Lucas-Kanade Reloaded: End-to-End Super-Resolution from Raw Image Bursts

Bruno Lecouat\textsuperscript{1,2} Jean Ponce\textsuperscript{1,3} Julien Mairal\textsuperscript{2}

\textsuperscript{1}Inria and DIENS (ENS-PSL, CNRS, Inria), Paris, France
\textsuperscript{2}Univ. Grenoble Alpes, Inria, CNRS, Grenoble INP, LJK, 38000 Grenoble, France
\textsuperscript{3}Center for Data Science, New York University, New York, USA
\texttt{firstname.lastname@inria.fr}

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{figure1.png}
\caption{\times 4 super-resolution results obtained from a burst of 30 raw images acquired with a handheld Panasonic Lumix GX9 camera at 12800 ISO for the top image and 25600 for the bottom image. Dcraw performs basic demosaicking.}
\end{figure}

\section*{Abstract}

This presentation addresses the problem of reconstructing a high-resolution image from multiple lower-resolution snapshots captured from slightly different viewpoints in space and time. Key challenges for solving this super-resolution problem include (i) aligning the input pictures with sub-pixel accuracy, (ii) handling raw (noisy) images for maximal faithfulness to native camera data, and (iii) designing/learning an image prior (regularizer) well suited to the task. We address these three challenges with a hybrid algorithm building on the insight from [45] that aliasing is an ally in this setting, with parameters that can be learned end to end, while retaining the interpretability of classical approaches to inverse problems. The effectiveness of our approach is demonstrated on synthetic and real image bursts, setting a new state of the art on several benchmarks and delivering excellent qualitative results on real raw bursts captured by smartphones and prosumer cameras. Our code is available at \texttt{https://github.com/bruno-31/lkburst.git}.

\section{1. Introduction}

The problem of reconstructing high-resolution (HR) images from lower-resolution (LR) ones comes in multiple flavors, that may significantly differ from each other in both technical detail and overall objectives. When a single LR image is available, the corresponding inverse problem is severely ill-posed, requiring very strong priors about the type of picture under consideration [18, 47]. For natural images, data-driven methods based on convolutional neural networks (CNNs) have proven to be very effective [26, 44]. Generative adversarial networks (GANs) have also been used to synthesize impressive HR images that may, how-
ever, contain “hallucinated” high-frequency details [9, 28].

In the true super-resolution setting [31, 39, 47], where multiple LR frames are available, HR details are present in the data, but they are spread among multiple misaligned images, with technical challenges such as recovering sub-pixel registration, but also the promise of recovering veridical information in applications ranging from amateur photography to astronomy, biological and medical imaging, microscopy imaging, and remote sensing.

Videos are of course a rich source of multiple, closely-related pictures of the same scene, with several recent approaches to super-resolution in this domain, often combining data-driven priors from CNNs with self-similarities between frames [21, 27, 43]. However, most digital videos are produced by a complex pipeline mapping raw sensor data to possibly compressed, lower-resolution frames, resulting in a loss of high-frequency details and spatially-correlated noise that may be very difficult to invert [12]. With the ability of modern smartphone and prosumer cameras to record raw image bursts, on the other hand, there is a new opportunity to restore the corresponding frames before the image signal processor (ISP) of the camera produces irretrievable damage [4, 45]. This is the problem addressed in this presentation, and it is challenging for several reasons: (i) images typically contain unknown motions due to hand tremor, making subpixel alignment difficult; (ii) converting noisy raw sensor data to full-color images is in itself a difficult problem known as demosaicking [22, 25]; and (iii) effective image priors are often data driven, thus requiring a differentiable estimation procedure for end-to-end learning.

