.91	32.75 (-0.16)
4	32.92	32.92 (+0.00)
8	33.12	33.13 (+0.01)
16	33.02	33.07 (+0.05)

field of view, for instance to remove the fusion blur introduced by multi-frame algorithms [17]. A third common category is camera shake [44] during exposure, resulting in global motion blur across the image. A single blur kernel [38] or smoothly varying blurs [44] may model the whole shake, thus validating the assumption of similar kernels across the image. An illustration is shown in Figure 3. In the three common sorts of blur list above, grouping patches with similar blur is thus feasible by leveraging the properties of the blur. Collecting patches that way amounts to finding the most relevant patches for a given deblurring task, which is a sort of handcrafted attention or "attention-without-attention". This patch selection could be done using CNNs or attention, but at the cost of either very deep models or important computation (corresponding patches may be hundreds or thousands of pixels apart), all that for computing something that could be known in advance.

Note that we have not discussed defocus blur where local kernels of similar aspect might be easily grouped if the depth of the scene is known [21]. Although depth estimation could be provided by recent monocular depth estimators, *e.g.,* [37], we keep this multimodal approach for future work and focus instead on blurs where the grouping can be done manually grouping can be done manually as for the examples in Figure 3.

5. Experiments

Experiments were all run on a single 16Gb NVIDIA V100 graphic card.

5.1. Collaborative model

In this work, we use a UNet model that is a general architecture used as the foundation of many practical deblurring models, *e.g.,* [9, 26]. Since it is an all-purpose model, it has no specific bias for removing blur in contrast to the state-of-the-art CNNs [10, 40]. It is thus an adequate model to measure the impact of the proposed collaborative scheme. We introduce two variants called UNet and UNet-T (for tiny) embedding respectively 4 and 3 downsampling/upsamling layers in the encoder/decoder: The initial number of feature maps C is 64 and the respective bottleneck sizes of these models are 512 and 256 channels. These models have respectively around 17M and 4.3M parameters. Before each down/upsampling module we place a pooling p to enforce collaboration.

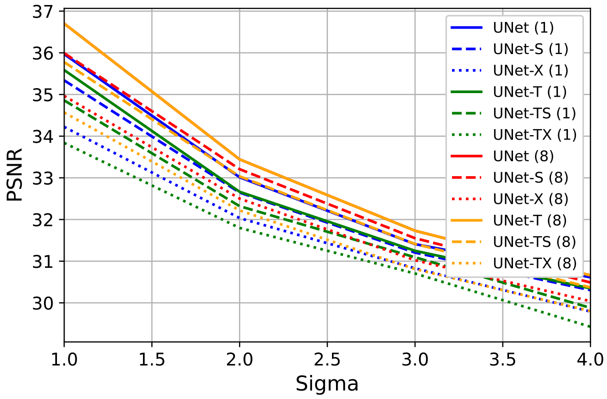


Figure 4. PSNR of UNet and UNet-T for 3 widths. N is in parenthesis. For each width, the models with $N = 8$ are above. The plain red curve is covered by the orange one.

So far we have only presented the broad idea that features should be shared within p, but we did not delve into details. We compare the max pooling approach of [3] for burst deblurring, the lambda layer [6] and a self-attention (A) layer as in the Transformer architecture [41]. The lambda layer is implemented with 4 feature channels. The three-layer perceptron of the self-attention layer is shaped as an inverted bottleneck. We also evaluated the feature mean to implement p but observed, as in [3], that it leads to the same results as the max pooling.

5.2. Validation on anisotropic Gaussian blur

We start with 2D anisotropic Gaussian blur kernels that may approximate several instances of real-world blur such as lens blur [23], defocus [21], and translational motion blur [16]. Evaluating our approach on Gaussian blur is thus a simple manner to validate our approach in a controlled setting that corresponds to many real-world blurring scenarios. We train UNet and UNet-T on 128 × 128 crops randomly sampled from the 800 training images of the DIV2K dataset [2]. We randomly flip and rotate the patches prior to blurring them with Gaussian blur kernels of standard deviation along the two principal axes uniformly sampled in $[0.3, 4]^2$, *i.e.,* up to a 33×33 blur spot, and the orientation is uniformly sampled in $[0, 2\pi)$ (same model as in [16]). We add moderate Gaussian noise with standard deviation randomly sampled in $[0.5, 2]/255$ after blurring.

