sensors

MDPI

Article

Edge Detection of Motion-Blurred Images Aided by Inertial Sensors

Luo Tian [1], Kepeng Qiu [1], Yufeng Zhao [2] and Peng Wang [1,*]

[1] Department of Precision Instrument, Tsinghua University, Beijing 100084, China; tian-l18@mails.tsinghua.edu.cn (L.T.); qiukp@mail.tsinghua.edu.cn (K.Q.)
[2] Heilongjiang North Tool Co., Ltd., Mudanjiang 157000, China; zhaoyufeng@beifang.ntesmail.com
* Correspondence: peng@mail.tsinghua.edu.cn; Tel.: +86-10-6277-2007

Abstract: Edge detection serves as the foundation for advanced image processing tasks. The accuracy of edge detection is significantly reduced when applied to motion-blurred images. In this paper, we propose an effective deblurring method adapted to the edge detection task, utilizing inertial sensors to aid in the deblurring process. To account for measurement errors of the inertial sensors, we transform them into blur kernel errors and apply a total-least-squares (TLS) based iterative optimization scheme to handle the image deblurring problem involving blur kernel errors, whose relating priors are learned by neural networks. We apply the Canny edge detection algorithm to each intermediate output of the iterative process and use all the edge detection results to calculate the network's total loss function, enabling a closer coupling between the edge detection task and the deblurring iterative process. Based on the BSDS500 edge detection dataset and an independent inertial sensor dataset, we have constructed a synthetic dataset for training and evaluating the network. Results on the synthetic dataset indicate that, compared to existing representative deblurring methods, our proposed approach demonstrates higher accuracy and robustness in edge detection of motion-blurred images.

Keywords: edge detection; image deblurring; inertial sensor; neural network

check for updates

Citation: Tian, L.; Qiu, K.; Zhao, Y.; Wang, P. Edge Detection of Motion-Blurred Images Aided by Inertial Sensors. *Sensors* **2023**, *23*, 7187. https://doi.org/10.3390/s23167187

Academic Editor: Shyqyri Haxha

Received: 7 June 2023
Revised: 5 August 2023
Accepted: 13 August 2023
Published: 15 August 2023

1. Introduction

To initiate an exploration into the realm of image processing, one must be drawn to the significance of edges—the fundamental features that underpin visual information. In practical applications, edge detection serves as a pivotal low-level operation, forming the bedrock for various high-level tasks such as feature extraction [1], image segmentation [2], object recognition [3], and object proposal [4]. However, when there is relative motion between the camera and the object during the exposure time, the captured image will appear with motion blur. The direct application of edge detection on motion-blurred images leads to significantly reduced accuracy due to the presence of artifacts, thereby affecting subsequent image processing tasks. Currently, there are two main challenges in performing edge detection on motion-blurred images.

Firstly, edge detection of motion-blurred images requires an effective deblurring method. In previous work, the majority of efforts have focused on utilizing the content of the image itself to remove motion blur [5–8], and it is an ill-posed problem since both the latent image and the blur kernel remain unknown. Inertial sensors, such as gyroscopes and accelerometers, can provide additional motion information about the imaging system during exposure. Utilizing inertial sensors to assist in deblurring can effectively reduce the ill-posedness of deblurring algorithms [9]. Nevertheless, due to time synchronization error and noise, accurate motion information cannot be obtained from sensor data, thus the deblurring methods aided by inertial sensor data often lack robustness [10].

On the other hand, motion deblurring methods have been developed for improvement of the image quality but are not designed for a better edge structure perception. Recent studies have demonstrated that compared to deblurring results with lower peak signal-to-noise ratio (PSNR), deblurring results with higher PSNR do not always achieve better performance in edge detection [11]. Coupling deblurring algorithms with edge detection tasks and ensuring that the deblurring method can effectively improve the edge detection accuracy is another key problem to be addressed.

Based on the above issues, our work made the following contributions:

- The sensor data with errors are transformed into blur kernel with errors, and we apply a TLS-based iterative optimization scheme to handle the image deblurring problem involving blur kernel errors, whose relating priors are learned by two types of neural networks. The inclusion of the blur kernel with sensor data error information in the training process makes the final deblurring method strongly robust.
- The canny edge detection algorithm is incorporated into the deblurring process for calculation of the final loss function. By coupling the edge detection task and the deblurring iterative process more tightly, we ensure that the edge detection task achieves higher accuracy through the image deblurring process.
- The BSDS500 edge detection dataset and an independent inertial sensor dataset are combined to create a synthetic dataset for edge detection of motion-blurred images. The results for the synthetic dataset demonstrates the effectiveness and robustness of the proposed method.

2. Related Work

Deblurring methods have remained an active research area in recent years and usually requires priori knowledge or additional capture information to obtain a valid solution [12,13]. More accurate information about the camera motion can be obtained using inertial sensors, such as gyroscopes and accelerometers, which have been successfully utilized to assist in motion deblurring. Joshi et al. built a single lens reflex camera equipped with a gyroscope and accelerometer to estimate the motion of the camera over the course of the exposure [9]. The sensor data was corrected beforehand under the guidance of a natural image. Park and Levoy employed a similar gyroscope calibration method to address the multiple image deblurring problem [14]. Šindelář and Šroubek developed a real-time deblurring method on mobile devices based on a spatially invariant blur approximation [15]. Due to hardware precision limitations, the sensor data play only a qualitative role in the blur kernel estimation. Zhang and Hirakawa combined inertial measurements and image based information to remove the blur [16]. All of these works assume that the sensor data they recorded is reliable, or they only slightly relax this assumption. However, there are more challenging problems when using inertial sensors that do not provide high quality sensor data for effective image clarification.