In this paper, we jointly address these issues and propose a new approach that retains the interpretability of classical inverse problem formulations while allowing end-to-end learning of models parameters. This may be seen as a bridge between the “old world” of signal processing and the “brave new one” of data-driven black boxes, without sacrificing interpretability: On the one hand, we address an inverse problem with a model-based optimization procedure alternating motion and HR image estimation steps, directly building on classical work from the 1980s [1, 29] and 1990s [16]. On the other hand, we also fully exploit modern technology in the form of a plug-and-play prior [6, 42] that gracefully mixes deep neural networks with variational approaches. In turn, unrolling the optimization procedure [7, 25, 48] allows us to learn the model parameters end to end by using training data with synthetic motions [4].

Since aliasing produces low-frequency artefacts associated with undersampled high-frequency components of the original signal, it is typically considered a nuisance, mo-

---

1 “Single-image super-resolution” has become a popular nickname for single-image upsampling under strong priors; here, we use the classical definition of super-resolution from multiple LR snapshots [39, 47].

2 Image bursts acquired on a tripod may also present subpixel misalignments in practice due to floor vibrations, as observed in our experiments.

3 There is, however, a trend today toward removing these filters, as in the prosumer camera used in some of our experiments with real images.
and the model regularized by bilateral total variation of Far-siu et al. [13]. The image formation parameters are either be assumed to be known a priori through calibration, or estimated jointly with the HR image. In general, inter-frame motion can either be estimated separately, or be treated as an integral part of the super-resolution problem [2, 16], thus avoiding motion estimation between LR frames, whose accuracy may be affected by undersampling [40]. The method proposed in the rest of this paper combines the best of both worlds since it performs joint estimation while aligning the LR frames with the reconstructed HR image.

Learning-based approaches. In this context, the multi-frame case has received less attention than its single-image counterpart, for which several loss functions and architectures have been proposed [9, 28, 48]. Most multi-frame algorithms focus on video super-resolution. Model-based techniques learn non-uniform interpolation or motion compensation using convolutional neural networks [37] but the most successful approaches so far are model free, leveraging instead diversity with 3D convolutions or attention mechanisms [21, 43]. Learning-based methods have also been used in remote sensing applications, using 3D convolutions [32] or joint registration/fusion architectures [8] for example. Finally, and closer to our work, Bhat et al. [4] have recently proposed a network architecture for raw burst super-resolution, together with a very interesting dataset featuring both synthetic and real images for training and testing. It is important to note that learning-based approaches to super-resolution are typically trained on synthetically generated LR images [30], a strategy that may not generalize well to real photographs unless great care is taken in modeling the image corruption process [5]. Learning super-resolution models from real LR/HR image pairs is quite challenging since it requires in general using separate cameras with different lenses and spatial resolution, with inevitable spatial and spectral misalignments. As shown by our experiments, our method, although trained from synthetic LR images, gives excellent results with real bursts taken from different smartphones and cameras. Leveraging real images at training time is, for now, left for future work.

3. Proposed Approach

This section presents the three main components of our approach: its image formation model, an optimization procedure for solving the corresponding inverse problem, and its unrolled implementation in a feedforward architecture whose parameters can be learned end to end.

3.1. Image formation model

Image acquisition in a digital camera starts from an instantaneous irradiance function \( f_{\gamma,t} : [0, 1]^2 \rightarrow \mathbb{R}^+ \) defined on a continuous retinal domain with nonnegative values, such that \( f_{\gamma,t}(u) \) is the spectral irradiance value at point \( u \), time \( t \), and wavelength \( \gamma \), accounting for blur due to optics, atmospheric effects, etc. The camera sensor integrates \( f_{\gamma,t} \) in the spatial, time, and spectral domains to construct a raw digital image \( y : [1, \ldots, n]^2 \rightarrow \mathbb{R}^+ \), where each pixel's spectral response is typically dictated by the \( 2 \times 2 \) RGGB Bayer pattern, with twice as many measurements for the green channel than for the red and blue ones [22]. Modern cameras turn the raw image \( y \) into a full blown, three-channel RGB image \( x \) with the same spatial resolution through an interpolation process called demosaicking.

