

Recording and Playback of Camera Shake: Benchmarking Blind Deconvolution with a Real-World Database

Rolf Köhler, Michael Hirsch, Betty Mohler,
Bernhard Schölkopf, and Stefan Harmeling

Max Planck Institute for Intelligent Systems, Tübingen, Germany
<http://webdav.is.mpg.de/pixel/benchmark4camerashake>

Abstract. Motion blur due to camera shake is one of the predominant sources of degradation in handheld photography. Single image blind deconvolution (BD) or motion deblurring aims at restoring a sharp latent image from the blurred recorded picture without knowing the camera motion that took place during the exposure. BD is a long-standing problem, but has attracted much attention recently, cumulating in several algorithms able to restore photos degraded by real camera motion in high quality. In this paper, we present a *benchmark dataset* for motion deblurring that allows quantitative performance evaluation and comparison of recent approaches featuring non-uniform blur models. To this end, we *record and analyse real camera motion*, which is played back on a robot platform such that we can record a sequence of sharp images sampling the six dimensional camera motion trajectory. The goal of deblurring is to recover one of these sharp images, and our dataset contains all information to assess how closely various algorithms approximate that goal. In a comprehensive comparison, we evaluate state-of-the-art single image BD algorithms incorporating uniform and non-uniform blur models.

Keywords: blind deconvolution, camera shake, benchmark, motion blur.

1 Introduction

Camera motion during exposure is a major problem in handheld photography, as it causes image blur that destroys details in the recorded photo. Especially photos taken in low light without flash suffer from motion blur due to the necessity of longer exposure times. Recently, single image blind deconvolution (BD) has attracted significant attention. Considerable progress was made not only by conceiving more efficient inference strategies but also by proposing more realistic imaging models, better able to capture real camera shake. Levin et al. [1] put together a benchmark dataset for the case of uniform (stationary) blur, allowing objective evaluation and comparison of single image BD algorithms. The lack of a proper benchmark for the case of non-uniform blur, however, makes it difficult to evaluate the performance of recent blind deblurring methods that feature

non-uniform blur models and go beyond the traditional invariant convolution model.

To close this gap, this paper presents a new benchmark dataset that is not restricted to translational motion, but mainly consists of example images with non-uniform blur originating from real camera trajectories. To this end we designed an experimental setup that allows the recording of unconstrained camera motion with full six-dimensional degree of freedom at sub-millimeter precision. Furthermore, we recorded and analysed real camera shake by humans who were asked to take photos with relatively long exposure times. We confirm the finding of [1] that real camera motion does indeed often involve rotational motion, rendering image blur non-uniform across the image plane, however, we also find that the amount of spatial variance of the point spread function (PSF) is often small. In this context, we also investigate and validate recently proposed imaging models [2,3], which consider only three (out of a possible six) degrees of freedom to capture real camera motion.

For playing back real camera shake, we use a Stewart platform (Hexapod) with six degrees of freedom, featuring repeatability at micrometer accuracy. This allows the recording of a sequence of sharp ground truth images, taken along the played-back camera trajectory.

In a comprehensive comparison, we evaluate state-of-the-art single image BD algorithms with both uniform and non-uniform blur models.

2 Related Work and Contributions

Blind deconvolution has a long history and is subject of numerous publications in the image and signal processing literature. Early works were motivated by applications in astronomical imaging and include those of e.g. [4] and [5], which date back to the early 70s. We refer to [6] for an overview of related methods.

One of the first works to apply BD to the problem of removing camera shake from a single photograph was [7], combining the variational approach of [8] with natural image statistics [9]. Subsequent work refined the approach [10], introduced new inference strategies and fast optimisation techniques [11], and proposed methods for robust kernel estimation [12,13,14]. See [1] for a comprehensive overview of these and related approaches.

Recent work [15,16,2,3,17,18,19] focuses on devising new imaging models that are better capable of capturing real motion blur that often violates the uniform blur assumption of previous methods.

Hardware-based approaches to obtain sharper images are based on manipulating the way images are taken. For instance, [20] reconstruct a single sharp image from a pair of blurred and noisy images. While [21] encodes the movement of objects by “fluttering” the shutter, [22] is able to remove linear object motion by capturing two images of the scene with a parabolic motion in two orthogonal directions. [23] exploit inertial measurement sensor data to recover the true trajectory of the camera during exposure.

Contributions of this paper:

1. A new setup for recording and playing back camera shake.
2. An extensible benchmark dataset with ground truth data.
3. A comprehensive comparison of state-of-the-art single image BD algorithms.

3 Recording Trajectories of Human Camera Shake

Human camera shake can be characterized by a six dimensional motion trajectory, with three translational and three rotational coordinates changing through time. We decided to measure such trajectories for several subjects holding a compact camera (Samsung WB600), since such cameras are in wide spread use. Typically, they have a small maximum aperture, and images thus get blurred by camera shake if the available light is insufficient and the use of a flash is excluded. In such cases, an exposure time of 1/3 sec, as used throughout our experiments, is realistic.