Image stack size N. In order to evaluate the choice of the stack size N, we train networks with input stacks of blurry

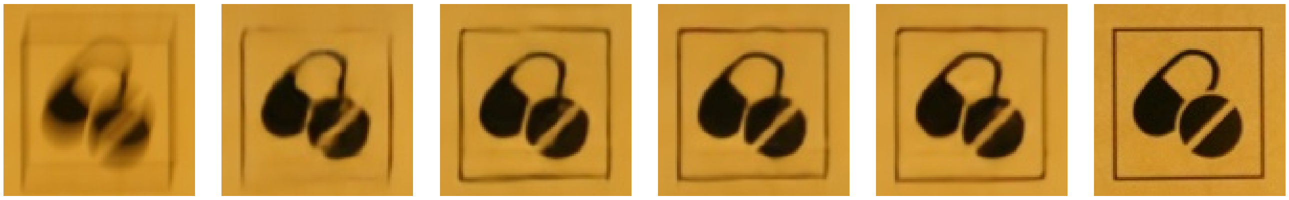

(a) Blurry. (b) UNet-T. (c) UNet-T (A). (d) UNet (A). (e) MIMOUNet. (f) Target.

Figure 5. An example from Realblur-J [38] deblurred with UNet-T, the versions of UNet and UNet-T embedding attention with $N = 16$ and MIMOUNet [10]. The collaboration improves the deblurring accuracy of UNet, to match that of MIMOUNet.

images sharing the same blur with a unique stack size N in $\{1, 2, 4, 8, 16\}$. We choose the max pooling strategy to not introduce additional learnable parameters, and isolate the impact of using several patches together. We train the models for 250k iterations with the Adam optimizer and initial learning rate of 5×10^{-5} that is reduced by 0.5 after each 40k iteration until reaching 10^{-6} where we observe convergence. We evaluate the models on sets of 100 images blurred with an anisotropic Gaussian blur kernel with standard deviation σ in $\{1, 2, 3, 4\}$, and with additional $0.5/255$ Gaussian noise. We show in Table 1 that the performance of both models (UNet and UNet-T) grows with N. For $N \geq 4$, the performance of both models is similar despite UNet having 4 times more parameters than UNet-T. All the results are averaged over 3 shuffles of the test images to draw different images per N-sized stack across different runs, but the error bars are marginal and thus not reported.

Reducing UNet. Since using N images instead of 1 actually compensates the difference of depth and parameters between UNet and UNet-T for $N \geq 4$, we also verify that by reducing the width of the UNet, using $N = 8$ images helps to maintain good performance. Besides training UNet (resp. UNet-T) with the original number of channels per feature like in the previous experiment, we propose the Slim and Extra-Slim variants dubbed UNet-S and UNet-X (resp. UNet-TS and UNet-TX) where the number of channels for each feature map is respectively divided by 2 and 4, *i.e.,* the bottleneck layer of UNet initially having 512 channels has now respectively 256 and 128 channels. For $N = 1$, these models have respectively 4.4M, 1.1M, 1.1M and 270K parameters, and about 5% more for $N > 1$ (because of the 1×1 convolution layers in Eq. (6)). We train the four new variants of UNet and UNet-T with the same procedure, evaluate them on the same 400 test images as previously, and show the average PSNR per value of σ in Figure 4. It can be seen that the models with $N = 8$ systematically achieve better results than their $N = 1$ counterparts. Remark that UNet-TX for $N = 8$ achieves results similar to models that have ×4 more parameters, and is between +0.3 and +0.8dB above its $N = 1$ counterpart. This observation suggests that the learning-free max pooling operation helps to design lightweight blind deblurring networks with performance on par with larger ones. UNet-TX has a number of parameters comparable to that of the lightweight demosaicking networks proposed in [31], which is akin to edge computing. This suggests that UNet-TX may be a valid candidate to replace mild blur deblurring algorithms, *e.g.,* [16, 17], in terms of parameters and thus energy consumption. More quantitative results are in the supplementary material.

5.3. Practical applications

We illustrate the method on the three practical problems listed in Figure 3: optical aberration correction, camera shake compensation, and image sharpening.

Camera shake. An important source of blur in personal photography (especially with handheld cameras) is camera shake. We train on the RealBlur-J dataset [38], composed 3760 sharp/blurry image pairs of static scenes taken with complex natural camera motions, a UNet with and without the collaborative layer (training for $N = 1$), and the MIMOUNet [10] specifically designed for this task.

