Non-uniform Motion Blur Kernel Estimation via Adaptive Decomposition

Guillermo Carbajal, Universidad de la República
Patricia Vitoria, Universitat Pompeu Fabra
Mauricio Delbracio, Google Research, USA
Pablo Musé, Universidad de la República
José Lezama, Universidad de la República

Analytic and Geometric Approaches to Machine Learning
ICMS Workshop, University of Bath

July 30, 2021

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

Universitat Pompeu Fabra Barcelona

Motivation

- Motion blur estimation and restoration are fundamental problems in image processing and computer vision.
- Motion blur is produced by unwanted camera shake during recording or by fast moving objects in the scene.
- Consequences: image quality degradation
  - Aesthetic blurry images
  - Performance degradation of subsequent computer vision tasks: tracking, detection, classification, etc.

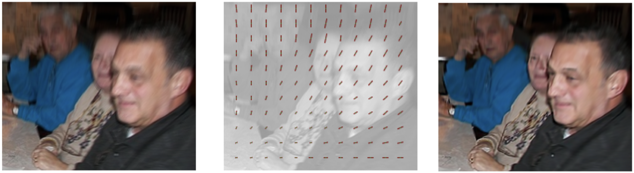
Objectives

We focus on the realistic non-uniform motion blur setting

- Major goal: provide dense, accurate estimates of non-uniform motion fields *via* local kernel estimation. Estimate kernels at the pixel level.
- Secondary goal: once the motion field has been estimated, perform non-blind image deblurring.
- Image deblurring is tackled here to validate our non-uniform kernel estimation method, by comparing to state-of-the-art deblurring techniques.

Related work: non-uniform motion kernel estimation
[Sun 2015]

- Predefined set of linear kernels of different lengths and orientations.
- Deep CNN to predict the probability of each kernel for each patch.
- Smooth dense motion blur kernels field *via* MRF regularization.
- Motion blur removed using a non-uniform blur model with EPLL prior.

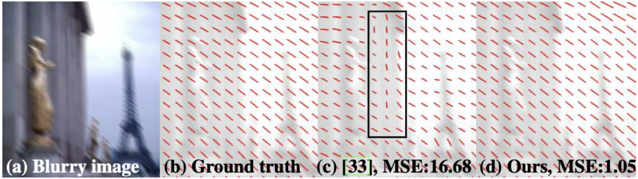


(a) Input image (b) Estimated motion blur field by CNN (c) Result after deblurring

Sun, Cao, Xu, Ponce. *Learning a CNN for non-uniform motion blur removal.* CVPR, 2015.

Related work: non-uniform motion kernel estimation
[Gong 2017]

- Estimate the motion flow from the blurred image using a deep FCNN.
- Train the FCN with simulated linear kernels to generate synthetic blurred-image / motion-flow pairs.
- Deblur with conventional non-blind method (ℓ_2 data fit, EPLL prior).



Limitations of [Sun 2015] and [Gong 2017]: Limited to line-shaped blur kernels, inaccurate and unrealistic in most scenarios.

Gong, Yang, Liu, Zhang, Reid, Shen, Van Den Hengel, Shi. *From motion blur to motion flow: a deep learning solution for removing heterogeneous motion blur.* CVPR, 2017.

Related work: Kernel Prediction Networks

Recently used, among others, for burst denoising, optical flow estimation and frame interpolation, stereo and video prediction.

Burst denoising [Mildenhall 2018]:

- The network produces a set of kernels at each pixel, which are then used to produce a pixel average in the neighborhood.
- Significant memory and computational costs: limited to compute for each pixel $N = 8$ kernels of size $K = 5$, therefore limiting the ability to denoise over frames with larger relative motion.

Mildenhall, Barron, Chen, Sharlet, Ng, Carroll. *Burst denoising with kernel prediction networks.* CVPR, 2018.

Related work: Kernel Prediction Networks

Burst denoising [Xia 2019]:

- To overcome this limitation, they propose a basis prediction network that given an input burst, predicts a set of global basis kernels and the corresponding per-pixel mixing coefficients.
- Kernel size limited to 15×15, and total number of basis kernels limited to 90.

Xia, Perazzi, Gharbi, Sunkavalli, Chakrabarti. *Basis prediction networks for effective burst denoising with large kernels*. ArXiv preprint, 2019.

Proposed approach

We propose to estimate the non-uniform motion blur kernels using a KPN.

For each blurry image we estimate:

- A set of basis motion kernels for the whole image.
- A set of mixing coefficients specific to each pixel.