To measure the six dimensional camera trajectory we used a Vicon tracking system with 16 high-speed Vicon MX-13 cameras running at a frame rate of 500 Hz. The cameras were calibrated to a cube of roughly 2.5m side length.

To measure the exact position of the camera during a shake, we connected a light-weight but rigid construction with markers (reflective balls, size 35mm), see Figure 1. We synchronized the camera shutter with the trajectory data using the flash of the camera. As the Vicon cameras are only sensitive to infrared light, the flash light was converted from the optical into the infrared spectrum by using a photo diode that triggers an infrared lamp.

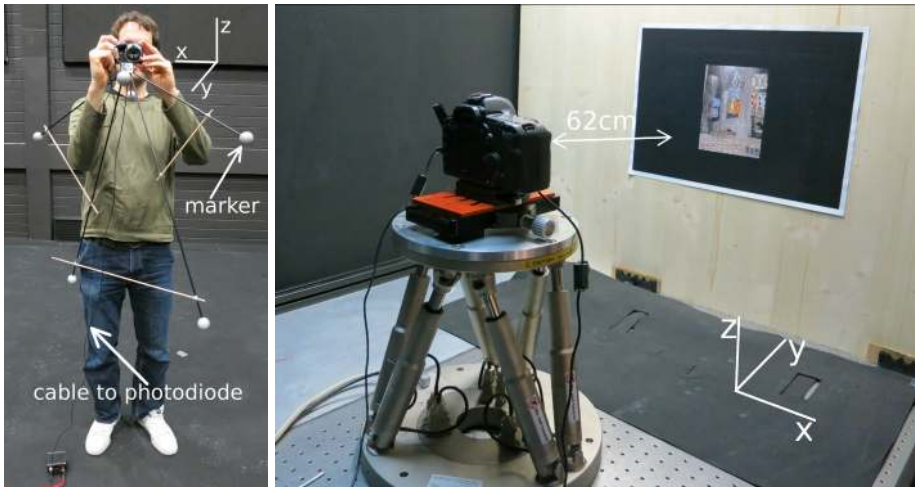


Fig. 1. Setup: (Left) Light-weight structure with reflective spherical markers for recording camera shake with a Vicon system. (Right) Camera attached to a high-precision hexapod robot to take pictures with played back camera shakes in a controlled setting.

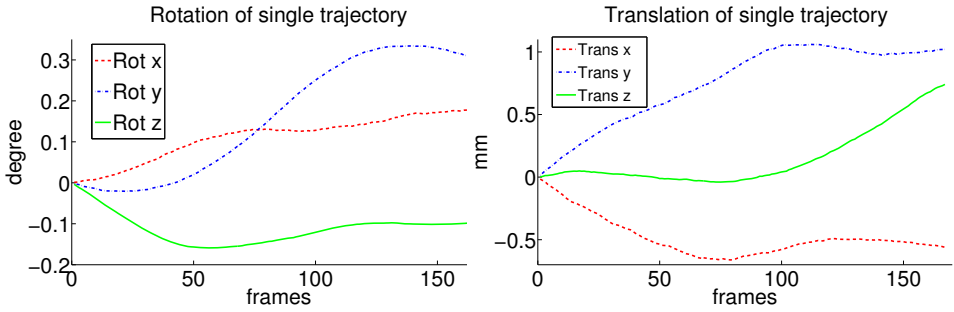


Fig. 2. Camera shake: trajectories of the rotation angles (left) and translations (right). Figure 3 depicts the corresponding PSF.