Our goal here is not to propose new state-of-the-art motion blur models, but rather use the RealBlur dataset as a real-world quantitative benchmark to quantify the impact of the proposed collaborative scheme. We show in Table 2 quantitative results on the test sets of the RealBlur datasets composed of 980 blurry/sharp image pairs. We run UNet and UNet-T by splitting 512×512 random crops from the training set into a unique stack of size N in $\{1, 4, 16\}$ with 25% of overlap to take into account the patches will be stitched back into a full-sized image for evaluation with the protocol of [38]. During training, we supervise each patch of the stack with the corresponding patch in the target with the loss (4). At test time, we start from the full image, slice it into N patches, and stitch together the predicted sharps variants into a full sharp estimate with the same windowing approach as in [39].

We benchmark different choices of pooling layer p: max (M), lambda (L) and self-attention (A). We use the same training strategy as MIMOUNet and also retrain the latter for fair comparison: We train for 1k epochs with the Adam optimizer with initial learning rate set to 0.0001 and de-

Table 2. Results on Realblur-J [38]. The models with * are those reported in [38] and are used as setters of the expected ballpark of metrics. MIMOUNet [10] is retrained following our protocol. Bold and underlined numbers indicate which variants of UNet are the best and second-to-best per UNet model. The difference with the corresponding UNet(-T) with $N = 1$ is shown in parenthesis.

Method	PSNR	SSIM	LPIPS	Params
DeblurGan-v2* [24]	29.69	0.870	-	-
SRN* [40]	31.38	0.909	-	6.8M
MIMOUNet [10]	30.32	0.897	0.097	6.8M
UNet-T ($N = 1$)	29.30	0.870	0.133	4.3M
UNet-T ($N = 4$, M)	29.88	0.883	0.121	4.6M
UNet-T ($N = 16$, M)	30.33	0.893	0.108	4.6M
UNet-T ($N = 4$, L)	29.96	0.885	0.117	5.6M
UNet-T ($N = 16$, L)	30.45	0.896	0.104	5.6M
UNet-T ($N = 4$, A)	30.13	0.888	0.111	6.4M
UNet-T ($N = 16$, A)	**30.63**	**0.900**	**0.098**	6.4M
UNet ($N = 1$)	30.16	0.893	0.104	17.7M
UNet ($N = 4$, M)	30.45	0.897	0.104	18.7M
UNet ($N = 16$, M)	30.62	0.900	0.099	18.7M
UNet ($N = 4$, L)	30.61	0.900	0.098	22.9M
UNet ($N = 16$, L)	30.78	0.904	0.094	22.9M
UNet ($N = 4$, A)	30.73	0.904	0.092	26.4M
UNet ($N = 16$, A)	**30.98**	**0.908**	**0.087**	26.4M

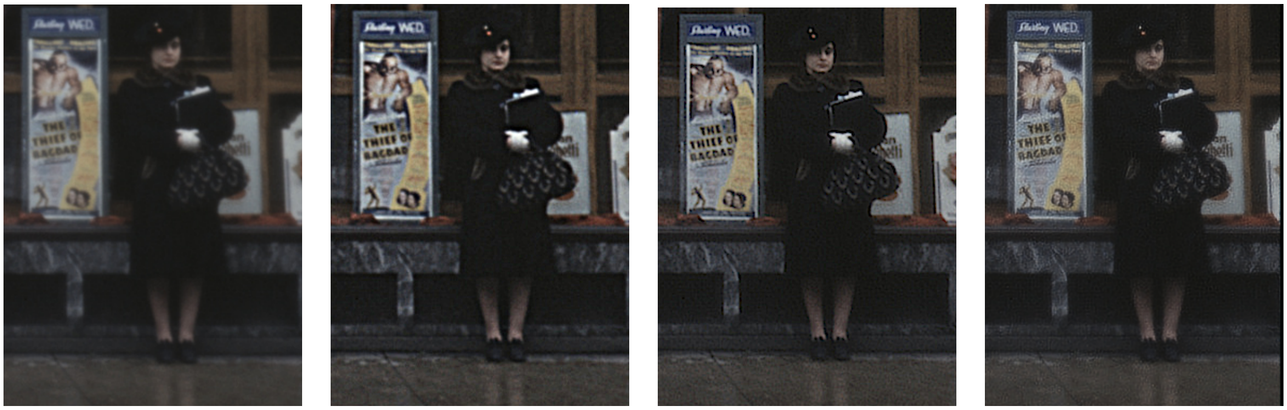
Table 3. Optical aberration removal for two lenses calibrated in [5]. The PSNR is reported from three different locations on the field of view: the edge E, the intermediate I and the central C regions. The difference between using 1 or 4 images is shown in parenthesis. Setting N to 4 helps compensating the reduction of the width and parameters.