Our work bears many similarities with [Xia 2019] kernel prediction strategy.

Major differences:

- Application: non-uniform motion blur estimation.
- The kernels are learned directly to solve the inverse problem, whereas we learn the kernels to fit the forward model.

Non-uniform motion blur degradation model

- Sharp image $\mathbf{u} \in \mathbb{R}^{H \times W}$
- For each pixel $i = 1, \ldots, H \times W$, a blur kernel $\mathbf{k}_i \in \mathbb{R}^{K \times K}$
- Kernels $\mathbf{k}_i$ are non-negative and $\|\mathbf{k}_i\|_1 = 1$

The blurry image $\mathbf{v}$ is the result of applying the per-pixel operation:

$$v_i = \langle \mathbf{u}_{nn(i)}, \mathbf{k}_i \rangle + n_i.$$

Taking into account sensor saturation and gamma correction,

$$v_i = R\left(\langle \mathbf{u}_{nn(i)}, \mathbf{k}_i \rangle + n_i\right)^{1/\gamma},$$

where (typical) $\gamma = 2.2$ and $R(x)$ is a smooth approximation of $\min(x, 1)$:

$$R(x) = x - \frac{1}{a} \log\left(1 + e^{a(x-1)}\right).$$

Low-rank approximation

- Full motion field of per-pixel kernels: very high-dimensional space (K^2HW) (hard and computationally intractable)
- Low-rank modeling (assume spatial redundancy): $\mathbf{k}_i \simeq \sum_{b=1}^{B} m_i^b \mathbf{k}^b$.
  - The B basis elements $\mathbf{k}^b$ are *image specific*
  - The mixing coefficients m_i^b are non-negative and $\sum_{b=1}^{B} m_i^b = 1$.
- Estimation problem of dimension $B(K^2 + HW)$.
- Forward model:

$$\boxed{v_i = R\left(\langle \mathbf{u}_{nn(i)}, \sum_{b=1}^{B} m_i^b \mathbf{k}^b \rangle + n_i\right)^{1/\gamma}}$$

Objective loss

- We aim to minimize the following two term loss:

$$\mathcal{L}_{reblur} + \mathcal{L}_{kernel}.$$

- At training time, for the synthetic dataset, we have:
  - The sharp image $\mathbf{u}^{\mathrm{GT}}$
  - The blured image $\mathbf{v}^{\mathrm{GT}}$
  - The ground truth kernels and mixing coefficients that were applied to each pixel, $\mathbf{k}_i^{\mathrm{GT}}$.

Reblur loss

Given a blurry image $\mathbf{v}^{\mathrm{GT}}$, we aim to find the global *kernel basis* $\{\mathbf{k}^b\}$ and *per-pixel mixing coefficients* $\{\mathbf{m}^b\}$ that minimize

$$\mathcal{L}_{reblur} = \sum_i w_i \left(v_i^{\mathrm{GT}} - v_i\right)^2,$$

where

- w_i is a weight inversely proportional to the number of pixels affected with the same ground truth blur kernel $\mathbf{k}_i^{\mathrm{GT}}$

Kernel Loss

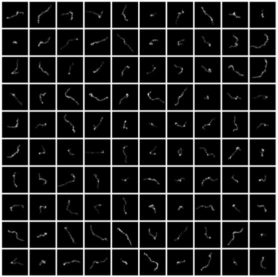
Given the ground truth per-pixel blur kernels $\mathbf{k}_i^{\mathrm{GT}}$, the computed kernel basis $\{\mathbf{k}^b\}$ and mixing coefficients $\{\mathbf{m}^b\}$, the *kernel loss* is defined as:

$$\mathcal{L}_{kernel} = \sum_i w_i \left\| \sum_{b=1}^{B} m_i^b \mathbf{k}^b - \mathbf{k}_i^{\mathrm{GT}} \right\|_p^p .$$

Synthetic training database generation procedure

To build a dataset of tuples $(\mathbf{u}^{GT}, \mathbf{v}^{GT}, \{\mathbf{k}\}^{GT}, \{\mathbf{m}\}^{GT})$, we make use of:

- A function that generates “camera shake" random kernels by using the physiological hand tremor data of [Gavant 2011][Delbracio 2015]
- The ADE20K image dataset: annotated images of segmented scenes.