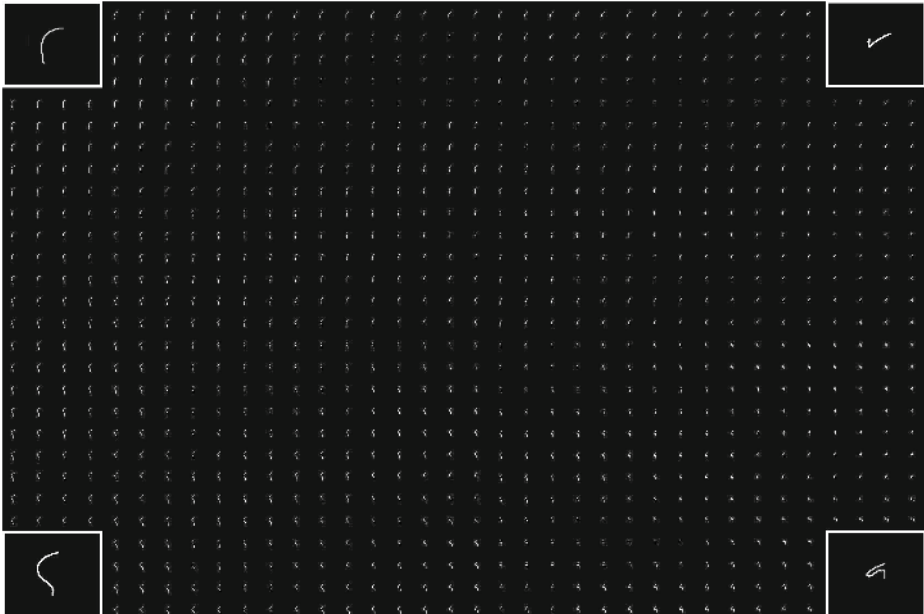


Fig. 3. Example of a point spread function (PSF) due to camera shake. Figure 2 shows the underlying 6D motion in detail. Best viewed on screen rather than in print.

Six subjects were asked to take photographs in a natural way. They were aware that the exposure time was 1/3 sec. 40 trajectories were recorded with this setup (three subjects with ten trajectories, three subjects with five trajectories; five trajectories had to be excluded due to temporary recording problems). Raw captured motion data is noisy in frequency ranges that can not be due to human motion. The noise was reduced with an appropriately chosen moving average filter. Figure 2 shows example filtered trajectories. Figure 3 shows an exemplary point spread function (PSF) obtained by simulating¹ an image of an artificial point grid (see Section 5.1 for details).

In the next section, we explain how such camera motions can be played back on a robot. This allows us to assess how accurate the motion capture was, and whether such camera trajectories can be used for a benchmark.

4 Playing Camera Shake on a Picture-Taking Robot

The six dimensional trajectories were *played back* on a Stewart platform (hexapod robot, model PI M-840.5PD), which allows minimum incremental motions of $3\mu\text{m}$ (x and y axis), $1\mu\text{m}$ (z axis) and $5\mu\text{rad}$ (rotations) with a repeatability $\pm 2\mu\text{m}$ (x and y axis), $\pm 1\mu\text{m}$ (z axis) and $\pm 20\mu\text{rad}$ (rotations). An SLR camera (Canon Eos 5D Mark II) that can be remote controlled was mounted to the Stewart platform to allow synchronization of the camera trigger with the platform (see Figure 1 for the complete setup). For the benchmark dataset (detailed in the following) a printed image was mounted at a distance of 62cm from the camera.

To qualitatively assess whether a real camera shake movement can be recorded and played back by our setup, we took long exposure images of a point grid of LEDs with a somewhat exaggerated camera shake (to obtain large visible blurs). Simultaneously, we recorded the six dimensional camera trajectory. The recorded trajectory was played back on the Stewart platform and the attached camera recorded a long exposure of the same point grid. Figure 4 shows that the real camera shake (left images) is correctly simulated by the Stewart platform (right images). The existing difference in the images is caused by the fact that different cameras were used for recording and playing back the trajectory and that the distance camera to point-grid could not guaranteed to be exactly 62cm at recording. Nonetheless, in both examples (upper and lower images) the true image blurred by camera shake and the image generated by playing back the camera trajectory are reasonably similar. This shows that our setup is able to capture the movement of real camera shake and to play it back again.

Note that the playback on the Stewart platform was performed in slower motion (duration 1sec instead of 1/3sec) to increase the smoothness of the movement — stretching time does not change the obtained PSF.

Note that we decided against using the point grid of LEDs during the recording of the human camera trajectories for the benchmark. We found that the small motions of real camera shake (where the photographer tries to keep the camera

¹ A distance of 62cm, focal length of 50mm were assumed.