Loc.	Method	Lens #1	Lens #2	Params
E	UNet-TX ($N = 1$)	30.58	32.83	270K
E	UNet-TX ($N = 4$, M)	31.02	33.45	284K
E	UNet-TS ($N = 1$)	31.60	34.10	1.1M
E	UNet-TS ($N = 4$, M)	31.82	34.57	1.1M
E	UNet-T ($N = 1$)	**32.21**	34.88	4.3M
E	UNet-T ($N = 4$, M)	**32.27**	**35.32**	4.6M
I	UNet-TX ($N = 1$)	34.45	34.89	270K
I	UNet-TX ($N = 4$, M)	34.79	35.03	284K
I	UNet-TS ($N = 1$)	35.12	35.55	1.1M
I	UNet-TS ($N = 4$, M)	35.49	35.73	1.1M
I	UNet-T ($N = 1$)	35.89	**36.22**	4.3M
I	UNet-T ($N = 4$, M)	**36.12**	**36.23**	4.6M
C	UNet-TX ($N = 1$)	36.94	36.71	270K
C	UNet-TX ($N = 4$, M)	37.57	36.92	284K
C	UNet-TS ($N = 1$)	38.02	37.93	1.1M
C	UNet-TS ($N = 4$, M)	38.38	37.96	1.1M
C	UNet-T ($N = 1$)	39.10	**38.99**	4.3M
C	UNet-T ($N = 4$, M)	**39.28**	**38.99**	4.6M

cayed by 0.5 every 200 epoch. The batch size is set to 8. We see that increasing the stack size N within the *same* patch and more refined pooling p benefit the deblurring accuracy for all metrics when using a UNet not initially designed for deblurring, in contrast to [40] and [10]. For instance Remark a 17M-parameter UNet cannot beat MIMOUNet [10], but after being upgraded, the performances are boosted by a margins up to +1.33dB. UNet-T with p implemented with self-attention and $N = 16$ notably beats MIMONet with less parameters. We also observe that both refining p and increasing N increase the performance, validating our hypothesis. An example is shown in Figure 5.

Optical aberrations. We leverage the near central symmetry of aberrations around the optical center of the lens [39] by sampling at $N = 4$ locations on the field of view, one for each quadrant of the Cartesian plane, (see Figure 3). Since no real-world pairs of aberrated and aberration-free image dataset exists, we generate synthetic data from two of the 70 PSFs calibrated in [5]. Each PSF consists of about 4,000 local RGB kernels accounting for both monochromatic and chromatic aberrations. We select the Canon EF16-35mm f/2.8L USM EI at shortest focal length and maximal aperture and the Canon EF24mm f/1.4L USM that are prone to aberrations after visual inspection, the former being poorer than the latter. We dub these lenses “Lens #1” and “Lens #2” respectively in our experiments. We convert from JPEG to pseudo-linear RGB images the 128×128 patches from the DIV2K images with the protocol of [7], and blur each color channel with the corresponding one from local filters sampled at random in the given PSF. We add $0.5/255$ Gaussian noise to account for noise residual after demosaicking in an ISP pipeline. We train for each PSF UNet-T, UNet-TS and UNet-TX for $N = 1$ and $N = 4$ with max pooling as collaborative strategy since in an ISP pipeline within a handheld camera, each module should be as lightweight as possible. We train for 500k iterations with the Adam optimizer with batch size of 16, and an initial learning rate of 10^{-4} decayed by 0.2 after 200k and 400k iterations.

Table 3 shows the PSNR for three locations on the lens field-of-view with growing loss of quality from the center to a corner. For both lenses the collaborative variant is always above the model with $N = 1$ with significant margins between +0.2 and +0.6dB for UNet-TX. We observe more important improvements for the 16mm lens than the 24mm. Since the former is a zoom lens, it has in average a poorer quality than a prime lens, thus benefiting more from collaboration, especially for UNet-TX, validating the assumption that collaboration helps to design practical lightweight networks. We can reasonably expect that on devices with worse optical quality like smartphones, collaboration may



(a) Blurry.