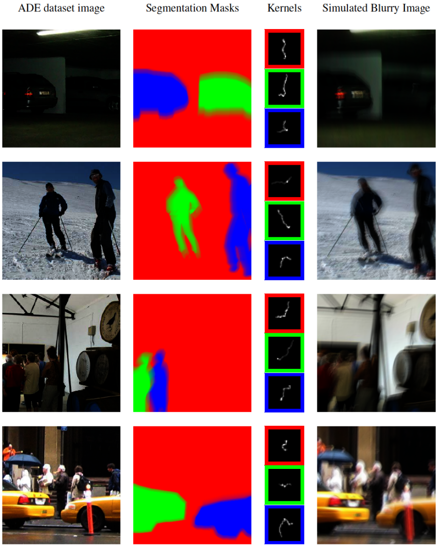

Gavant, Alacoque, Dupret, David. *A physiological camera shake model for image stabilization systems.* SENSORS, IEEE, 2011.
Delbracio, Sapiro. *Removing Camera Shake via Weighted Fourier Burst Accumulation.* TIP, 2015.
Zhou, Zhao, Puig, Fidler, Barriuso, Torralba. *Scene Parsing through ADE20K Dataset.* CVPR, 2017.

Synthetic training database generation procedure

1. Sample an image $\mathbf{u}^{\mathrm{GT}}$ from ADE20K
2. Sample a kernel $\mathbf{k_1}^{\mathrm{GT}}$ generated with [Delbracio 2015]
3. Convolve $\mathbf{u}^{\mathrm{GT}}$ with $\mathbf{k_1}^{\mathrm{GT}}$
4. If the image contains segmented objects,
   4.1 sort a new kernel $\mathbf{k_2}^{\mathrm{GT}}$
   4.2 Convolve the segmented region of $\mathbf{u}$ with $\mathbf{k_2}^{\mathrm{GT}}$
5. Repeat (4) until no more segmented objects are present in $\mathbf{u}^{\mathrm{GT}}$.

Return: sharp and blurred images $\mathbf{u}^{\mathrm{GT}}$, $\mathbf{v}^{\mathrm{GT}}$, kernels $\mathbf{k_n}^{\mathrm{GT}}$ with associated masks.

Synthetic training database generation procedure

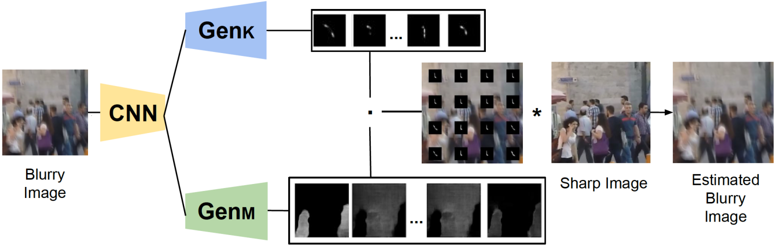


Implementation details

Parameters

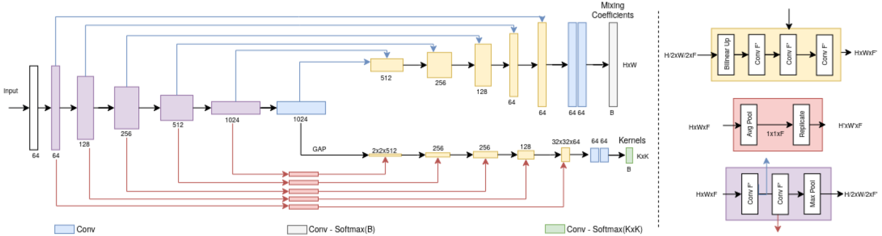
- Number of basis kernels $B = 25$
- Kernel size: $K \times K = 33 \times 33$

Training pipeline



Implementation details

Architecture



Synthetic images datasets

State-of-the-art deblurring networks are mostly trained with datasets that synthesize realistic motion blur averaging several short exposure frames.

- GoPro [Nah 2017]: first one of this kind, widely used both for training and as a benchmark
- DVD [Su 2017]: significantly reduced *ghosting*, but noticeable compression artifacts.

Nah, Kim, Lee. *Deep multi-scale convolutional neural network for dynamic scene deblurring.* CVPR 2017.
Su, Delbracio, Wang, Sapiro, Heidrich, Wang. *Deep video deblurring for hand-held cameras.* CVPR 2017.

Real images datasets

- [Lai 2016]: real (and synthetic) datasets, usually used for evaluation purposes
- [Köhler 2012]: reduced set of example images with (slightly) non-uniform blur originating from real camera trajectories
- Realblur [Rim 2020]: two cameras shoot at the same time. One camera captures a blurred image with a low shutter speed, the other captures a GT image with a high shutter speed.