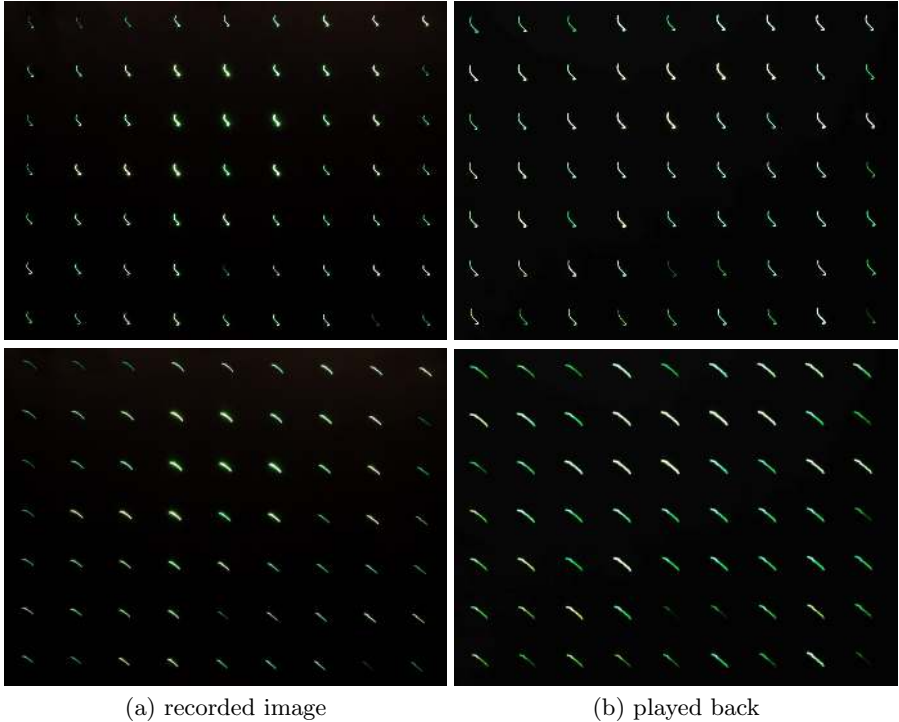


Fig. 4. Two examples (first and second row) of an image with camera shake and its counterpart generated by playing back the camera trajectory on a Stewart platform

steady) can not be seen well with the LED point grid. The grid would have to be positioned rather close to the camera, resulting in an unnatural feeling of tightness, normally not present while taking a photograph.

5 Analyzing Camera Shake Trajectories

5.1 Is Camera Shake Uniform or Non-uniform?

Often surprisingly, blind deconvolution algorithms for uniform blur work quite well on real world camera shake, even though theoretically, the resulting blur should be non-uniform, i.e. varying across the image. To study this effect with our trajectories we generate a blurred point grid for a given camera trajectory, see Figure 3 for an example.

We denote the camera trajectory by a point sequence p_1, \dots, p_T where each point has six dimensions,

$$p_t = [\theta_x, \theta_y, \theta_z, x, y, z]^T \quad (1)$$

with the last three coordinates being the shifts in mm, the axis positioned indicated in Figure 1, the first three coordinates being the rotation angle around the indicated axis.

Given an image u showing a point grid (14×10 equispaced points, size 2808×1872), we can visualize the point spread function (PSF) of the camera trajectory by converting each point p_t in time into its corresponding homography H_t (assuming distance of 2m for the point grid), assigning p_1 the identity transformation. Then the resulting blurry image v is the superposition of the transformed point grids:

$$v = \sum_{t=1}^T h_t(u) \quad (2)$$

where $h_t(u)$ is the image u transformed by H_t . The local blur kernels of the resulting PSF ξ can be read off the blurry image v . To quantify the non-uniformness (abbreviated NU) of the PSF ξ , we introduce the following measure: given four local blur kernels $\xi_{lr}, \xi_{ur}, \xi_{ll}, \xi_{ul}$ of the lower right, upper right, lower left and upper left corner, we can calculate how similar they are by comparing the lower left with the upper right and the upper left with the lower right blur kernel:

$$\text{NU}(\xi) = \frac{\|\xi_{ll} - \xi_{ur}\|^2 + \|\xi_{ul} - \xi_{lr}\|^2}{2}, \quad (3)$$

where each blur kernel is normalized to have L2-norm one. Note that the difference between two blur kernels has to be minimized over all possible shifts (omitted in the formula for clarity). This can be achieved efficiently by noting that $\|\xi_1 - \xi_2\|^2 = \xi_1^T \xi_1 + \xi_2^T \xi_2 - 2\xi_1^T \xi_2$, where the first summands are one due to normalization and the inner products for different shifts are the cross correlations between ξ_1 and ξ_2 . Minimizing $\|\xi_1 - \xi_2\|^2$ for all possible shifts is thus done by maximizing the cross-correlation between ξ_1 and ξ_2 .

Note that for a perfectly uniform blur (pure shift trajectory along x and z axes), i.e. a camera that has no rotations and no y component, the non-uniformness index NU is equal to zero. Figure 5 shows the extreme corners of two PSF images generated from recorded camera trajectories. The left example is the most uniform blur according to the NU criterion, the right example is the most non-uniform blur. Also qualitatively, we observe that the NU criterion captures non-uniformness. In the middle is the histogram of the NU values of all 40 PSFs.

5.2 How Well Can We Approximate the 6D Trajectory with 3D?

To decrease the complexity of possible PSFs, the six dimensional real camera trajectory is often approximated with three dimensions. There are two main approaches:

- (a) ignore the shifts and model only three rotations ($\theta_x, \theta_y, \theta_z$) [2].
- (b) model only shifts (x and z) and rotation around the visual axis (θ_y) [3,18],

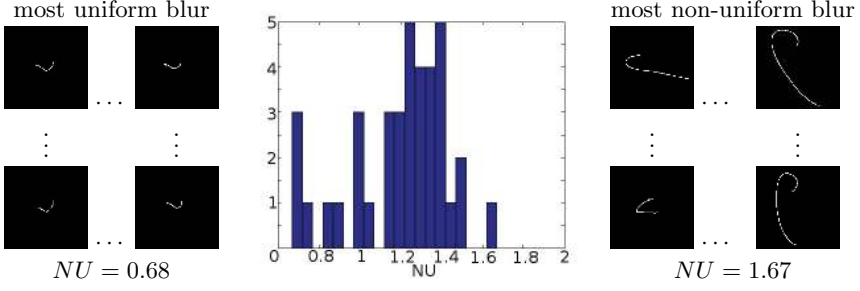


Fig. 5. PSF images of recorded camera trajectories being most uniform (left) and most non-uniform (right) according to the NU criterion. Histogram of all NU values (middle).