Mustaniemi et al. applied gyroscopic data for the first time for single image deblurring based on deep learning networks [17]. In their work, sensor data errors have been taken into account during network training. However, they only used the sensor data to simplify the shape of the blur kernel to a straight line, which would result in partial loss of information in the physical phase and thus would have an impact on the final performance of the Convolutional Neural Network (CNN). Nan and Ji recently proposed a TLS-based iterative optimization scheme for dealing with the kernel error problem in image deblurring [18], which can obtain good deblurring results even when there is an error in the estimation of the blur kernel. This framework is particularly suitable for handling blur kernel errors caused by sensor data errors.

In the aspect of coupling deblurring algorithms with edge detection tasks, an extensive theoretical overview of task-adapted image reconstruction was presented in the work by Adler et al. [19]. Their study revealed that joint reconstruction-segmentation approaches achieved more accurate segmentations compared to both sequential and end-to-end methods. Yang et al. proposed a new cooperative game framework for joint image restoration and edge detection [20]. It used an iterative approach to solve the two tasks, and the interactive facilitation between the tasks during iteration resulted in improvements in both image restoration and edge detection performance.

Despite these efforts, effectively addressing edge detection tasks in the presence of motion blur remains a challenging endeavor. The key aspect lies in ensuring the efficacy of the deblurring process while also striving to achieve higher edge detection precision with the deblurred results.

3. Method

3.1. Initial Kernel Estimation

Recall the spatially-invariant convolutional model of the image blurring process: $\boldsymbol{g} = \boldsymbol{f} * \boldsymbol{k} + \boldsymbol{n}$, where $\boldsymbol{g}$ and $\boldsymbol{f}$ denote the motion-blurred image and its latent sharp image, * is the convolution operator, $\boldsymbol{k}$ is the blur kernel and the noise term $\boldsymbol{n}$ is often formulated as the additional white Gaussian noise.

In order to incorporate the inertial sensor data that record the camera motion information, the motion-blurred image is formulated as the summation of multiple sharp images under a sequence of projective motions during the exposure interval:

$$\boldsymbol{g}(\boldsymbol{x}) = \frac{1}{Np} \sum\nolimits_{t=1}^{N_p} \boldsymbol{f}(\boldsymbol{H}_t \boldsymbol{x}) + \boldsymbol{n} \tag{1}$$

where $\boldsymbol{x} \in \mathbb{R}^{3\times1}$ denotes the pixel location in homogeneous coordinate, $\boldsymbol{H}_t$ denotes the homography matrix, $\boldsymbol{n}$ denotes the noise term and N_p denotes the number of all camera poses during the exposure time. Considering the planar homography that maps the initial projection of points at $t = 0$ to any other time t, the homography matrix $\boldsymbol{H}_t$ can be characterized as [21]:

$$\boldsymbol{H}_t = \boldsymbol{K}(\boldsymbol{R}_t + \frac{\boldsymbol{T}_t \boldsymbol{N}^T}{d})\boldsymbol{K}^{-1} \tag{2}$$

for a particular depth d. $\boldsymbol{R}_t$ is the rotation transformation matrix, $\boldsymbol{T}_t$ is the translation vector, and $\boldsymbol{N}^T$ is the unit vector that is orthogonal to the image plane. The camera intrinsic matrix $\boldsymbol{K}$ can be characterized by the focal length $\boldsymbol{f}$ and camera optical center (O_x, O_y):

$$\boldsymbol{K} = \begin{bmatrix} f & 0 & O_x \\ 0 & f & O_y \\ 0 & 0 & 1 \end{bmatrix} \tag{3}$$

Parameters related to motion, the rotation transformation matrix $\boldsymbol{R}_t$ and the translation vector $\boldsymbol{T}_t$, can be calculated from the measurements of the gyroscope and accelerometer, respectively. The process of image blurring caused by camera motion is shown in Figure 1.

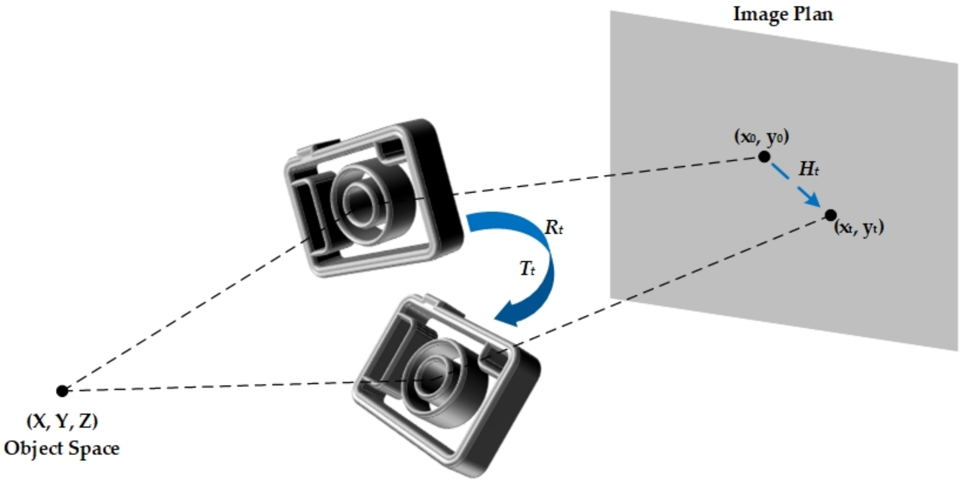

Figure 1. Image blurring model caused by camera motion.