Lai, Huang, Hu, Ahuja, Yang. *A comparative study for single image blind deblurring.* CVPR, 2016.
Köhler, Hirsch, Mohler, Schölkopf, Harmeling. *Recording and playback of camera shake: Benchmarking blind deconvolution with a real-world database.* ECCV, 2012.
Rim, Lee, Won, Cho. *Real-World Blur Dataset for Learning and Benchmarking Deblurring Algorithms. ECCV, 2020.*

To validate the proposed non-uniform kernel estimation method we perform deblurring using a modified Richardson-Lucy algorithm based on [Whyte, 2014]:

$$\mathbf{H}\mathbf{x} = \sum_{b=1}^{B} \mathbf{M}_b \mathbf{K}_b \mathbf{x}, \quad \mathbf{H}^{\mathbf{T}}\mathbf{x} = \sum_{b=1}^{B} \mathbf{K}_b^T \mathbf{M}_b \mathbf{x}.$$

- Adapted to spatially variant blur
- Deals separately with saturated and unsaturated pixels
- Includes TV regularization

$$\hat{\mathbf{u}}_{\mathcal{U}}^{t+1} = \hat{\mathbf{u}}_{\mathcal{U}}^{t} \circ \mathbf{H}^T \left(\frac{\mathbf{v} \circ R'\left(\mathbf{H}\hat{\mathbf{u}}^t\right) \circ \mathbf{z}}{R\left(\mathbf{H}\hat{\mathbf{u}}^t\right)} + \mathbf{1} - R'\left(\mathbf{H}\hat{\mathbf{u}}^t\right) \circ \mathbf{z} \right),$$

$$\hat{\mathbf{u}}_{\mathcal{S}}^{t+1} = \hat{\mathbf{u}}_{\mathcal{S}}^{t} \circ \mathbf{H}^T \left(\frac{\mathbf{v} \circ R'\left(\mathbf{H}\hat{\mathbf{u}}^t\right)}{R\left(\mathbf{H}\hat{\mathbf{u}}^t\right)} + \mathbf{1} - R'\left(\mathbf{H}\hat{\mathbf{u}}^t\right) \right).$$

$$\hat{\mathbf{u}}_{unreg}^{t+1} = \hat{\mathbf{u}}_{\mathcal{S}}^{t+1} + \hat{\mathbf{u}}_{\mathcal{U}}^{t+1}$$

$$\hat{\mathbf{u}}^{t+1} = \frac{\hat{\mathbf{u}}_{unreg}^{t+1}}{1 + \nabla_{\mathbf{u}} \mathcal{L}_{prior}(\hat{\mathbf{u}}^t)}.$$

Experiments: basis kernels selection and mixing coefficients

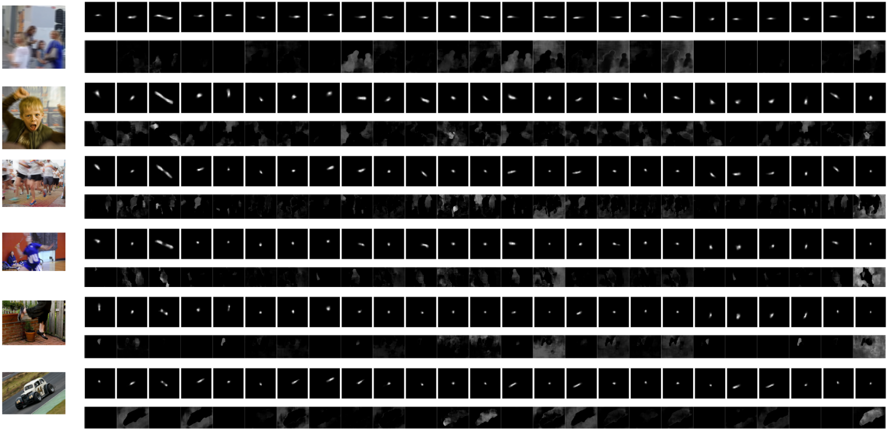


Figure 1: **Examples of generated kernel basis $\{\mathrm{k}^b\}$ and corresponding mixing coefficients $\{\mathrm{m}^b\}$** predicted from the blurry images shown on the left. The adaptation to the input is more notorious for the elements that have significant weights.