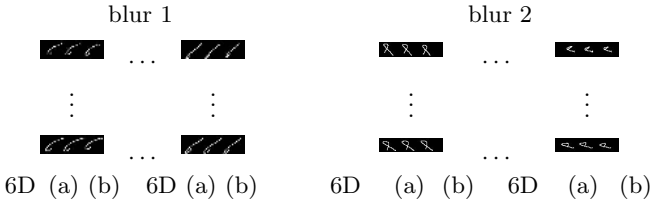


Fig. 6. Mapping the 6D camera trajectory to 3D using Whyte et al.’s approach (a) and Gupta et al.’s approach (b). Each image patch compares the blur due to the original 6D representation to (a) and (b). We set the focal length to 50mm and the object distance to 2m.

Both approximations achieve good results, as can be seen in Figure 6, so we would like to analyze whether this is in correspondence with our recorded trajectories. To this end, we transform the trajectories to three dimensions following the two approaches. Note that the influence of the angles on the shifts and vice versa depends on the distance d of the object to the camera:

$$\begin{aligned}
 \text{(a) } p_t = \begin{bmatrix} \theta_x \\ \theta_y \\ \theta_z \\ x \\ y \\ z \end{bmatrix} &\mapsto \begin{bmatrix} 0 \\ \theta_y \\ 0 \\ x - d \sin(\theta_z) \\ 0 \\ z + d \sin(\theta_x) \end{bmatrix} & \quad \text{(b) } p_t = \begin{bmatrix} \theta_x \\ \theta_y \\ \theta_z \\ x \\ y \\ z \end{bmatrix} &\mapsto \begin{bmatrix} \theta_x - \arcsin(x/d) \\ \theta_y \\ \theta_z + \arcsin(z/d) \\ 0 \\ 0 \\ 0 \end{bmatrix}
 \end{aligned}$$

Assuming an object distance of $d = 2m$ we transform all 40 camera trajectories to three dimensions using mapping (a) and mapping (b). Figure 6 shows based on two examples that both 3D transformations are qualitatively valid. For future research it might be interesting whether there exists another *canonical* transformation from 6D to 3D which preserves the PSFs.

6 Benchmark Dataset for Real-World Camera Shakes

6.1 Recording Images with Played Back Camera Shake

As described in Section 4 the image blurs created by the Stewart platform are good approximations to real image blurs. So we create a benchmark dataset of images blurred by camera shake by playing back human camera shakes on the hexapod for different images, see Figure 7.

We randomly selected two blur trajectories of each of the six subjects applying them to four images (Figure 7) resulting in 48 blurry images. Some local blur kernels turned out rather large. We decided to leave those blur kernels in the benchmark, as they reflect natural camera shake, probably caused by holding the camera too relaxed and not paying attention to hold it still, as could happen in real imaging situations. Even though such images are often erased, it is interesting whether current or future deblurring algorithms can cope with such large blur kernel sizes.



Fig. 7. The four original images used in the benchmark

As discussed in Section 5.1 some of the blurs are approximately uniform and some are not. Eyeballing the generated PSFs (included in the supplementary material), the blur kernels numbered 1,3,4,5,8,9,11 (see table 1) tend to be approximately uniform, while blur kernels 2,6,7,10,12 appear non-uniform, in accordance with the NU criterion. Kernels 8 and 11, although having a high NU value still visually appear to be uniform. The non-uniformness detected by the NU value is not clearly visible due to the large size of the kernels.

For recording the image database, the Stewart platform was placed inside a light-tight box. The SLR camera was set to ISO 100, aperture $f/9.0$, exposure time 1sec, taking images in the Canon raw format SRAW2. A Canon EF 50mm $f/1.4$ lens was used. The illumination inside the box was adjusted to the camera parameters and kept constant for all images taken. The true sharp image was mounted as a poster at a distance of 62cm to the camera. The complete setup is shown in Figure 1.

The recorded SRAW2 images (16bit) were processed by DCRAW², generating 8bit images without correcting gamma. From the center of the image, a 800×800 patch was cropped. The blurry images are named (i, j) for the i -th image blurred with the j -th camera trajectory.