Assuming that the rotation center locates at the optical center of the camera, the rotation transformation matrix can be approximated as

$$\boldsymbol{R}_t = \begin{bmatrix} 1 & -d\theta_t^z & d\theta_t^y \\ d\theta_t^z & 1 & -d\theta_t^x \\ -d\theta_t^y & -d\theta_t^x & 1 \end{bmatrix} \boldsymbol{R}_{t-1} \tag{4}$$

by employing the sinusoidal approximation when the angular rotation is small. Given the gyroscope measurement $\boldsymbol{\omega}_t = \left[\omega_{tx},\ \omega_{ty},\ \omega_{tz}\right]^T$ at time t and the sampling interval Δt, $\left[d\theta_t^x,\ d\theta_t^y,\ d\theta_t^z\right]^T = \boldsymbol{\omega}_t \times \Delta t$. Since only relative rotation is considered, the initial rotation transformation matrix $\boldsymbol{R}_0 = Identity$.

As for the translation vector $\boldsymbol{T}_t$, since there are currently mobile devices that can eliminate the influence of gravity using data from other sensors, we can perform a double integration on the accelerometer measurement $\boldsymbol{a}_t = \left[a_{tx},\ a_{ty},\ a_{tz}\right]^T$ without subtracting the gravitational acceleration.

After calculating the homography matrix $\boldsymbol{H}_t$ at any given moment t, we can ultimately obtain the projected trajectory within the exposure time. Moreover, we approximate the blur kernel of the entire image using the projected trajectory of the central pixel point. Specifically, we set all pixel values to 0 except for the central coordinates, where the pixel value is set to 255, to obtain the image $\boldsymbol{f}^c(\boldsymbol{x})$. Then, we set the white noise to 0 and use $\boldsymbol{f}^c(\boldsymbol{x})$ as input in Equation (1) to compute the image $\boldsymbol{g}^c(\boldsymbol{x})$, which represents the initial blur kernel $\boldsymbol{k}$.

Due to the influence of time synchronization error and noise, the direct calculation of blur kernel using sensor data will lead to the difference between the estimated blur kernel and the exact blur kernel. Thus, the problem turns to how to use the blur kernel with errors for image deblurring.

3.2. TLS-Based Iterative Optimization Scheme for Blur Kernel with Errors

When considering the blur kernel with errors, the image blurring model is as follows

$$\boldsymbol{g} = \left(\hat{\boldsymbol{K}} - \Delta\boldsymbol{K}\right)\boldsymbol{f} + \boldsymbol{n} = \hat{\boldsymbol{K}}\boldsymbol{f} - \Delta\boldsymbol{K}\boldsymbol{f} + \boldsymbol{n} \tag{5}$$

where $\hat{\boldsymbol{K}}$ is the matrix form of the convolution operator.

The TLS estimator finds the solution to (5) through the resolution of a constrained optimization problem.

$$\min_{\Delta\boldsymbol{K},\boldsymbol{n},\boldsymbol{f}} \|\Delta\boldsymbol{K}\|_F^2 + \|\boldsymbol{n}\|_2^2 \quad s.t.\ \left(\hat{\boldsymbol{K}}\boldsymbol{f} - \Delta\boldsymbol{K}\boldsymbol{f} = \boldsymbol{g} - \boldsymbol{n}\right) \tag{6}$$

By introduction of an auxiliary variable $\boldsymbol{u}$ that represents the kernel error term $\Delta\boldsymbol{K}\boldsymbol{f}$, we reformulate the problem (6) into an optimization problem as follows:

$$\min_{\boldsymbol{f},\boldsymbol{u}} \|\Delta\boldsymbol{K}\|_F^2 + \|\boldsymbol{g} - \hat{\boldsymbol{k}} * \boldsymbol{f} - \boldsymbol{u}\|_2^2 + \lambda\|\boldsymbol{u} - \Delta\boldsymbol{K}\boldsymbol{f}\|^2 + \Phi(\boldsymbol{f}) \tag{7}$$

where $\Phi(\boldsymbol{f})$ denotes the regularization term with respect to certain image prior which is usually imposed on high-frequency image components, as they are the main parts lost in the blurring process. By introduction of an auxiliary variable $\boldsymbol{z}$ and applying the half-quadratic splitting, the problem (7) can be reformulated as:

$$\min_{\boldsymbol{f},\boldsymbol{u},\boldsymbol{z}} \|\boldsymbol{g} - \hat{\boldsymbol{k}} * \boldsymbol{f} - \boldsymbol{u}\|_2^2 + \varphi(\boldsymbol{u}|\boldsymbol{f}) + \|diag(\lambda)(\Gamma\boldsymbol{f} - \boldsymbol{z})\|_2^2 + \rho(\boldsymbol{z}) \tag{8}$$