Experiments: non-uniform motion blur kernel estimation

Figure 3: **Visual comparison of non-uniform motion blur kernel estimation.** From top to bottom: Gong *et al*. [15], Sun *et al*. [44] and our proposed approach. From left to right: two examples from CUHK blur detection dataset [42], two from GoPro [33] and one from REDs [32]. Best viewed in electronic format.

Experiments: comparison with kernel estimation-based deblurring methods

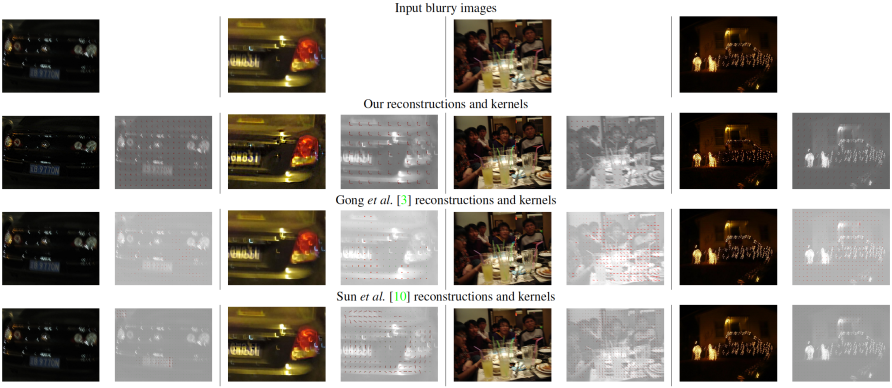
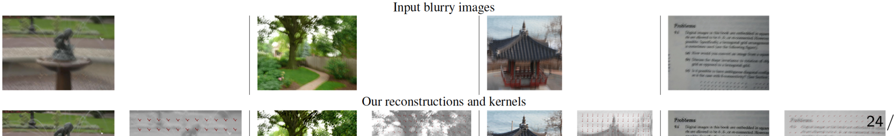


Figure 4: Examples of kernels predicted by our method and corresponding deblurred images on the Lai dataset [6] at half resolution. Comparison with [3] and [10]. Note that these approaches show a significant correlation with the image structure, and are more prone to fail at capturing the motion structure in low contrasted regions.



Experiments: comparison to state-of-the-art motion deblurring methods

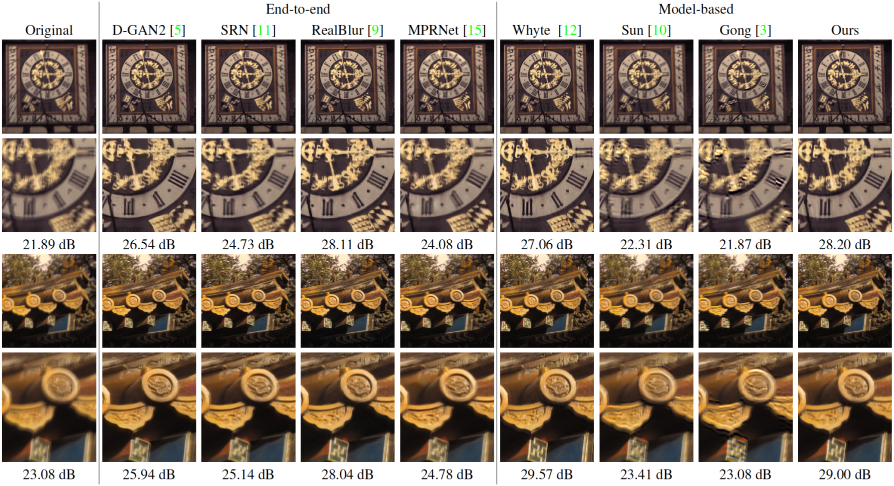


Figure 6: Qualitative comparison of different deblurring methods on Köhler's Dataset [4].

Example of other applications of motion blur kernel estimation: motion segmentation

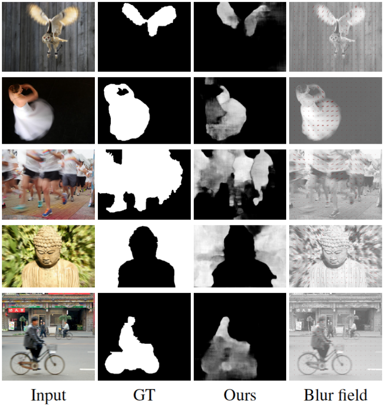


Figure 6: **Blur segmentation.** We use the norm of the predicted non-uniform motion blur kernels to detect regions with motion blur. Images from the CUHK blur detection dataset [42].