² DCRAW was called with parameters `-W -g 1 1 -r 1.802933 1.0 2.050412 1.0 -b 4)`

6.2 Recording Ground Truth Images

Even though the original images are available since we printed them onto posters, comparing these images with deblurred images from some algorithm is not easy, because of different image resolution, different lighting, and possible color differences due to the printing process. Instead our robotic setup allows us to generate ground truth images *along* the trajectory by playing the camera trajectory step by step and taking an image per step.

Using this strategy, we recorded for each of the 48 blurry images (12 camera shakes each blurring 4 images) 167 images along the trajectory. Additionally, we created 30 images at intermediate positions calculated by the hexapod during playback. We denote the ground truth images by u_1^*, \dots, u_N^* .

6.3 Measuring the Deblurring Performance

To compare similarity between two images a and b (represented as vectors), we first estimate the optimal scaling $\hat{\alpha}$ and translation \hat{T} such that the L2 norm between a and b becomes minimal³, i.e. $\hat{\alpha}, \hat{T} = \min_{\alpha, T} \|a - T(\alpha b)\|^2$. We then calculate the peak-signal-to-noise ratio (PSNR) as

$$\text{PSNR}(a, b) = 10 \log_{10} \frac{m^2}{\langle \|a_i - \hat{T}(\hat{\alpha} b_i)\|^2 \rangle_i} \quad (4)$$

with $\langle \cdot \rangle_i$ denoting an average over pixels and m being the maximal possible intensity value, i.e. $m = 255$ as we work with 8bit encoding. Given a sequence of ground truth images u_1^*, \dots, u_N^* along the trajectory, we define the PSNR similarity between an estimated image \hat{u} and the ground truth as the maximum PSNR between \hat{u} and any of the images along the trajectory,

$$\text{SIM} = \max_n \text{PSNR}(u_n^*, \hat{u}). \quad (5)$$

7 Comparing State-of-the-Art Algorithms in Motion Deblurring

In the following comparative evaluation, we present the results of current state-of-the-art algorithms in single image BD on our benchmark dataset. The algorithms can be divided into two groups:

- Algorithms which assume a uniform blur model and account for translational motion only [7,10,11,12,13].
- Algorithms which assume a non-uniform blur model. In particular, [2,19] assumes rotational motion (yaw, pitch and roll), while [18] considers translations and in-plane rotations only.

³ We allow for integer pixel translations only, which we estimate with the Matlab function `dftregistration` by [24].

Table 1. PSNR values of various state-of-the-art motion deblurring algorithms for all 48 images of our benchmark dataset. Color code : blue corresponds to low PSNR (poor performance), green to intermediate and red to high PSNR values (good performance).