In practice, we do not have access to \( f_{\gamma,t} \) to use as ground truth for learning an image restoration process, even when an accurate model of the \( f_{\gamma,t} \rightarrow x \) map is available. Thus, we model instead the process \( x \rightarrow y_k \), where \( x \) is a latent high-resolution (HR) image we wish to recover, and the low-resolution (LR) images \( y_k (k = 1, \ldots, K) \) have been observed in a burst of length \( K \). We assume that \( x \) is sharp, without any blur, and noiseless. The burst images are obtained through the following forward model (Figure 3):

\[
y_k = DBW_{p_k} x + \varepsilon_k \text{ for } k = 1, \ldots, K,
\]

where \( \varepsilon_k \) is some additive noise. Here, both the HR image \( x \) and the frames \( y_k \) of the burst are flattened into vector form. The operator \( W_{p_k} \) parameterized by \( p_k \) warps \( x \) to compensate for misalignments between \( x \) and \( y_k \) caused by camera or scene motion between frames, assumed here to be a 6-parameter affine transformation of the image plane, then resamples the warped image to align its pixel grid with that of \( y_k \). Finally, the corresponding HR image is blurred to account for integration over both space (the LR pixel area, using either simple averaging or, as in the figure, a Gaussian filter) and time (accounting for camera and/or scene motion during exposure), and it is finally downsampled in both the spatial and spectral domains by the operator \( D \), with an (a priori) arbitrary choice of where to pick the sample from (pixel corner or center for example), the spatial part corresponding to selecting one of the three RGB values to assemble the raw image. It will prove convenient in the sequel to
Figure 3: Image formation: The HR image $x$ is warped then resampled to align it with the LR image $y$ using the operator $W_{p_t}$. It is then blurred by the operator $B$ to account for integration over LR pixels and finally downsampled in the spatial and spectral domains by the operator $D$ (the spectral downsampling from RGB to R, G or B is not illustrated here for simplicity).

Rewriting (1) as $y = U_p x + \epsilon$, where

$$U_p = \begin{bmatrix} DBW_{p_1} \\ \vdots \\ DBW_{p_K} \end{bmatrix}, \quad y = \begin{bmatrix} y_1 \\ \vdots \\ y_K \end{bmatrix}, \quad p = \begin{bmatrix} p_1 \\ \vdots \\ p_K \end{bmatrix}, \quad \epsilon = \begin{bmatrix} \epsilon_1 \\ \vdots \\ \epsilon_K \end{bmatrix}. \quad (2)$$

Before closing this section, let us note that simpler motion models with two (translation) or three (rigid motion) parameters, or (much) more complex piecewise-affine or elastic models could be considered depending on the application. We focus here on the scenario where a user wishes to zoom in on a relatively small crop (say, between 100 × 100 to 800 × 800 pixels) of a multi-megapixel image, and the affine model has proven effective with real handheld cameras in this setting. This implicitly corresponds to a globally piecewise-affine motion model.

3.2 Inverse problem and optimization

Given the image formation model of Eqs. (1)–(2), recovering the HR image $x$ from the $K$ LR frames $y_k$ in the burst can be formulated as finding the values of $x$ and $p$ that minimize

$$\frac{1}{2}||y - U_p x||^2 + \lambda \phi_0(x), \quad (3)$$

where $\phi_0$ is a parameterized regularizer, to be detailed later, and $\lambda$ is a parameter balancing the data-fidelity and regularization terms. Many methods are of course available for minimizing this function. Like others (e.g., [24]), and mainly for simplicity, we choose here a quadratic penalty method [33, Sec. 17.1] often called half-quadratic splitting (or HQS) [14]: the original objective is replaced by

$$E_\mu(x, z, p) = \frac{1}{2}||y - U_p z||^2 + \frac{\mu}{2}||z - x||^2 + \lambda \phi_0(x), \quad (4)$$