where $\varphi(\boldsymbol{u}|\boldsymbol{f}) = \min_{\Delta\boldsymbol{K}} \|\Delta\boldsymbol{K}\|_F^2 + \lambda\|\boldsymbol{u} - \Delta\boldsymbol{K}\boldsymbol{f}\|^2$ is the regularization term related to the prior imposition on the correction term caused by kernel error and Γ denotes high-pass filters. An alternating iterative scheme can be employed to solve the optimization problem (7):

$$\boldsymbol{f}^{(t)} = \underset{\boldsymbol{f}}{argmin} \| \boldsymbol{g} - \hat{\boldsymbol{k}} * \boldsymbol{f} - \boldsymbol{u}^{(t-1)} \|_2^2 + \lambda \| diag(\lambda) \left(\Gamma \boldsymbol{f} - \boldsymbol{z}^{(t-1)} \right) \|_2^2 \tag{9}$$

$$\boldsymbol{z}^{(t)} = \underset{\boldsymbol{z}}{argmin} \mu \| \Gamma \boldsymbol{f}^{(t)} - \boldsymbol{z} \|_2^2 + \rho(\boldsymbol{z}) \tag{10}$$

$$\boldsymbol{u}^{(t)} = \underset{\boldsymbol{u}}{argmin} \| \boldsymbol{g} - \hat{\boldsymbol{k}} * \boldsymbol{f}^{(t)} - \boldsymbol{u} \|_2^2 + \varphi(\boldsymbol{u} | \boldsymbol{f}^{(t)}) \tag{11}$$

The first step (9) is an inversion process which can be solved using discrete Fourier transform given $\boldsymbol{u}^{(t-1)}$ and $\boldsymbol{z}^{(t-1)}$ of last iteration.

The second step (10) is a denoising process, which eliminates potential artifacts present in the high-pass image channels. The CNN-based denoising neural network called Dn-CNN [22] can be used to remove noise in $\boldsymbol{f}^{(t)}$.

The third step (11) is a correction process, which corrects the term relating to kernel error. Proposed by Nan and Ji [18], the neural network Dual-Path U-net can be used as a tool to estimate the correction term $\boldsymbol{u}^{(t)}$ by combining the downsampled codes from $\boldsymbol{f}^{(t)}$ and the residual $\boldsymbol{g} - \hat{\boldsymbol{k}} * \boldsymbol{f}^{(t)}$.

By adopting the above framework, we have obtained an effective deblurring method assisted by inertial sensors which could handle blur kernel errors caused by sensor data errors. Our proposed deblurring method can be described as Algorithm 1.

```
Algorithm 1: Deblurring Assisted by Inertial Sensors
Input:  gyroscope data $\{\boldsymbol{w}\}_i$, accelerometer data $\{\boldsymbol{a}\}_i$, blurred image $\boldsymbol{g}$
Output: deblurred image $\boldsymbol{f}$
Procedure:
(1) obtain $\{\boldsymbol{T}\}_t$ performing a double integration on $\{\boldsymbol{a}\}_i$
(2) obtain $\{\boldsymbol{R}\}_t$ using (4)
(3) obtain $\{\boldsymbol{H}\}_t$ using (2), (3)
(4) set the center pixel to 255, and all other pixels to 0, obtaining the image $\boldsymbol{f^c}$
(5) obtain blur kernel $\boldsymbol{k}$ applying (1), using all the obtained $\{\boldsymbol{H}\}_t$ and the image $\boldsymbol{f^c}$ with the
    noise set to 0
(6) initialize $\boldsymbol{z}^0$ and $\boldsymbol{u}^0$ to 0
(7) obtain $\boldsymbol{f}^0$ using discrete Fourier transform to solve (8) with blur kernel $\boldsymbol{k}$
(8) for $iter$ = 1 to $N$ do
        obtain $\boldsymbol{z}^{iter}$ using Dn-CNN with $\boldsymbol{f}^{iter-1}$
        obtain $\boldsymbol{u}^{iter}$ using DP-Unet with blur kernel $\boldsymbol{k}$ and $\boldsymbol{f}^{iter-1}$
        obtain $\boldsymbol{f}^{iter}$ using discrete Fourier transform to solve (8) with blur kernel $\boldsymbol{k}$, $\boldsymbol{z}^{iter}$
        and $\boldsymbol{u}^{iter}$
    end
(9) $\boldsymbol{f}^N$ is the final output $\boldsymbol{f}$
end
```

3.3. *Overall Network Structure with Canny Edge Detection Algorithm Added*

The loss function in the framework proposed by Nan and Ji is defined as [18]:

$$L = \frac{1}{J} \sum_{j=0}^{J} \left(\| \boldsymbol{f}_j^{(T,+,1)} - \boldsymbol{f}_j \|_2^2 + \sum_{i=2}^{T} \mu_i \| \boldsymbol{f}_j^{(i)} - \boldsymbol{f}_j \|_2^2 \right) \tag{12}$$

where $\left\{ \boldsymbol{f}^{(1)}, \boldsymbol{f}^{(2)}, \ldots, \boldsymbol{f}^{(T+1)} \right\}$ are the sequence of deconvoluted images corresponding to $T + 1$ iterations in the optimization algorithm. To ensure that the network's optimization goal is to improve the accuracy of edge detection, we incorporate the effect of edge detection results into the iterative process. Specifically, at each step of the iterative process, we perform Canny edge detection [23] on the output of Equation (11) and use the edge detection result to calculate the edge cross-entropy function