	Blurred	Cho [11]	Xu [12]	Shan [10]	Fergus [7]	Krishnan [13]	Whyte [19]	Hirsch [18]
(1,1)	27.577	33.948	33.236	31.366	20.994	33.849	33.999	33.161
(1,2)	30.529	32.971	34.202	32.290	33.451	33.721	34.276	32.975
(1,3)	35.034	33.164	35.003	33.921	37.436	34.609	37.880	35.776
(1,4)	30.234	31.521	33.720	31.532	34.466	33.095	34.364	33.527
(1,5)	26.643	33.008	34.259	28.419	19.434	30.038	33.940	33.723
(1,6)	27.617	31.171	33.310	32.129	26.480	27.702	33.729	33.358
(1,7)	27.222	33.460	33.899	30.528	21.229	28.152	31.680	32.902
(1,8)	22.120	24.940	26.328	21.683	21.282	15.978	21.791	21.989
(1,9)	22.541	29.056	28.681	26.154	22.142	24.742	22.561	26.672
(1,10)	23.248	25.937	26.748	24.144	22.334	23.211	22.816	23.507
(1,11)	25.582	26.949	28.354	25.857	20.935	24.679	26.534	24.365
(1,12)	27.059	31.187	31.953	25.002	26.726	26.765	27.241	28.256
(2,1)	21.611	28.516	28.522	26.002	16.730	28.448	27.883	29.830
(2,2)	24.692	29.155	29.542	25.229	26.672	29.921	27.790	30.036
(2,3)	28.996	27.812	29.028	27.471	24.965	28.549	30.120	29.731
(2,4)	25.148	27.298	28.290	24.272	25.726	27.934	27.698	29.248
(2,5)	20.858	28.096	28.704	23.091	13.769	24.382	28.056	28.987
(2,6)	21.889	26.611	26.985	21.284	20.845	22.278	27.060	22.944
(2,7)	21.907	27.305	27.838	22.336	19.711	22.909	26.744	27.624
(2,8)	16.438	20.809	20.411	16.432	15.572	13.186	16.422	16.302
(2,9)	17.023	26.745	26.322	19.799	15.357	20.096	17.051	20.367
(2,10)	17.269	22.139	23.451	18.221	15.062	18.104	17.063	17.192
(2,11)	19.578	22.568	24.853	21.998	14.356	19.697	20.585	22.057
(2,12)	20.249	25.295	25.737	19.572	15.523	20.037	27.084	24.562
(3,1)	27.530	35.754	35.551	31.976	29.262	33.454	36.249	33.804
(3,2)	30.435	33.519	34.555	29.240	32.388	32.535	35.884	32.498
(3,3)	35.483	34.369	33.196	33.139	37.719	34.487	35.972	36.822
(3,4)	30.928	33.654	34.383	31.559	32.972	32.189	35.216	33.842
(3,5)	34.807	34.493	35.062	34.527	34.909	34.227	35.636	34.490
(3,6)	28.295	33.551	32.445	28.171	24.555	25.843	34.188	33.122
(3,7)	28.199	33.543	33.862	28.898	27.216	29.232	33.058	32.325
(3,8)	22.386	22.611	22.782	21.989	20.188	17.546	20.877	22.122
(3,9)	21.716	29.673	31.490	24.093	21.177	22.752	21.629	25.560
(3,10)	23.405	25.334	24.856	22.163	20.733	21.344	24.333	22.218
(3,11)	26.147	26.289	26.667	26.106	17.128	22.863	27.209	24.490
(3,12)	26.282	32.974	32.502	25.459	18.292	24.657	33.331	28.387
(4,1)	22.828	31.753	32.476	28.346	15.523	30.540	31.329	31.367
(4,2)	25.732	31.796	32.033	26.353	25.011	28.197	31.702	24.839
(4,3)	29.834	30.409	31.643	28.560	25.919	29.030	33.348	33.449
(4,4)	26.061	31.094	31.650	27.034	24.748	29.853	32.016	31.905
(4,5)	34.587	33.937	33.879	34.125	34.921	33.688	34.063	34.067
(4,6)	23.083	28.715	28.985	25.834	17.495	22.537	29.567	28.715
(4,7)	22.423	29.349	29.994	25.547	14.634	22.919	27.213	24.414
(4,8)	17.270	20.995	20.623	16.667	16.236	14.670	17.265	17.531
(4,9)	17.990	26.506	26.893	21.556	16.794	20.987	18.031	21.396
(4,10)	18.077	22.413	21.414	17.413	16.747	17.938	17.823	17.823
(4,11)	20.274	22.338	23.612	20.291	15.380	19.475	19.919	22.675
(4,12)	21.818	26.461	27.859	25.001	19.912	21.822	27.209	26.048

low PSNR

high PSNR

For fair comparison, we asked the authors of the tested algorithms to run their code on our benchmark dataset and to provide us with their best deblurring results. The results reported for [10,12,18,19] were provided by the authors, while for [7,11,13] we used provided code to yield the deblurring results ourselves. Regarding parameter settings, we followed the provided instructions or the personal advice of the authors and did our best to optimally adjust the free deblurring parameters incl. kernel size (and image patch in the case of [7]). Table 1 reports the PSNR for all of the 48 benchmark images, where we computed the PSNR according to Eq. (5) as detailed in the previous section. For better visual assessment we color coded the results from blue (low PSNR, poor performance), green (intermediate PSNR) to red (high PSNR, good deblurring performance). All deblurred images incl. the estimated kernels are shown in the supplementary material.

Insights:

- Not all deblurring algorithms are able to improve image quality, which is due to artifacts which arise during the deblurring process.
- Overall performance is content specific as evident by the horizontal band structure in Table 1. In particular, image motives 1 and 3 yield consistently better deblurring results.
- All except for the methods of [11,12] have difficulties with large blur (kernel 8-11).
- The algorithm of Xu et al. [12], which features a robust two-phase kernel procedure, performs best despite its assumption of a uniform blur model, followed by the recently proposed algorithm of Whyte et al. [19] which allows for non-uniform blur.
- More recently proposed algorithms yield a better overall performance, being evidence of the progress in the field.

Although runtime is a critical factor and a discriminative feature of deblurring algorithms, we do not report any runtimes here as the results were obtained by heterogenous implementations (e.g. Matlab vs. C) and different hardware systems (e.g. CPU vs. GPU).

8 Discussion and Conclusion

In this paper, we presented a new benchmark dataset for evaluating single image BD algorithms. To this end we first recorded and analysed real camera shake and investigated how well currently employed imaging models approximate and capture true motion blur. To mimick real camera shake we employed a robot platform featuring six degrees of freedom and therefore fully capable of replaying the recorded camera trajectories, while at the same time providing the opportunity to sample the true camera motion trajectory with steady image captures to yield a sequence of ground truth images.