where $z$ is an auxiliary variable, and $\mu$ is a parameter increasing at each iteration, such that, as $\mu \to +\infty$, the minimization of (4) with respect to $x$, $z$ and $p$ becomes equivalent to that of (3) with respect to $x$ and $p$ alone. Each iteration of HQS can be viewed as one step of a block-coordinate descent procedure for minimizing $E$, changing one of variables $z$, $x$ and $p$ at a time while keeping the others fixed, with the value of $\mu$ increasing after each iteration. Convergence guarantees for quadratic penalty methods require an approximate minimization of Eq. (4) with increasing precision over time [33]. Following common practice in computer vision (e.g. [24]), we use HQS without formally checking that its precision indeed increases with iterations. This very simple procedure turns out to work well in practice. Its steps are detailed in the next three paragraphs, the exponent $t$ being used to designate the value of the variables at iteration $t$. The sequence of weights $(\mu^t)_{t \geq 0}$ is learned end-to-end as explained in Section 3.3.

Updating $z$. Several strategies are possible for minimizing Eq. (4) with respect to $z$. Given the dimension of the problem, one may choose for instance a fast iterative minimization procedure such as conjugate gradient descent. Since an approximate minimization is sufficient for our needs, we have chosen to use instead a single step of plain gradient descent, which converges more slowly in theory, but is also simpler and more easily amenable to the unrolled optimization strategy for end-to-end learning that will be presented next. The update at iteration $t$ is given by

$$z^t \leftarrow z^{t-1} - \eta_t \left[ U_p^\top (U_p z^{t-1} - y) + \mu (z^{t-1} - x^{t-1}) \right], \quad (5)$$

where $\eta_t > 0$ is some step size, also learned end to end.

Updating the motion parameters $p$. Let $p_k$ denote the part of the parameter vector $p$ responsible for the alignment of $z^t$ and $y_k$ in (4). The corresponding optimization problem can be rewritten as

$$\min_{p_k} \frac{1}{2}||y_k - DBW_{p_k} z^t||^2. \quad (6)$$

This is a non linear least-squares problem, which can once again be solved using many different techniques. Here, we pick a Gauss-Newton approach, which corresponds to a variant of the Lucas-Kanade algorithm [1, 29], showing again that a 40-year old technique can still be relevant today. Specifically, we perform one Gauss-Newton step at each iteration $t$ for each $p_k$ in parallel:

$$p_k^{t+1} \leftarrow p_k^{t-1} - (J_k^\top J_k)^{-1} J_k^\top r_k^t, \quad (7)$$

where $r_k^t = U_p^{t-1} z^t - y_k$ is the residual of the non-linear least-squares problem (6), and $J_k^t = (\partial U_p^{t-1} / \partial p_k) z^t$ is the Jacobian of the $DBW_{p_k}$ operator. The only difference with a Lucas-Kanade iteration is the presence of a high-resolution frame $z^t$ and the downsampling operator $DB$. This is similar to [16], or more recently [2, 17], which align high-resolution images with low-resolution ones.
Estimating the HR image $x$. The $x$ update is obtained as

$$x^t \leftarrow \arg\min_x \frac{\mu_t}{2} \|z^t - x\|^2 + \lambda \phi_\theta(x),$$

which amounts to computing the proximal operator of the prior $\phi_\theta$. In practice, we follow a “plug-and-play” approach [6, 35, 42], and replace the proximal operator by a parametric function $f_\theta(z_i)$ (here, a CNN, see implementation details). Using such an implicit prior has proven very effective in our setting. More traditional image priors such as total variation could of course have been used as well.

3.3. Unrolled optimization and backpropagation

The optimization procedure described so far requires choosing hyper-parameters such as the sequence $(\mu_t)_{t \geq 0}$, and its implicit prior also involves model parameters $\theta$. By using a training set of $n$ LR burst/HR image pairs, we propose to learn all these parameters in a supervised fashion. We denote the training set by $(Y_i, x_i)^n_{i=1}$, where $Y_i = \{ y_{ij} \}_{j=1}^K$ is the $i$-th burst of LR images associated to the HR image $x_i$. We then unroll the optimization procedure for $T$ steps and, denoting by $\hat{x}_T(Y_i)$ the HR image estimated from burst $Y_i$, we consider the objective function

$$\frac{1}{n} \sum_{i=1}^n L(\hat{x}_T(Y_i), x_i), \quad (8)$$

where $L$ is the $\ell_2$ or $\ell_1$ loss (in practice we have observed that the $\ell_1$ loss performs slightly better). Because every step of our estimation procedure is differentiable, we minimize (8) by stochastic gradient descent.

