



Published in Image Processing On Line on 2017-03-19.
 Submitted on 2016-12-09, accepted on 2017-02-21.
 ISSN 2105-1232 © 2017 IPOL & the authors CC-BY-NC-SA
 This article is available online with supplementary materials,
 software, datasets and online demo at
<https://doi.org/10.5201/ipol.2017.197>

Implementation of Local Fourier Burst Accumulation for Video Deblurring

Jérémy Anger, Enric Meinhardt-Llopis

CMLA, ENS Cachan, France

(`{jeremy.anger, enric.meinhardt}@cmla.ens-cachan.fr`)

Communicated by Jérôme Gilles

Demo edited by Jérémy Anger

Abstract

This article presents Local Fourier Burst Accumulation, a recent motion deblurring method. This method processes image bursts and can be naturally extended to video deblurring. The algorithm first registers the frame to its neighboring frames by consistency-controlled optical flow and then fuses the frames temporally by a weighted average in the Fourier domain. As the method does not require the estimation of the blur kernel, it is less sensitive to common deblurring issues such as ringing. The algorithm is detailed and comparison with the results of the original authors are performed.

Source Code

The reviewed source code for the algorithms described in this document are available from [the web page of the article](#)¹. Compilation and usage instructions are included in the `README.txt` file.

Keywords: motion deblurring; camera shake; Fourier accumulation; video restoration

1 Introduction

Image deblurring is a hard problem that has been studied for decades. This ill-posed problem can be expressed by Equation (1), where v is the observed image, k is the blur kernel, n is an additive white noise, and $*$ denotes the convolution,

$$v = k * u + n. \quad (1)$$

The goal of image deblurring is to recover the latent image u from v , knowing neither k nor n . Note that a shift invariant blurring kernel is assumed, which is a rough approximation but is often used

¹<https://doi.org/10.5201/ipol.2017.197>

to simplify the problem. Video deblurring is modeled as a sequence of blurred images with a time varying kernel as described by

$$v_i = k_i * u_i + n_i. \quad (2)$$

Existing single image deconvolution methods such as the one proposed by Shan et al. [8] achieve excellent image quality by estimating both the blur kernel and the latent image. Although very attractive, these methods are not designed to process videos, mainly due to their high computational cost and to the requirement that the deblurring result must be coherent across frames to avoid jitter.

Video deblurring differs from image deblurring by the fact that temporal information can be used to achieve good results without requiring complex optimization processes. Indeed, instead of considering each image of the video as independent, it is possible to exploit the temporal redundancy. Considering a static camera – or assuming pre-registered frames – one can write

$$v_i = k_i * u + n_i, \quad (3)$$

which is a better posed model than Equation (2) applied to each single frame.

Recently, Cho et al. [1] proposed a patch-based deblurring method which fuses similar 2D patches in a video using a weighted average in the spatial domain, thus transferring sharp regions to neighboring frames. Inspired by their work, Delbracio et al. [3, 2] proposed the Fourier Burst Accumulation method (FBA) in which the frames are averaged in the frequency domain. This fusion method is particularly effective when the blur is temporally variant, for example in videos captured with hand-held cameras. In this method, the frames are first registered using SIFT features and the ORSA algorithm by Moisan et al. [6] to compute a homography and are then processed by FBA globally. Delbracio et al. [2] improved FBA and designed the Local Fourier Burst Accumulation method by using an optical flow algorithm to consistently register the frames and fused them locally on overlapping tiles. This paper describes this local method which is less sensitive to moving objects and does not presuppose a homography for the camera motion.

Based on the work of Delbracio et al. [3], Wieschollek et al. [9] designed a neural network which includes a modified Fourier Burst Accumulation step. In their work, the weights of the accumulation are learnt by the network. Gilles et al. [5] replaced the Fourier space by wavelet spaces and applied the resulting methods to turbulence mitigation.

2 Local Fourier Burst Accumulation

The Fourier Burst Accumulation (FBA) can be viewed as a lucky image algorithm working in the frequency domain instead of the spatial one. In classical lucky image methods, the burst should contain sharp images or at least sharp regions. If all images have blur, the result will be poor. Instead, FBA fuses the burst's images by considering their frequency coefficients. As explained by Delbracio et al. [2], the motion blur process attenuates strongly some Fourier coefficients while leaving others almost unaltered. FBA seeks to aggregate the Fourier coefficients depending on their magnitude. Indeed, if a given Fourier coefficient in a frame is larger than its corresponding coefficients in the nearby frames, this fact indicates that this Fourier coefficient has not been attenuated by some motion blur. Motion blur indeed is a low pass filter, therefore all of its Fourier coefficients have modulus smaller than 1. Thus, a high magnitude indicates a low degradation. The method is effective even if each image is blurred, as long as this blur is sufficiently time-variant to ensure that each Fourier coefficient has been left unaltered at least once in some neighboring frames.