The benchmark has been designed to be extendable. In ongoing work we further investigate the limitations of current approaches by adding complexity

to the imaged scenes such as varying depth. We intend to extend our benchmark for new sets of images and camera shakes where current methods fail to give satisfactory results. At the same time we examine other quality measures than PSNR, some of which hold the promise to better resemble human perception.

Another interesting future direction is to study whether users exhibit repeating camera shake patterns. Such a finding would motivate to learn personalized priors which ultimately could facilitate BD.

The benchmark dataset is publicly available at the accompanying project webpage⁴. We hope that it will find widespread acceptance within the community as a useful tool to evaluate the performance of new BD algorithms, rendering it a valuable resource for monitoring the state-of-the-art in the field.

References

1. Levin, A., Weiss, Y., Durand, F., Freeman, W.T.: Understanding and evaluating blind deconvolution algorithms. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR (2009)
2. Whyte, O., Sivic, J., Zisserman, A., Ponce, J.: Non-uniform deblurring for shaken images. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR (2010)
3. Gupta, A., Joshi, N., Lawrence Zitnick, C., Cohen, M., Curless, B.: Single Image Deblurring Using Motion Density Functions. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part I. LNCS, vol. 6311, pp. 171–184. Springer, Heidelberg (2010)
4. Richardson, W.H.: Bayesian-based iterative method of image restoration. *Journal of the Optical Society of America* 62, 55–59 (1972)
5. Lucy, L.B.: An iterative technique for the rectification of observed distributions. *The Astronomical Journal* 79, 745–754 (1974)
6. Kundur, D., Hatzinakos, D.: Blind image deconvolution. *Signal Processing Magazine* 13, 43–64 (1996)
7. Fergus, R., Singh, B., Hertzmann, A., Roweis, S.T., Freeman, W.T.: Removing camera shake from a single photograph. *ACM Transactions on Graphics, SIGGRAPH* (2006)
8. Miskin, J., MacKay, D.J.C.: Ensemble learning for blind image separation and deconvolution. In: *Advances Independent Component Analysis* (2000)
9. Field, D.J.: What is the goal of sensory coding? *Neural Computation* 6, 559–601 (1994)
10. Shan, Q., Jia, J., Agarwala, A.: High-quality motion deblurring from a single image. *ACM Transactions on Graphics, SIGGRAPH* (2008)
11. Cho, S., Lee, S.: Fast motion deblurring. *ACM Transactions on Graphics, SIGGRAPH ASIA* (2009)
12. Xu, L., Jia, J.: Two-Phase Kernel Estimation for Robust Motion Deblurring. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part I. LNCS, vol. 6311, pp. 157–170. Springer, Heidelberg (2010)
13. Krishnan, D., Tay, T., Fergus, R.: Blind deconvolution using a normalized sparsity measure. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR (2011)

⁴ <http://webdav.is.mpg.de/pixel/benchmark4camerashake>

14. Levin, A., Weiss, Y., Durand, F., Freeman, W.T.: Efficient marginal likelihood optimization in blind deconvolution. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR (2011)
15. Tai, Y.W., Tan, P., Brown, M.S.: Richardson-lucy deblurring for scenes under a projective motion path. Technical Report of Korea Advanced Institute of Science and Technology, KAIST (2009)
16. Tai, Y.W., Tan, P., Brown, M.S.: Richardson-lucy deblurring for scenes under a projective motion path. IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI (2011)
17. Harmeling, S., Hirsch, M., Schölkopf, B.: Space-variant single-image blind deconvolution for removing camera shake. In: Advances in Neural Information Processing Systems, NIPS (2010)
18. Hirsch, M., Schuler, C.J., Harmeling, S., Schölkopf, B.: Fast removal of non-uniform camera-shake. In: Proceedings of the IEEE International Conference on Computer Vision, ICCV (2011)
19. Whyte, O., Sivic, J., Zisserman, A.: Deblurring shaken and partially saturated images. In: Proceedings of the IEEE Workshop on Color and Photometry in Computer Vision, with ICCV 2011 (2011)
20. Yuan, L., Sun, J., Quan, L., Shum, H.Y.: Image deblurring with blurred/noisy image pairs. ACM Transactions on Graphics, SIGGRAPH (2008)
21. Raskar, R., Agrawal, A., Tumblin, J.: Coded exposure photography: motion deblurring using fluttered shutter. ACM Transactions on Graphics, SIGGRAPH (2006)
22. Cho, T.S., Levin, A., Durand, F., Freeman, W.T.: Motion blur removal with orthogonal parabolic exposures. In: IEEE International Conference in Computational Photography, ICCP (2010)
23. Joshi, N., Kang, S.B., Zitnick, C.L., Szeliski, R.: Image deblurring using inertial measurement sensors. ACM Transactions on Graphics, SIGGRAPH (2010)
24. Guizar-Sicairos, M., Thurman, S.T., Fienup, J.R.: Efficient subpixel image registration algorithms. Optical Letters 33, 156–158 (2008)