(b) Unsharp mask.

(c) Polyblur [16].

(d) UNet-TX.

Figure 6. Sharpening on the heritage image from [39]. The two classical methods have hyper-parameters to tune whereas our UNet-TX with $N = 8$ has not and produce a halo-free sharp result.

help even further.

Sharpening. In the absence of any benchmark evaluating sharpening algorithms, we simply run a qualitative comparison for sharpening. Since mild blurs may be reasonably approximated with Gaussians [17], we select the lightweight UNet-TX trained in Section 5.2 and compare it with Polyblur [16] and the unsharp mask algorithm, two approaches used on-device, for instance to postprocess multi-frame algorithms like in the context of super-resolution or high-dynamic range imaging [28]. The comparison is run on a heritage photograph from [39], and shown in Figure 6.

Discussion and limitations. We have shown that the hypothesis that similar blurs exist in blurry images and may be gathered is verified, and leads to improvements in each situation. Second, these improvements are obtained for a learning-free max pooling layer that leads to lightweight efficient models for sharpening and optical aberration correction. In these cases, $N = 4$ or $N = 8$ similar patches are enough. For more diverse blur families such as camera shake, learning-based pooling strategies and more important stack sizes, *e.g.,* $N = 16$ instead of 4, lead to significant boost, suggesting that in this case, the more patches and pooling capacity, the better.

Nevertheless the proposed approach has limits. Finding patches with similar blurs is not straightforward when the blur may vary non-continuously, in particular with segmentation-aware blurs such as the dynamic blur featured in the GoPro dataset [33], or depth-depending defocus blur [21]. Another problem may arise if not enough patches with the same blur are collected. We remarked during our experiments that important noise alters the collaboration, thus limiting the benefits of our approach in high-noise regimes. Yet, deblurring in such regime is an objective [4] beyond the scope of this paper. We have shown that the method is effective for three realistic blurring scenarios.

6. Conclusion

In this paper, we have presented a simple modification of existing CNNs for enhanced blind image deblurring. It consists in processing together images with the same latent blur to share the different features of the blur across the images within a network which has internal collaborative layers taking the form of feature maps pooling. Finding several images having similar latent blurs is verified to be possible for a wide range of practical single-image blind deblurring applications. Experiments on both synthetic and real-world images covering camera shake removal, optical aberration compensation and sharpening validate our approach and highlight the versatility of possible collaborating layers to design efficient models. This seems to establish a practical framework upon which building both lightweight and state-of-the-art architectures.

Acknowledgment. This work was partly financed by the DGA Astrid Maturation project SURECAVI ANR-21-ASM3-0002 and the ANR project IMPROVED ANR-22-CE39-0006-04. This work was performed using HPC resources from GENCI-IDRIS (grants 2023-AD011011801R3, 2023-AD011012453R2, 2023-AD011012458R2). Centre Borelli is also with Université Paris Cité, SSA and INSERM.

References

[1] Abdullah Abuolaim and Michael S. Brown. Defocus deblurring using dual-pixel data. In *European Conference on Computer Vision*, pages 111–126. Springer, 2020. 2

[2] Eirikur Agustsson and Radu Timofte. NTIRE 2017 challenge on single image super-resolution: Dataset and study.

In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 1122–1131, 2017. 3, 5

[3] Miika Aittala and Frédo Durand. Burst image deblurring using permutation invariant convolutional neural networks. In *European Conference on Computer Vision*, pages 748–764. Springer, 2018. 2, 3, 4, 5

[4] Jérémy Anger, Gabriele Facciolo, and Mauricio Delbracio. Blind image deblurring using the ℓ_0 gradient prior. *Image Processing On Line*, 9:124–142, 2019. 8

[5] Matthias Bauer, Valentin Volchkov, Michael Hirsch, and Bernhard Schölkopf. Automatic estimation of modulation transfer functions. In *IEEE International Conference on Computational Photography*, pages 1–12, 2018. 1, 4, 7

[6] Irwan Bello. LambdaNetworks: Modeling long-range interactions without attention. In *International Conference on Learning Representations*, pages 1–14. OpenReview.net, 2021. 3, 4, 5