Algorithm 1 describes the entry point of the method. It processes a sequence of images using a temporal sliding window of length $2M + 1$. For each image of the sequence, the algorithm extracts this temporal window, registers the frames and applies the Fourier Burst Accumulation.

Algorithm 1: Entry Point of Local Fourier Burst Accumulation

input : $\{I_n\}$ sequence of N images, M half-size of the temporal sliding window ($= 3$)
output: deblurred sequence $\{I_n^*\}$
for i **from** 0 **to** $N - 1$ **do**
 $\{W_m\} = \{I_n, n \in \llbracket i - M, i + M \rrbracket\}$ *Extract the temporal window around I_i*
 $\{W_m^*\} = \text{consistent_registration}(\{W_m\})$ *Register the frames (Algorithm 2)*
 $I_i^* = \text{fourier_burst_accumulation}(\{W_m^*\})$ *Process the temporal window (Algorithm 3)*

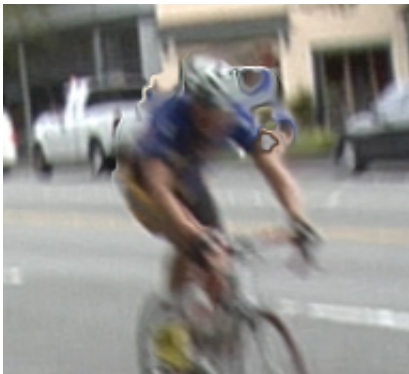
2.1 Consistent Registration

In order to reach the ideal setting of Equation (3), the FBA method proposes to locally register the frames using a displacement field such as $u = u_i \circ \phi_i$, so that the registered images write

$$v_i = k_i * (u_i \circ \phi_i) + n_i. \quad (4)$$

The registration method proposed by the authors is presented in Algorithm 2. Each of the $2M + 1$ frames are registered to the reference, which is the image at the center of the sliding window (at index M).

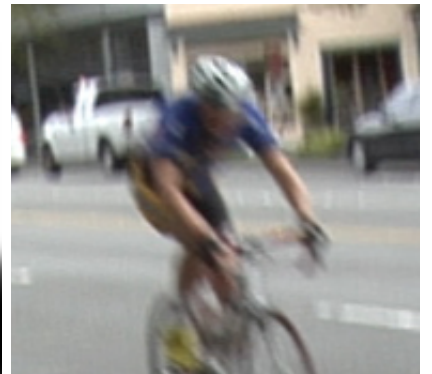
As the optical flow may be poorly defined in some areas, due to occlusion for example, a consistency map is computed using the forward and backward flow. For each pixel, if the flows introduce a displacement error, a low confidence will be associated with the pixel. The registered image is defined as the warped frame if the confidence is high and as the reference frame otherwise. This ensures that no objects are present at two different places in the same temporal window, thus avoiding ghosting effects. In order to smoothly fuse both images, an erosion and smoothing of the consistency map is applied. Figure 1 shows a result of the consistency registration around a moving object.



(a) Inconsistent registration



(b) Consistency map



(c) Correction of the registration

Figure 1: Illustration of the consistent registration. The consistency map shows in black the parts that should be replaced by the reference. This eliminates in (c) the phantom boundaries around the cyclist appearing in (a).

Downsampling the input images before computing the optical flows reduces the influence of the motion blur in the registration process while speeding it up. In fact, applying an accurate full-resolution optical flow between two randomly motion blurred images would likely result in the compensation of the blur, thus resulting in two similarly blurred images and making the FBA method less effective. Figure 2 shows the effects of the downsampling. One can see that using a fully resolved image produces artifacts and modifies the blur orientation. Following the original paper by Delbracio et al. [2], the TV-L1 optical flow method by Zach et al. [10] was computed on the images downsampled by a factor $\tau = 1/3$ and then upsampled back to the original image resolution. We use here the

implementation of the multiscale TV-L1 optical flow by Perez et al. [7]. The tuning of this parameter depends on the blur kernel size, which could be estimated from accelerometers.