$$l_{edge}^{(i)} = -\frac{1}{J}\sum_{j=0}^{J}\left(e_j^{(i)} log(\hat{e}_j^{(i)}) + \left(1 - e_j^{(i)}\right) log(1 - \hat{e}_j^{(i)})\right) \tag{13}$$

where $e_j^{(i)}$ is the edge detection result of $f^{(i)}$ and $\hat{e}_j^{(i)}$ represents the edge ground truth. The overall loss function is redefined as

$$L = l_{edge}^{(T+1)} + \sum_{i=2}^{T} \mu_i l_{edge}^{(i)} \tag{14}$$

where the weights $\{\mu_i\}_{i=1}^{T-1}$ are set to 0.8. In summary, our deblurring method can be represented by the schematic diagram in Figure 2.

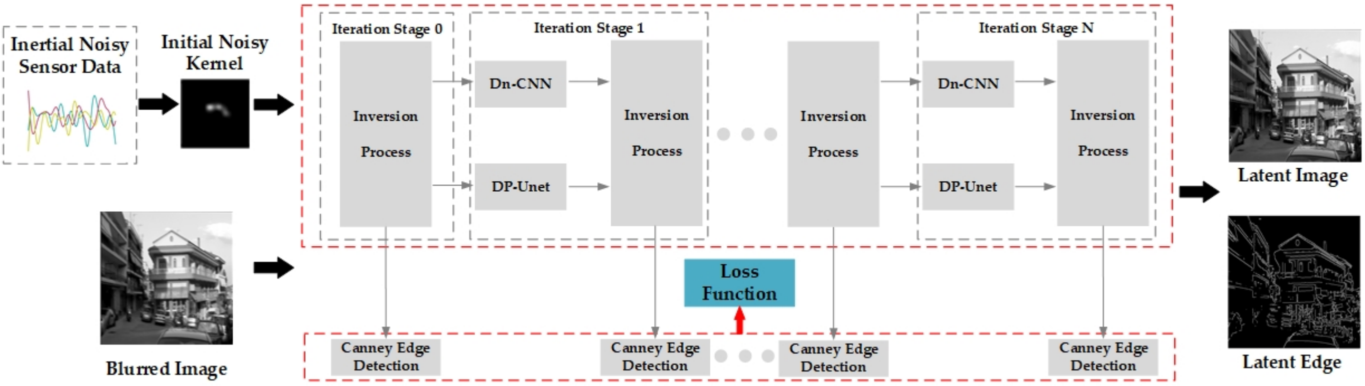

Figure 2. Diagram of the proposed deblurring method.

Incorporating edge detection into the deblurring process and using the edge loss function to adjust network parameters can encourage the model's output to be close to the ground truth edges, as specified by the first term in Equation (14). This can be achieved using currently advanced deblurring methods. However, our approach differs in that we use an iterative optimization-based deblurring algorithm, which allows us to obtain intermediate edge detection results. This ensures that the intermediate results are not too far from the ground truth edges, as specified by the second term in Equation (14). Indeed, our method incorporates richer edge information and is expected to perform better in edge detection tasks for motion-blurred images.

3.4. Synthetic Dataset of Motion-Blurred Images with Inertial Sensor Data

Sufficient high-quality training samples are essential for deep learning-based models. We propose a method for synthesizing a comprehensive dataset that includes ground truth edges, blurred images, and inertial sensor data captured during the exposure time of each blurred image.

We construct our dataset based on the real images and their corresponding ground truth edges from the BSDS500 dataset [24]. The BSDS500 dataset has already partitioned the data into training, validation, and test sets. For each image and its ground truth in the BSDS500 training set, we crop them into patches of 256 × 256. In the direction with pixel value 321, the first cropping uses a stride of 1 pixel, and subsequently, a step size of 32 pixels is used, resulting in a total of 4 patches. In the direction with pixel value 481, the first three croppings use a stride of 1 pixel, the fourth cropping uses a stride of 2 pixels, and thereafter, a stride of 4 pixels is used, resulting in a total of 60 patches (The purpose of doing this is to make the number of patches obtained from cropping adaptable to a wider range of batch sizes). As a result, 200 × 4 × 60 = 48,000 sharp images along with their corresponding ground truth edges are obtained.

The inertial sensor data, specifically the angular velocity and the acceleration of each axis, are modeled using a Gaussian distribution with a mean of 0. The standard deviation of angular velocity of each axis is $\sigma_{\omega x} = \sigma_{\omega y} = 1 \times 10^{-6}$ rad/s and $\sigma_{\omega z} = 0.1$ rad/s; the standard deviation of acceleration of each axis is $\sigma_{ax} = \sigma_{ay} = 1 \times 10^{-3}$ m^2/s and $\sigma_{\omega z} = 1 \times 10^{-5}$ m^2/s. After randomly determining the exposure time within the range of (0.02, 0.2) seconds, we sample the sensor data within the exposure time at a frequency of $f_s = 200$ Hz. To simulate continuous motion, each sensor data sample is interpolated from the preceding data point, with angular velocity data linearly and acceleration data approximated. Utilizing the sharp images and their edges ground truth obtained from cropping the BSDS500 training set, combined with the generated sensor data, the overview of our training data generation scheme is shown in Figure 3.