Learned data prior. Good image priors are essential for solving ill-posed inverse problems. As noted earlier, instead of using a classical one, such as total variation (TV) or bilateral total variation (BTV) [13], we learn an implicit prior parameterized by a convolutional neural network $f_\theta$ in a data-driven manner. We use the ResUNet architecture introduced in [48] in practice. It involves four scales, each of which has an identity skip connection between downsampling and upsampling operations.

3.4. Implementation details and variants

Downsampling and blurring operators D. B. We have tried different variants of downsampling/blurring strategies such as Gaussian smoothing. In practice, we have observed that simple averaging, which is differentiable and parameter-free, gives good results in all our experiments. As a consequence, we do not assume any knowledge about the blur used to generate data, corresponding to an operator $B$ that only captures blur due to photon integration on the sensor without addressing optical blur. We argue that this limited model is relevant because modern cameras and smartphone are aliased [45], which may explain the generalization to real images, as soon as the scene is static.

Initialization by coarse alignment. To initialize the motion parameters $p$, we cannot minimize (6) as in the previous section, because no good estimate of the HR image is available. Therefore, we align each LR frame to an arbitrary one from the burst (e.g., the first one) by using the Lucas-Kanade forward additive algorithm [1, 36] which is known to be robust to noise. Note that another difficulty lies in the raw format of images. To overcome this issue, we simply convert raw images into grayscale images by using bilinear interpolation. This is of course sub-optimal, but sufficient for obtaining coarse motion parameters.

Initialization via coarse-to-fine strategy. For extreme upsampling factors ($\times 16$), we found a coarse-to-fine initialization strategy to be useful: We initialize the motion parameters $p^0$ and high-resolution image $z^0$ by using the output of the algorithm trained at a lower upsampling factor. For instance, $\times 16$ can be obtained by applying twice a $\times 4$ algorithm, or four times $\times 2$ algorithm.

4. Experiments

Experiments were conducted on synthetic and real raw image bursts. We also provide experiments on RGB bursts in the appendix, allowing easier comparison with earlier approaches that cannot handle raw data.

Training procedure and data. For synthetizing realistic raw bursts from groundtruth RGB images, we follow the approach described in [4], using the author’s publicly available code on the training split of the Zurich raw to RGB dataset [19]. The approach consists of applying the inverse RGB to raw pipeline introduced in [5]. Displacements are randomly generated with Euclidean motions and frames are downsampled with bilinear interpolation in order to simulate LR frames containing aliasing. Synthetic, yet realistic, noise is added to the frames, and color values are discarded according to the Bayer pattern. Then, we train our models for minimizing the loss (8). We perform 100,000 iterations of the ADAM optimizer with a batch size of 10, a burst size of 14 and with a learning rate of $3 \times 10^{-5}$ decaying by a factor 2 after 50,000 iterations. Our approach is implemented in Pytorch and takes approximately 1.5 days to train on an Nvidia Titan RTX GPU. We evaluate our models in all our experiments with a burst size of 14 unless specified.

Extreme $\times 16$ upsampling on RGB images. As a proof-of-concept, we also perform experiments for an unusual $\times 16$ super-resolution task, using the coarse-to-fine strategy of Sec. 3.4. A result is presented in Fig. 2, showing impressive reconstruction and additional ones can be found in the appendix. Even though not realistic, we believe the exper-

4https://github.com/goutamgmb/NTIRE21_BURSTSR.
iment to be of interest, as it demonstrates the effectiveness of our approach in an idealistic, yet extreme, setting.