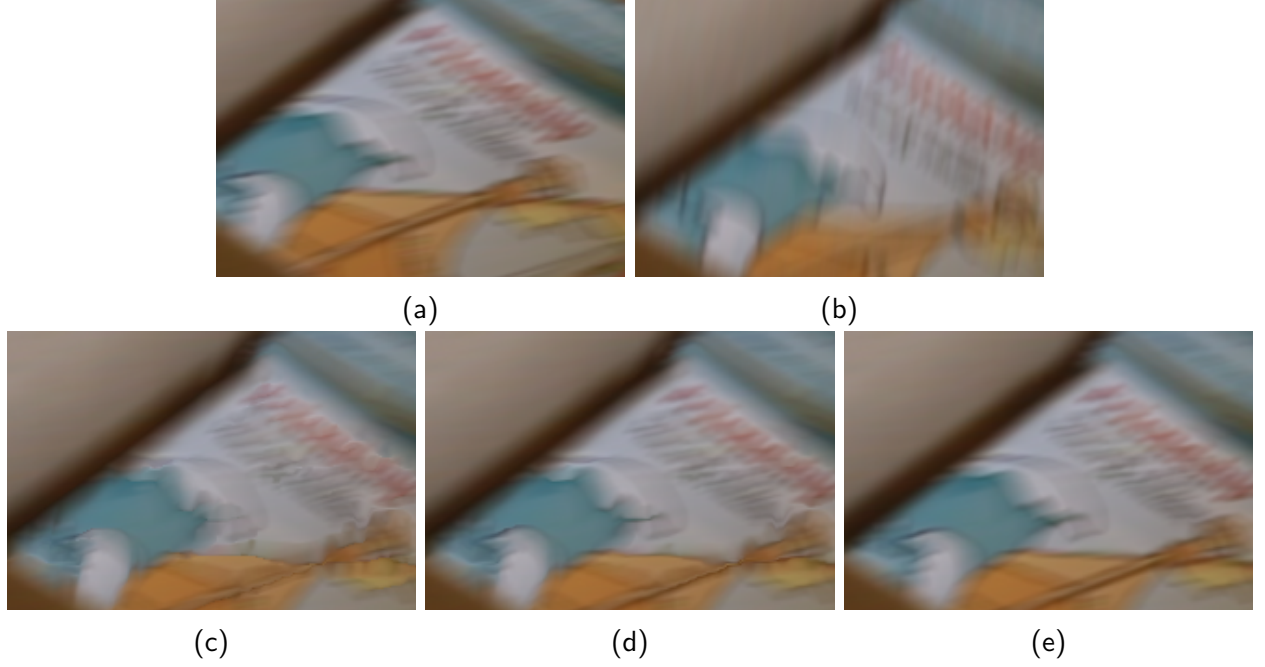


Figure 2: Influence of the downsampling during registration. Here, (a) and (b) are two consecutive artificially blurred images. Images (c), (d) and (e) are the results of the registration of (a) to (b) with the optical flow computed at scales 1, 0.5 and 0.25 respectively. In (c) and (d) the downsampling is not sufficient because the motion blur is transferred during registration. In (e) the motion blur is not altered, hence the registration is adapted to FBA.

Algorithm 2: Consistent Registration

input : $\{I_m\}$ sequence of $2M + 1$ images, σ_m blur radius ($= 5$), r erosion radius ($= 5$),
 ϵ displacement tolerance ($= 1$), τ downsampling factor ($= 1/3$)

output: registered sequence $\{I_m^*\}$

$\{D_m\} = \text{convert_grey}(\text{downsample}_\tau(\{I_m\}))$ *Downsample and convert to grey*

for m **from** 0 **to** $2M$ **do**

$F_{m \rightarrow M} = \text{upsample}_\tau(\text{optical_flow}(D_m, D_M))$

$F_{M \rightarrow m} = \text{upsample}_\tau(\text{optical_flow}(D_M, D_m))$

$\text{displacement_error}(x) = |F_{M \rightarrow m}(x) + F_{m \rightarrow M}(x + F_{M \rightarrow m}(x))|_2$

$\text{cMap} = \text{blur}_{\sigma_m}(\text{erode}_r(\text{displacement_error} < \epsilon))$

$I_m^*(x) = I_m(x + F_{m \rightarrow M}(x))$

$I_m^* = I_m^* \cdot \text{cMap} + I_M \cdot (1 - \text{cMap})$

Forward flow [7]