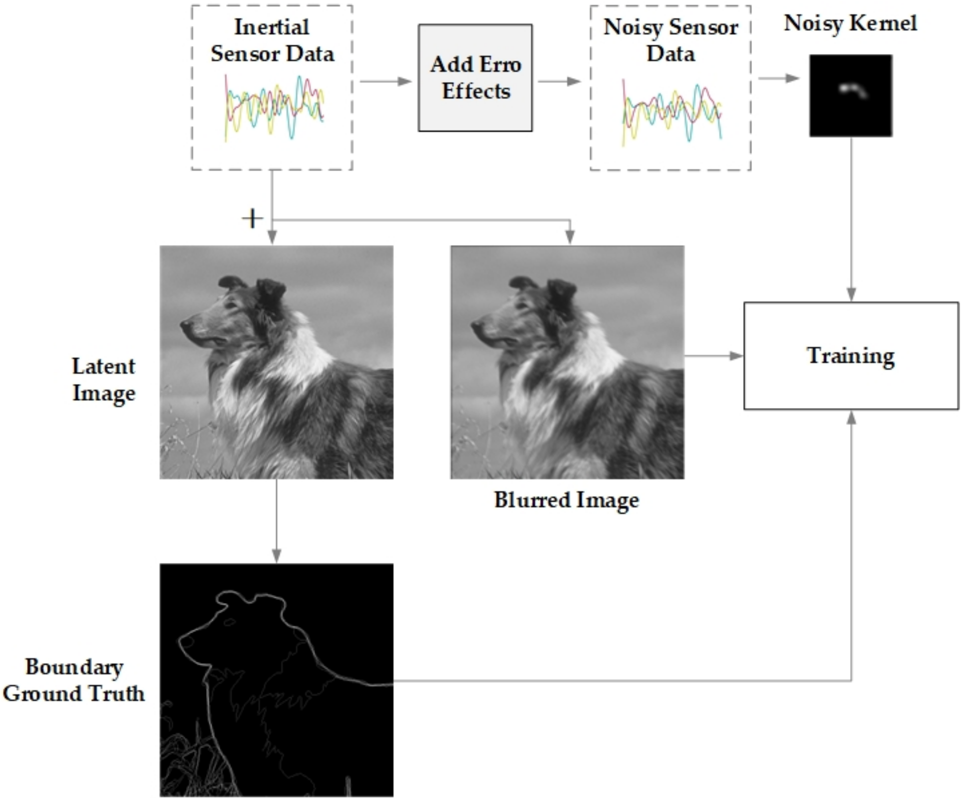

Figure 3. Synthetic training data generation scheme.

It is essential to note that the exact sensor data is directly combined with the clear images to calculate the motion-blurred images for the training set, while the sensor data with errors, obtained by adding synchronization errors and noise terms to the exact sensor data, is used to compute the estimated blur kernel for the training set. Following the steps outlined in [25], we set the time delay t_d randomly picked from a Gaussian distribution $N\left(0.03,\ 0.01^2\right)$ in seconds, and the noise for inertial sensor data as additive white Gaussian noise with standard deviation as 1/10 of the standard deviation of corresponding data.

The validation and test set images of the BSDS500 dataset are not cropped, and we also employ the aforementioned scheme to generate the validation and test sets for our dataset.

4. Experiments

4.1. Experimental Setup

As for $\left\{\lambda^{(t)}\right\}$ in Equation (9), we set $\lambda^{(0)} = 0.005$ for stage 0 and $\lambda^{(t)} = 0.5$ for later stage. The network is trained using the Adam optimizer [26]. The learning rate, training batch size and the number of epochs for network training are set to be 1 × 10^{-4}, 4 and 100, respectively. The iterative parameter N has a significant impact on the performance of the approach we proposed. Therefore, we determine its value through the heuristic method. We set the value of N to range from 2 to 6 and exploit the cross-entropy loss of the final iteration with different N. Figure 4 shows the loss trend on the test set with the change of N. It can be observed that the performance improvement becomes marginal when N exceeds 4. Considering that the network architecture should not be overly complex, N is set to 4.

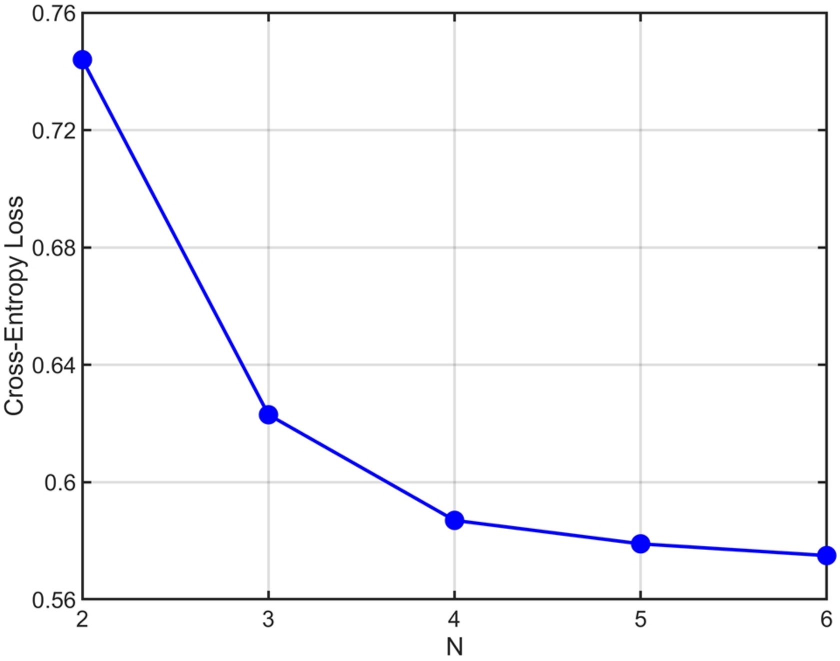

Figure 4. Cross-entropy loss of the final iteration by varying the iteration parameter N.