[7] Tim Brooks, Ben Mildenhall, Tianfan Xue, Jiawen Chen, Dillon Sharlet, and Jonathan T. Barron. Unprocessing images for learned raw denoising. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11036–11045, 2019. 7

[8] Mo Chen, Jessica Fridrich, Miroslav Goljan, and Jan Lukás. Determining image origin and integrity using sensor noise. *IEEE Transactions on information forensics and security*, 3 (1):74–90, 2008. 2

[9] Shiqi Chen, Huajun Feng, Dexin Pan, Zhihai Xu, Qi Li, and Yue-ting Chen. Optical aberrations correction in postprocessing using imaging simulation. *ACM Transactions on Graphics*, 40(5):192:1–192:15, 2021. 2, 5

[10] Sung-Jin Cho, Seo-Won Ji, Jun-Pyo Hong, Seung-Won Jung, and Sung-Jea Ko. Rethinking coarse-to-fine approach in single image deblurring. In *IEEE/CVF International Conference on Computer Vision*, pages 4621–4630, 2021. 1, 2, 5, 6, 7

[11] Xiaojie Chu, Liangyu Chen, Chengpeng Chen, and Xin Lu. Improving image restoration by revisiting global information aggregation. In *European Conference on Computer Vision*, pages 53–71. Springer, 2022. 2

[12] Florent Couzinie-Devy, Jian Sun, Karteek Alahari, and Jean Ponce. Learning to estimate and remove non-uniform image blur. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1075–1082, 2013. 2

[13] Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen O. Egiazarian. Image denoising by sparse 3-D transform-domain collaborative filtering. *IEEE Transactions on Image Processing*, 16(8):2080–2095, 2007. 2

[14] Mauricio Delbracio and Peyman Milanfar. Inversion by direct iteration: An alternative to denoising diffusion for image restoration. *arXiv preprint arXiv:2303.11435*, 2023. 2

[15] Mauricio Delbracio and Guillermo Sapiro. Removing camera shake via weighted Fourier burst accumulation. *IEEE Transactions on Image Processing*, 24(11):3293–3307, 2015. 2

[16] Mauricio Delbracio, Ignacio Garcia-Dorado, Sungjoon Choi, Damien Kelly, and Peyman Milanfar. Polyblur: Removing mild blur by polynomial reblurring. *IEEE Transactions on Computational Imaging*, 7:837–848, 2021. 1, 2, 3, 4, 5, 6, 8

[17] Thomas Eboli, Jean-Michel Morel, and Gabriele Facciolo. Breaking down polyblur: Fast blind correction of small anisotropic blurs. *Image Processing On Line*, 12:435–456, 2022. 2, 4, 5, 6, 8

[18] Thomas Eboli, Jean-Michel Morel, and Gabriele Facciolo. Fast two-step blind optical aberration correction. In *European Conference on Computer Vision*, pages 693–708. Springer, 2022. 4

[19] Hongyun Gao, Xin Tao, Xiaoyong Shen, and Jiaya Jia. Dynamic scene deblurring with parameter selective sharing and nested skip connections. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3848–3856, 2019. 2

[20] Amit Goldstein and Raanan Fattal. Blur-kernel estimation from spectral irregularities. In *European Conference on Computer Vision*, pages 622–635. Springer, 2012. 1, 2, 3

[21] Samuel W. Hasinoff and Kiriakos N. Kutulakos. A layer-based restoration framework for variable-aperture photography. In *IEEE International Conference on Computer Vision*, pages 1–8, 2007. 2, 5, 8

[22] Zhe Hu and Ming-Hsuan Yang. Learning good regions to deblur images. *International Journal on Computer Vision*, 115(3):345–362, 2015. 1

[23] Eric Kee, Sylvain Paris, Simon Chen, and Jue Wang. Modeling and removing spatially-varying optical blur. In *IEEE International Conference on Computational Photography*, pages 1–8, 2011. 4, 5

[24] Orest Kupyn, Tetiana Martyniuk, Junru Wu, and Zhangyang Wang. DeblurGAN-v2: Deblurring (orders-of-magnitude) faster and better. In *IEEE/CVF International Conference on Computer Vision*, pages 8877–8886, 2019. 2, 7

[25] Jamy Lafenetre, Gabriele Facciolo, and Thomas Eboli. Implementing handheld burst super-resolution. *Image Processing On Line*, 13, 2023. 4