Backward flow [7]

Consistency map

Register the frame

Copy the reference over inconsistent areas

2.2 Aggregation

Having registered the frames, Equation (3) applies. Thus, temporal redundancy can be exploited by aggregating the frames together following Equation (5) and as described in Algorithm 3

$$\hat{I}^* = \frac{\sum_m \widehat{I_m} \cdot (G_\sigma |\widehat{I_m}|)^p}{\sum_m (G_\sigma |\widehat{I_m}|)^p}. \quad (5)$$

Here, \widehat{I} denotes the Fourier representation of I and G_σ is a Gaussian filter of standard deviation σ which makes the aggregation less sensitive to noise.

Instead of applying Equation (5) on the entire frame, the image space is split into overlapping patches of size $W \times W$. The overlap ratio can be set to 0.5, in which case each pixel of the final image is obtained by averaging four processed pixels.

For each patch position, the algorithm considers a temporal window containing the patches at this position. As explained by Delbracio et al. [2], by averaging the frequencies using their magnitude as weights, it is possible to reduce the motion blur of a burst.

One of the effects of processing the burst by patches is to alleviate the issue of a spatially-variant blur kernel. Working by patches also reduces the propagation of potential frequency artifacts such as ringing.

The parameter p tunes the contributions of the Fourier coefficients. With $p = 0$, each frame has the same contribution weight and the result coincides with an arithmetic average, which has a good denoising behavior but blurs the image even more. As p is increased, coefficients of high magnitude get a stronger influence and the resulting image becomes sharper. If the noise level is low, setting $p = 11$ provides a good compromise between denoising and deblurring.

Algorithm 3: Local Fourier Burst Accumulation

input : $\{I_m^*\}$ sequence of M registered images of size $w \times h$, W spatial window size ($= 128$),
 p ($= 11$), σ_w blur radius ($= \frac{50}{W}$), ϵ ($= 10^{-8}$)
output: deblurred image I^*
 $S = \{i, i \text{ from } 0 \text{ to } w - \frac{W}{2} \text{ with step } \frac{W}{2}\} \times \{j, j \text{ from } 0 \text{ to } h - \frac{W}{2} \text{ with step } \frac{W}{2}\}$ *Generate the patch set*

foreach $(i, j) \in S$ **do**
 for m **from** 0 **to** $2M$ **do**
 $P_m = I_m^*((i, j) \text{ to } (i + W, j + W))$ *Extract the patch using symmetric boundaries*
 $\widehat{P}_m = \text{FFT}(P_m)$ *Per-channel Fourier transform*
 $w_m = |\text{blur}_{\sigma_w}(\text{convert_grey}(|\widehat{P}_m|))|^p$ *Weight map of the patch*
 $\widehat{P}^* = \frac{\sum_m \widehat{P}_m \cdot w_m}{\sum_m w_m + \epsilon}$ *Weighted average of the patches' spectrum*
 $P_{i,j} = \text{IFFT}(\widehat{P}^*)$ *Inverse transform*

$S_{x,y} = \{(i, j), \forall (i, j) \in S \mid x \in [i, i + W[\text{ and } y \in [j, j + W[\}$ *Set of patches at a given pixel*
 $I^*(x, y) = \frac{1}{\text{card}(S_{x,y})} \sum_{(i,j) \in S_{x,y}} P_{i,j}(x - i, y - j)$ *Average the patches*

To ensure the reproducibility of our work, Table 1 supplements the implementation details of the algorithms.

2.3 Additional Processes

In addition to the registration and aggregation, Delbracio et al. [2] propose to apply an unsharp masking filter to increase the sharpness of the results at a low cost. This helps remove potential remaining isotropic blur. This step is not included in the provided source code because it is not specific to motion deblurring.

Furthermore, Delbracio et al. [2] propose an iterative scheme. By applying FBA iteratively, sharp frames propagate further than the chosen temporal window size. The authors suggest to compute between 1 and 4 iterations. This scheme is included in our published source code, but for computational time constraints only one iteration is done on the online demo.