Since there may be random dislocation between the deblurred image and the corresponding sharp image, we adopt the same procedure as described in [27] to align the deblurred images with the sharp images and then cut off the boundary pixels. Before evaluating the results on the test set, the same alignment operation is performed on the edge detection results, ensuring they are accurately aligned with the ground truth edges.

4.2. Ablation Study

Our ablation study focuses on the performance gain brought by the introduction of intermediate edge detection results in the iterative optimization process. We consider the following three cases for comparison: the original deblurring network structure without edge information, incorporating edge information only in the final output and incorporating richer intermediate edge information during the iterative process(ours). We keep the training settings consistent, and the same Canny edge detection algorithm is applied to obtain the edge detection results for all the deblurred results. The edge detection performance on the test set of our proposed synthetic dataset is shown in Table 1, and the F-measure at both Optimal Dataset Scale (ODS) and Optimal Image Scale (OIS) are recorded for evaluation. The F-measure is a widely used metric in edge detection evaluation, and it balances the precision and recall of detected edges. ODS refers to computing the F-measure by selecting the best threshold for each individual image in the dataset, while OIS calculates the F-measure by choosing the optimal threshold globally across all images. These metrics are crucial for assessing the performance of our proposed approach for edge detection in motion-blurred images.

Table 1. Edge detection performance of ablation study on synthetic dataset.

Method	ODS	OIS
w/o edge info	0.558	0.585
with output edge info	0.566	0.593
ours	**0.569**	**0.596**

It can be seen from Table 1 that our method achieves the highest ODS and OIS scores while the original deblurring network structure without edge information performs the worst, which indicates that incorporating richer edge information during the deblurring process can indeed improve the edge detection performance.

See Figure 5 for the comparison of the deblurred and edge detection results for these three methods. Figure 5a shows a motion-blurred image, while Figure 5e displays its edge detection result. Figure 5b–d depict the deblurred results of these three methods: the network without edge information, the network with output edge information, and our proposed network with richer intermediate edge information, and Figure 5f–h represent the edge detection results obtained after deblurring the images using these three methods. Due to the incorporation of richer edge information during training, our proposed method enhances the contrast between objects and the background when deblurring motion-blurred images. Although this may increase the gap between the deblurred result and the sharp image, it improves edge detection performance and produces clearer and more stable edges.

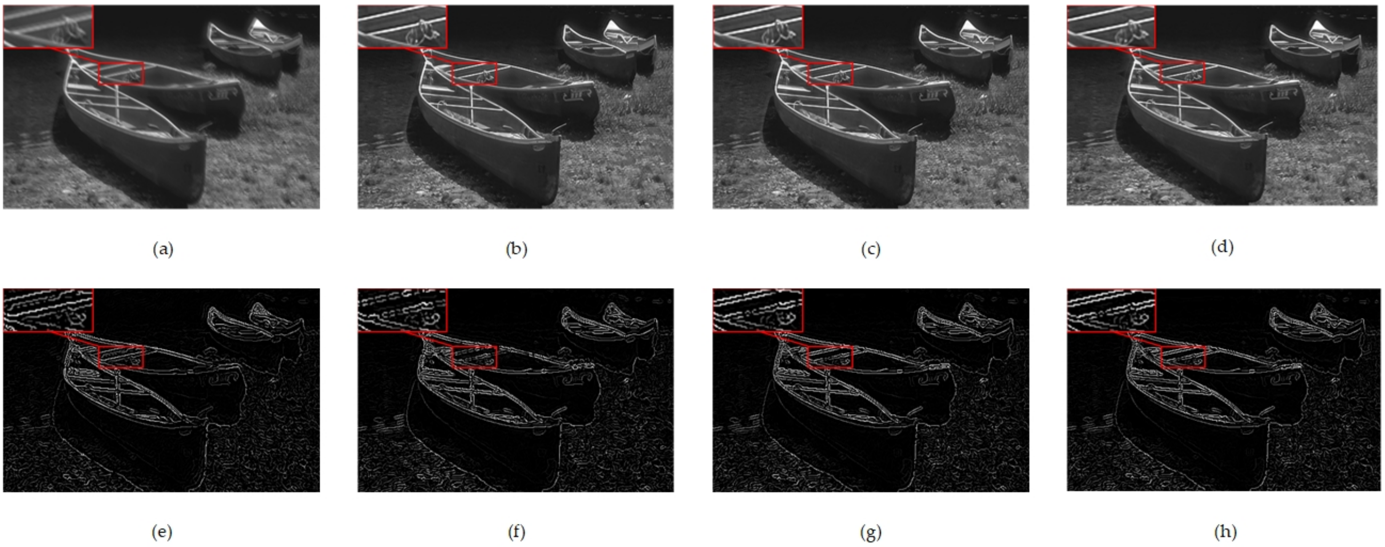

Figure 5. Visual inspection of ablation study. (**a**,**e**) Motion-blurred image and its edge detection result; (**b**,**f**) Deblurred result of the network without edge information and its edge detection result; (**c**,**g**) Deblurred result of the network with output edge information and its edge detection result; (**d**,**h**) Deblurred result of our network with intermediate edge information and its edge detection result.