[26] Wei-Sheng Lai, Yichang Shih, Lun-Cheng Chu, Xiaotong Wu, Sung-Fang Tsai, Michael Krainin, Deqing Sun, and Chia-Kai Liang. Face deblurring using dual camera fusion on mobile phones. *ACM Transactions on Graphics*, 41(4): 148:1–148:16, 2022. 5

[27] Jean-François Lalonde, Srinivasa G Narasimhan, and Alexei A Efros. What do the sun and the sky tell us about the camera? *International Journal of Computer Vision*, 88 (1):24–51, 2010. 2

[28] Bruno Lecouat, Thomas Eboli, Jean Ponce, and Julien Mairal. High dynamic range and super-resolution from raw image bursts. *ACM Transactions on Graphics*, 41(4):38:1–38:21, 2022. 4, 8

[29] Anat Levin, Yair Weiss, Frédo Durand, and William T. Freeman. Understanding and evaluating blind deconvolution algorithms. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1964–1971, 2009. 1, 2, 3

[30] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *IEEE/CVF International Conference on Computer Vision*, pages 9992–10002, 2021. 1

[31] Karima Ma, Michaël Gharbi, Andrew Adams, Shoaib Kamil, Tzu-Mao Li, Connelly Barnes, and Jonathan Ragan-Kelley.

Searching for fast demosaicking algorithms. *ACM Transactions on Graphics*, 41(5):172:1–172:18, 2022. 4, 6

[32] Tomer Michaeli and Michal Irani. Blind deblurring using internal patch recurrence. In *European Conference on Computer Vision*, pages 783–798. Springer, 2014. 2

[33] Seungjun Nah, Tae Hyun Kim, and Kyoung Mu Lee. Deep multi-scale convolutional neural network for dynamic scene deblurring. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 257–265, 2017. 1, 2, 3, 8

[34] Jinshan Pan, Deqing Sun, Hanspeter Pfister, and Ming-Hsuan Yang. Deblurring images via dark channel prior. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(10):2315–2328, 2018. 1

[35] Dongwon Park, Dong Un Kang, Jisoo Kim, and Se Young Chun. Multi-temporal recurrent neural networks for progressive non-uniform single image deblurring with incremental temporal training. In *European Conference on Computer Vision*, pages 327–343. Springer, 2020. 2

[36] Sung Hee Park and Marc Levoy. Gyro-based multi-image deconvolution for removing handshake blur. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3366–3373, 2014. 1

[37] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *IEEE/CVF International Conference on Computer Vision*, pages 12159–12168, 2021. 5

[38] Jaesung Rim, Haeyun Lee, Jucheol Won, and Sunghyun Cho. Real-world blur dataset for learning and benchmarking deblurring algorithms. In *European Conference on Computer Vision*, pages 184–201. Springer, 2020. 2, 4, 5, 6, 7

[39] Christian J. Schuler, Michael Hirsch, Stefan Harmeling, and Bernhard Schölkopf. Blind correction of optical aberrations. In *European Conference on Computer Vision*, pages 187–200. Springer, 2012. 1, 2, 4, 6, 7, 8

[40] Xin Tao, Hongyun Gao, Xiaoyong Shen, Jue Wang, and Jiaya Jia. Scale-recurrent network for deep image deblurring. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 8174–8182, 2018. 2, 5, 7

[41] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008, 2017. 1, 3, 4, 5

[42] Zhendong Wang, Xiaodong Cun, Jianmin Bao, Wengang Zhou, Jianzhuang Liu, and Houqiang Li. Uformer: A general U-shaped transformer for image restoration. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17662–17672, 2022. 2

[43] Jay Whang, Mauricio Delbracio, Hossein Talebi, Chitwan Saharia, Alexandros G. Dimakis, and Peyman Milanfar. Deblurring via stochastic refinement. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16272–16282, 2022. 2

[44] Oliver Whyte, Josef Sivic, Andrew Zisserman, and Jean Ponce. Non-uniform deblurring for shaken images. *International Journal on Computer Vision*, 98(2):168–186, 2012. 1, 2, 4, 5

[45] Li Xu, Shicheng Zheng, and Jiaya Jia. Unnatural L_0 sparse representation for natural image deblurring. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1107–1114, 2013. 2

[46] Zhengyou Zhang. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11):1330–1334, 2000. 2