Figure 4: Visual comparison on synthetic raw image bursts used in [4]. Demosaic+SISR is our single-image baseline based on the ResUNet architecture [48] (see main text). The two right columns are produced by methods dedicated to raw burst processing, respectively [4] and ours.

**Evaluation on synthetic RAW images.** The evaluation protocol of [4] allows us to perform quantitative comparison with their state-of-the-art method for processing raw image bursts. An additional comparison with [45] would have been interesting but this method is part of a commercial product that could not be shared with us.

<table>
<thead>
<tr>
<th>Method</th>
<th>PSNR (db)</th>
<th>Geom (pix)</th>
<th>SSIM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scores on public validation set</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ETH [4]</td>
<td>39.09</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Ours (refine)</td>
<td><strong>41.45</strong></td>
<td>-</td>
<td>0.95</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Scores on our own validation set to conduct the ablation study</th>
<th>PSNR (db)</th>
<th>Geom (pix)</th>
<th>SSIM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bicubic Single Image</td>
<td>33.45</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Multiframe L2 only</td>
<td>34.21</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Multiframe L2 + TV prior</td>
<td>34.48</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Single Image</td>
<td>36.80</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Ours (no refinements)</td>
<td>40.38</td>
<td>0.55</td>
<td>0.958</td>
</tr>
<tr>
<td>Ours (refinements)</td>
<td><strong>41.30</strong></td>
<td>0.32</td>
<td>0.963</td>
</tr>
<tr>
<td>Ours (known motion)</td>
<td>42.41</td>
<td>0.00</td>
<td>0.971</td>
</tr>
</tbody>
</table>

Table 1: Results with synthetic raw image bursts of 14 images generated from the Zurich raw to RGB dataset [19] with synthetic affine motions. Reconstruction error in average PSNR and geometrical registration error in pixels for our models. “known p” is the oracle performance our model could achieve, if motion estimation was perfect.

We provide a quantitative comparison in Table 1 with the model introduced in [4], as well as a single-image upsampling baseline based on the ResUNet architecture [48], which we use as a plug-and-play prior in our model.

To that effect, we first use the validation set of [4] available online (with no overlap with the training set), for which motions are unknown, allowing us to compare with their method, which we outperform by more than 2dBs. In order to perform further comparison and conduct the ablation study, we also build an additional validation set by randomly extracting 266 images from the Zurich raw to RGB dataset, allowing us to generate validation data with known motion. We evaluate variations of our model in the same table, notably comparing the registration accuracy achieved by these variants by using the geometrical error presented in [36]. More precisely, we perform a small ablation study by introducing a simpler baseline that does not perform joint alignment and only exploits the coarse registration module (no refine baseline). Performing motion refinement significantly improves the registration accuracy and subsequently the image reconstruction quality. Last, we also report the oracle performance of our model with known motions.

We provide a visual comparison in Figure 4 with single-image SR baselines and the state-of-the-art method [4] for processing raw image bursts. Only the two approaches processing bursts are able to recover high-frequency details, demonstrating their ability to leverage and remove aliasing artefacts, which are very present in the top image. Significantly better quality results are obtained with our approach.

**Impact of burst length and cropping size.** The dataset Zurich rgb-to-raw [19] was very useful for training our models, but it unfortunately features relatively small image crops of size $96 \times 96$ without giving access to the original megapixel images. By experimenting with real raw data, it became apparent to us to our method was performing better with larger crops (e.g., more than $200 \times 200$ pixels), achieving better registration and visually better results. To study the impact of the crop size and burst length, we have thus synthesized additional raw bursts from the DIV2K dataset, and report our experimental results in Figure 5, confirming our findings. Note that this does not appear to be a strong limitation of our approach, since in real-life scenarios, we can always assume that the original megapixel image is available. As expected, the performance of our approach is also increasing with the burst size, even though our models were trained with bursts of size 14.