Subroutine	Parameters and remarks
Optical flow	$\tau = 0.25, \lambda = 0.25, \theta = 0.3, N_{scales} = 5, \eta = 0.5, N_{warps} = 3$ $\epsilon = 0.001, N_{max.iter} = 10$ (see Perez et al. [7] for notations). Image warping uses bicubic interpolation.
Upsampling	Upscale the image with bicubic interpolation.
Downsampling	Downscale the image by an averaging filter.
Erosion	Disk as structuring element.
Fourier coefficients blurring	We used Vliet-Young-Verbeek Gaussian convolution [4].
Patch extraction	For patches overlapping with the image borders, symmetric padding is added.

Table 1: Additional implementation details.

3 Results

In this section, we analyze some results obtained with our implementation. The presented results can be reproduced on [the online demo](#)², where real and simulated sequences are available as inputs.

3.1 Real Sequences

To study the behavior of the presented implementation, we used the video sequences provided at Delbracio’s [website](#)³. This website also contains the original implementation results shown in Figure 3(b) and 4(b). Unless explicitly specified, the default parameters were $M = 3$ (7 frames to process a single image), $W = 128$, $p = 11$, a downsampling factor of 1/3 and one iteration.



Figure 3: Comparison between the original implementation and ours. Note that Delbracio et al. use an unsharp masking step as postprocessing.

Figure 3 shows a sample of a sequence processed by our implementation and compared with the results of Delbracio et al. [2]. As expected, the results are very similar. Delbracio et al.’s implementation includes a sharpening step which we haven’t included because it is outside of the scope of motion deblurring.

Figure 4 highlights a slight difference between the consistent registration implementations. In this example, we obtain less artifacts around the moving object. This variation is due to different parameters used in the optical flow computation, such as the number of iterations.

²<https://doi.org/10.5201/ipol.2017.197>

³<http://dev.ipol.im/~mdelbra/videoFA/>

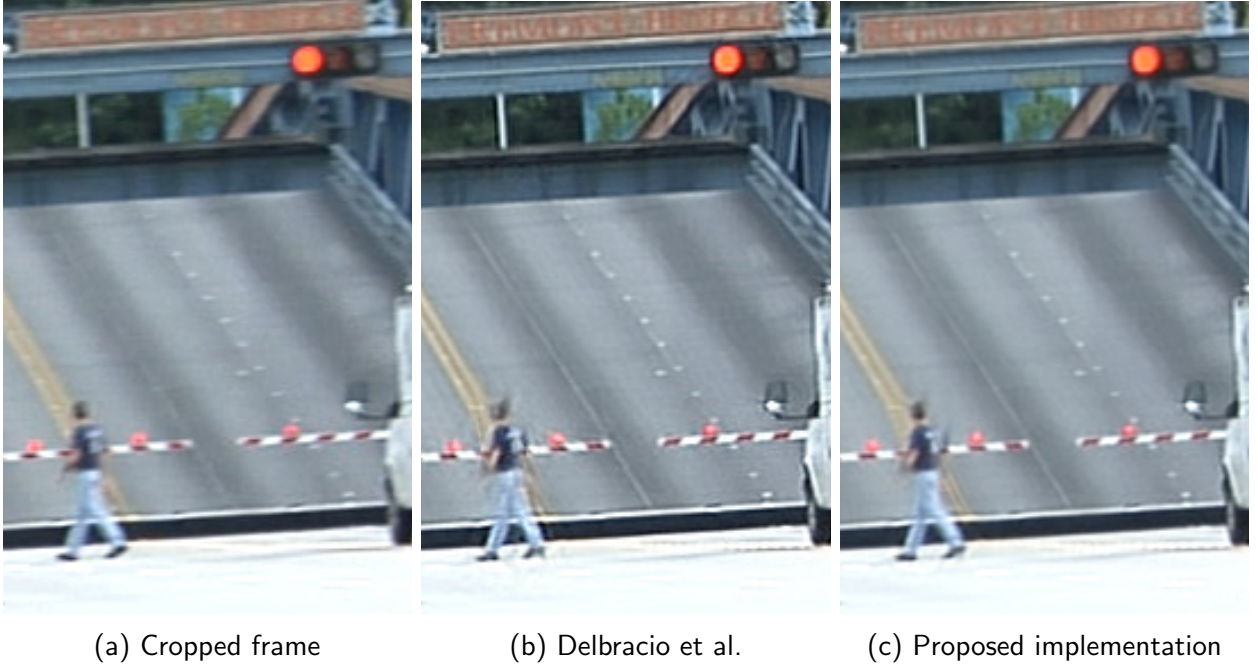


Figure 4: Comparison between the original implementation and ours.