**Results on real raw image bursts, dataset of [4].** In Figure 6, we show a comparison with [4] using their dataset featuring small crops of size $96 \times 96$. As discussed previously, this setup is suboptimal for our approach, but still produces visually pleasant results. Choosing which method performs best here is however very subjective and we found conclusions hard to draw on this dataset. Whereas the im-
Finally, we demonstrate the effectiveness of our approach on real raw bursts acquired by different devices. We consider a Panasonic Lumix GX9 camera, which is interesting for SR as it does not feature an optical anti-aliasing filter, a Canon Powershot G7X camera, a Samsung S7 and a Pixel 4a smartphones. Results obtained in high noise regimes have already been presented in Figure 1, showing that our approach is surprisingly robust to noise. We believe that the result is of interest since it may allow photographers to use high ISO settings in low-light conditions, without sacrificing image quality. Other results are presented in Figure 6 on low-noise outdoor conditions with bursts of 20 to 30 raw images. In all cases, the method succeeds at recovering high-frequency details. Many more examples and comparisons with other multiframe methods are provided in the supplementary material. We also present failure cases, corresponding in large parts to scene motion. Last, we remark that our method is relatively fast at inference time. Processing a burst of 20 raw 300 × 300 images takes for instance about 1 s on an Nvidia Titan RTX GPU, producing an upsampled image of size 1200 × 1200.

5. Conclusion

We have presented a simple but effective method for super-resolution that combines the interpretability of model-based approaches to inverse problems with the flexibility of data-driven architectures and can be learned from pairs of synthetic LR and real HR images. We plan several extensions, including using multiple cameras to add real LR-burst/HR-image pairs to the training mix, and at test time to take advantage of the multiplicity of imaging devices now available on high-end smartphones. This will open the door to wide-baseline super-resolution applications, such as the construction of high quality panoramas and finely detailed texture maps in multi-view stereo reconstructions. Finally, we plan to explore several other extensions of our approach, including tackling blurry bursts, extending super-resolution to reconstruct HDR images, and pursuing applications in the astronomy and microscopy domains.

Acknowledgments

We thank Frédéric Guichard for useful discussions and comments. This work was funded in part by the French government under management of Agence Nationale de la Recherche as part of the “Investissements d’avenir” program, reference ANR-19-P3IA-0001 (PRAIRIE 3IA Institute). JM and BL were supported by the ERC grant number 714381 (SOLARIS project) and by ANR 3IA MIAI@Grenoble Alpes (ANR-19-P3IA-0003). JP was supported in part by the Louis Vuitton/ENS chair in artificial intelligence and the Inria/NYU collaboration. This work was granted access to the HPC resources of IDRIS under the allocation 2020-AD011011252 made by GENCI.
Figure 7: Results from real raw image bursts obtained with various cameras. We provide comparisons with single image and multiframe baselines. Finest restored details can be seen by zooming on a computer screen. The last three digits of the phone number, only legible in our reconstruction, are masked in the figure for privacy concerns.
References


[29] Bruce D Lucas and Takeo Kanade. An iterative image registration technique with an application to stereo vision. In


**Supplementary material**

This supplementary material presents additional qualitative and quantitative results. In Figure A1 we present additional visual comparison with two burst denoising methods on real images. In Table A1 we present additional experiments on RGB images. Figures A2 and A3 are devoted to super-resolution experiments from real raw data from different smartphones (Google Pixel 3a and 4a, Samsung S7 and S10) and cameras (Panasonic Lumix GX9 and Canon Powershot G7X) and comparison with additional baselines. In Figures A4, we present extreme upsampling results by using synthetic RGB image bursts. In Figure A5, we present restoration results obtained from real images with very low SNR to illustrate the efficiency of our method to perform blind denoising. In Figures A6 and A7, we study the effect of the number of frames in the burst on the reconstruction, both in the low SNR and high SNR settings. Finally, we present failure cases in Figure A8, where fast moving objects are present in the scene.

**Comparison with burst denoising methods.**  We perform additional qualitative comparison on a real image with two burst denoising methods. We compare our method with [10] which performs joint denoising and demosaicking on a burst of raw images. We use the code and the pretrained model made available online. We also use the code and pretrained model of [46]. However the model is only designed to perform grayscale burst denoising, so we perform denoising independently on each RGB channel and then perform demosaicking to get an RGB image. Despite our best efforts for tuning the parameters of these methods to maximize visual quality, the results obtained are not as good as our method (see Figure A1 below). We believe this is not surprising since each one of these methods only addresses a subset of our problem. Adapting them successfully to our general setting is not trivial.

![Comparison with joint denoising and demosaicking methods.](image)

Figure A1: Comparison with joint denoising and demosaicking methods.

**Evaluation on RGB images.**  We compare our approach on the BSD68 dataset against state-of-the-art single-image and video super-resolution algorithms (considering a burst as a video sequence) and report the HR image reconstruction accuracy in terms of average PSNR in Table A1. For the training with RGB data, we perform 80,000 iterations of the ADAM optimizer with a batch size of 10, a burst size of 14 and with a learning rate of $3 \times 10^{-5}$ decaying by a factor 2 after 40,000 iterations. For evaluating the model VSR-DUF [21], we use the code and the pretrained models made available online by the authors. Other single-image reconstruction results are from [48]. In the present setting, we consistently outperform other baselines, notably demonstrating that burst SR cannot simply be addressed effectively by current video SR approaches. We also note that our models perform better with less blurring (and more aliasing). Finally, we evaluate variations of our model in the same table, notably comparing the registration accuracy achieved by these variants by using the geometrical error presented in [36]. More precisely, we perform a small ablation study by introducing a simpler baseline that does not perform joint alignment and only exploits the coarse registration module (no refine baseline). Performing joint alignment and image estimation systematically improves motion estimation. Last, we also report the oracle performance of our model with known motions.

<table>
<thead>
<tr>
<th>Method</th>
<th>Scaling factor / blurring kernel std</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$\times 2/\sigma = 0.7$</td>
</tr>
<tr>
<td><strong>Single Image SR</strong></td>
<td></td>
</tr>
<tr>
<td>RCAN [50]</td>
<td>29.48</td>
</tr>
<tr>
<td>IRCNN [49]</td>
<td>29.60</td>
</tr>
<tr>
<td>USRNet [48]</td>
<td>30.55</td>
</tr>
<tr>
<td><strong>Video SR</strong></td>
<td></td>
</tr>
<tr>
<td>Ours (no refine)</td>
<td><strong>42.36/0.10</strong></td>
</tr>
<tr>
<td>Ours</td>
<td><strong>43.73/0.07</strong></td>
</tr>
<tr>
<td>Ours (known $p$)</td>
<td>45.72/0.00</td>
</tr>
</tbody>
</table>

Table A1: **Results for RGB with synthetic affine motions.** of different methods for different combinations of scale factors and blur kernels. Results are given in term of average PSNR in dBs and geometrical registration error in pixels for our models. “known $p$” is the oracle performance our model could achieve, if motion estimation was perfect.
Figure A2: Results from real raw image bursts obtained with various cameras. We provide comparisons with single image and multiframe baselines. Finest restored details can be seen by zooming on computer screen.
Figure A3: Results from real raw image bursts obtained with various cameras. We provide comparisons with single image and multiframe baselines. Finest restored details can be seen by zooming on computer screen.
Figure A4: Extreme $\times 16$ upsampling experiment. The right image is obtained by processing a burst of 20 LR images presented on the left obtained with synthetic random affine movements and average pooling downsampling.
Figure A5: Image restoration of images taken at night with very low signal to noise ratio by using a Panasonic GX9 camera.
Figure A6: Visual differences caused by merging a different number of frames in the case of low SNR scenes. With a larger number of frames we can observe a quality increase and better denoising.

Figure A7: Visual differences caused by merging a different number of frames in the case of high SNR scenes. With a larger number of frames we can observe a quality increase.

Figure A8: Misalignments artefacts due to moving objects in the scene. Our current implementation does not handle fast moving objects and then generates visual artefacts. Dealing with fast dynamic scenes will be the focus of future work.