3.2 Simulated Sequences

In order to study the influence of the motion blur, we simulated a few sequences by extracting a sharp frame from the real sequences and applying different artificial blur kernels obtained by randomly rotating a line segment. These kernels approximate a simple translation of the camera during exposure time. The result is a stationary sequence with time-variant blur. It can be processed directly by the Local Fourier Burst Accumulation algorithm without registration.

An example of result with registration is shown in Figure 5(c) and without registration in Figure 5(b). This experiment clearly demonstrates that the registration by optical flow tends to compensate the blur and risks degrading the performances of FBA.



Figure 5: Result on a simulated sequence. If the frames are stationary, the registration degrades the output.

4 Conclusion

We have presented an implementation of the Local Fourier Burst Accumulation method. We illustrated our implementation with various results, showing that the results of the original authors were successfully reproduced.

This method proposes to deblur a burst of images or a video using a computationally efficient weighted average in the Fourier domain. Before accumulating, the frames are registered in a consistent manner using a dense optical flow. The fusion itself consists simply of an accumulation of the Fourier coefficients using a higher weight for frequencies with high magnitude.

The FBA method uses relatively few parameters and provides high quality results especially when the blur kernels are time independent. Its consistent registration scheme makes the method robust to moving objects.

Acknowledgments

Work partly founded by the Centre National d'Etudes Spatiales (CNES, MISS Project), BPIFrance and Région Ile de France in the FUI 18 Plein Phare project, the Office of Naval research (ONR grant N00014-14-1-0023), ANR-DGA project ANR-12-ASTR-0035, and a 2016 DGA PhD scholarship, jointly supported with Fondation Mathématique Jacques Hadamard.

Image Credits



Dataset of Cho et al. [1]

References

- [1] SUNGHYUN CHO, JUE WANG, AND SEUNGYONG LEE, *Video deblurring for hand-held cameras using patch-based synthesis*, ACM Transactions on Graphics, 31 (2012), pp. 64:1–64:9. <https://dx.doi.org/10.1145/2185520.2185560>.
- [2] MAURICIO DELBRACIO AND GUILLERMO SAPIRO, *Hand-held video deblurring via efficient Fourier aggregation*, IEEE Transactions on Computational Imaging, 1 (2015), pp. 270–283. <https://dx.doi.org/10.1109/TCI.2015.2501245>.
- [3] —, *Removing camera shake via weighted Fourier burst accumulation*, IEEE Transactions on Image Processing, 24 (2015), pp. 3293–3307. <https://dx.doi.org/10.1109/TIP.2015.2442914>.
- [4] PASCAL GETREUER, *A Survey of Gaussian Convolution Algorithms*, Image Processing On Line, 3 (2013), pp. 286–310. <https://doi.org/10.5201/ipol.2013.87>.
- [5] JÉRÔME GILLES AND STANLEY OSHER, *Wavelet burst accumulation for turbulence mitigation*, Journal of Electronic Imaging, 25 (2016), p. 033003. <https://dx.doi.org/10.1117/1.JEI.25.3.033003>.
- [6] LIONEL MOISAN, PIERRE MOULON, AND PASCAL MONASSE, *Automatic homographic registration of a pair of images, with a contrario elimination of outliers*, Image Processing On Line, 2 (2012), pp. 56–73. <https://doi.org/10.5201/ipol.2012.mmm-oh>.
- [7] JAVIER SÁNCHEZ PÉREZ, ENRIC MEINHARDT-LLOPIS, AND GABRIELE FACCIOLO, *TV-L1 optical flow estimation*, Image Processing On Line, 3 (2013), pp. 137–150. <https://doi.org/10.5201/ipol.2013.26>.

- [8] QI SHAN, JIAYA JIA, AND ASEEM AGARWALA, *High-quality motion deblurring from a single image*, in ACM Transactions on Graphics (TOG), vol. 27, ACM, 2008, p. 73. <https://doi.org/10.1145/1399504.1360672>.
- [9] PATRICK WIESCHOLLEK, BERNHARD SCHÖLKOPF, HENDRIK LENSCH, AND MICHAEL HIRSCH, *End-to-end learning for image burst deblurring*, arXiv preprint, (2016).
- [10] CHRISTOPHER ZACH, THOMAS POCK, AND HORST BISCHOF, *A duality based approach for realtime TV-L1 optical flow*, in Joint Pattern Recognition Symposium, Springer, 2007, pp. 214–223. https://doi.org/10.1007/978-3-540-74936-3_22.