4.3. Performance Evaluation and Comparison

The proposed method is compared with representative image deblurring methods. The competing methods include traditional single image-based deblurring method hyper-Laplacian (HL) [28] and deep learning-based deblurring methods DeblurGAN-v2 [29], FDN [30] and IRCNN [31], and the same Canny edge detection algorithm is applied to obtain the edge detection results for all the deblurred results. DeblurGAN-v2 [29] is a blind deblurring method, while FDN [30] and IRCNN [31] are non-blind deblurring method swhere the blur kernels are estimated using the method described in Section 3.1, and all these deep learning-based methods have been retrained on our synthetic dataset to establish fair play.

Figure 6 and Table 2 show the edge detection results of all the methods. It can be seen that the ODS and OIS scores of our proposed method is modestly better than FDN [30] and IRCNN [31] and surpasses HL [28] and DeblurGAN-v2 [29] by a significant margin. This indicates that our method outperforms these existing methods in terms of edge detection performance for motion-blurred images.

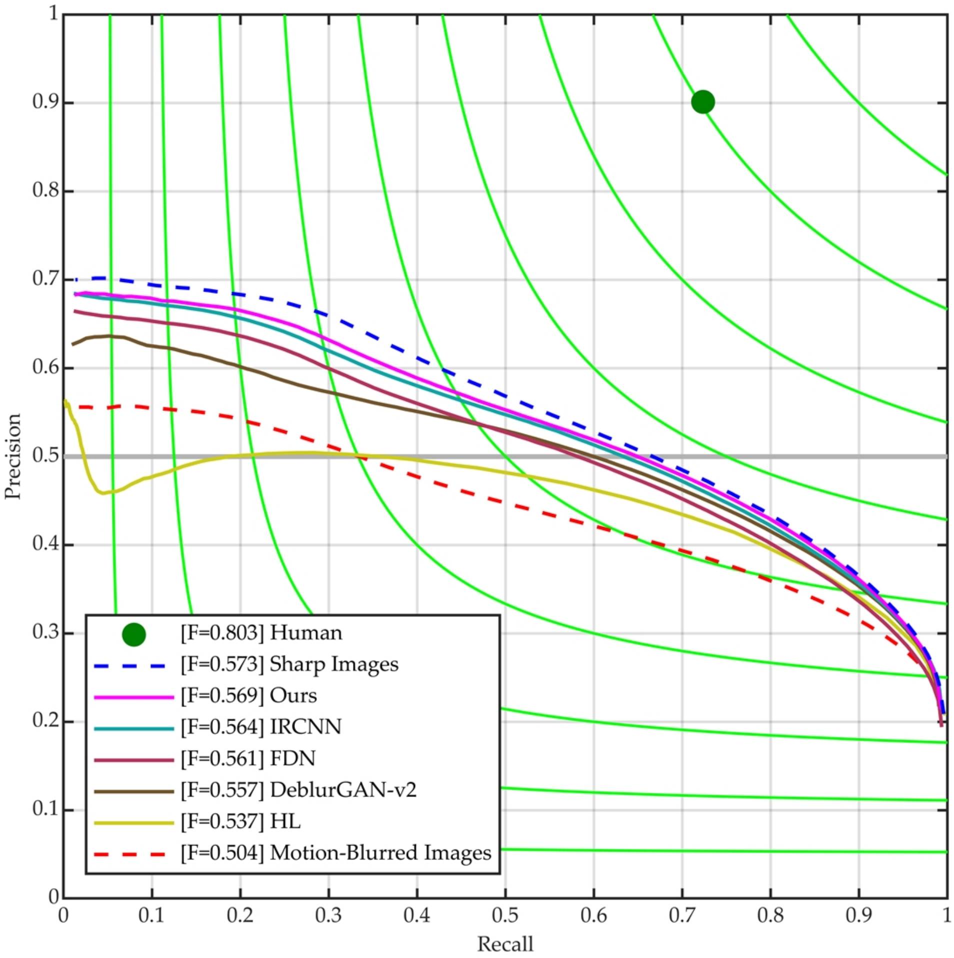

Figure 6. Precision-Recall curves of our method and some competitors on synthetic dataset.

Table 2. Edge detection performance of our method and some competitors on synthetic dataset.

Method	ODS	OIS
Sharp Images	0.573	0.605
Motion-Blurred Images	0.504	0.540
HL [28]	0.537	0.576
DeblurGAN-v2 [29]	0.557	0.583
FDN [30]	0.561	0.581
IRCNN [31]	0.564	0.590
Ours	**0.569**	**0.596**

To test the robustness of the proposed method against sensor data errors, we amplified the sensor data error levels in the synthetic dataset by a factor of 2 and 3 (the exposure time was also scaled by a factor of 2 and 3 correspondingly) to generate new test images for evaluating the edge detection performance of the above methods. The comparison results with other methods are shown in Figure 7. It can be clearly seen that the accuracy of our proposed method decreases less as the error level increases, which indicates the robustness of our method in dealing with sensor data errors.

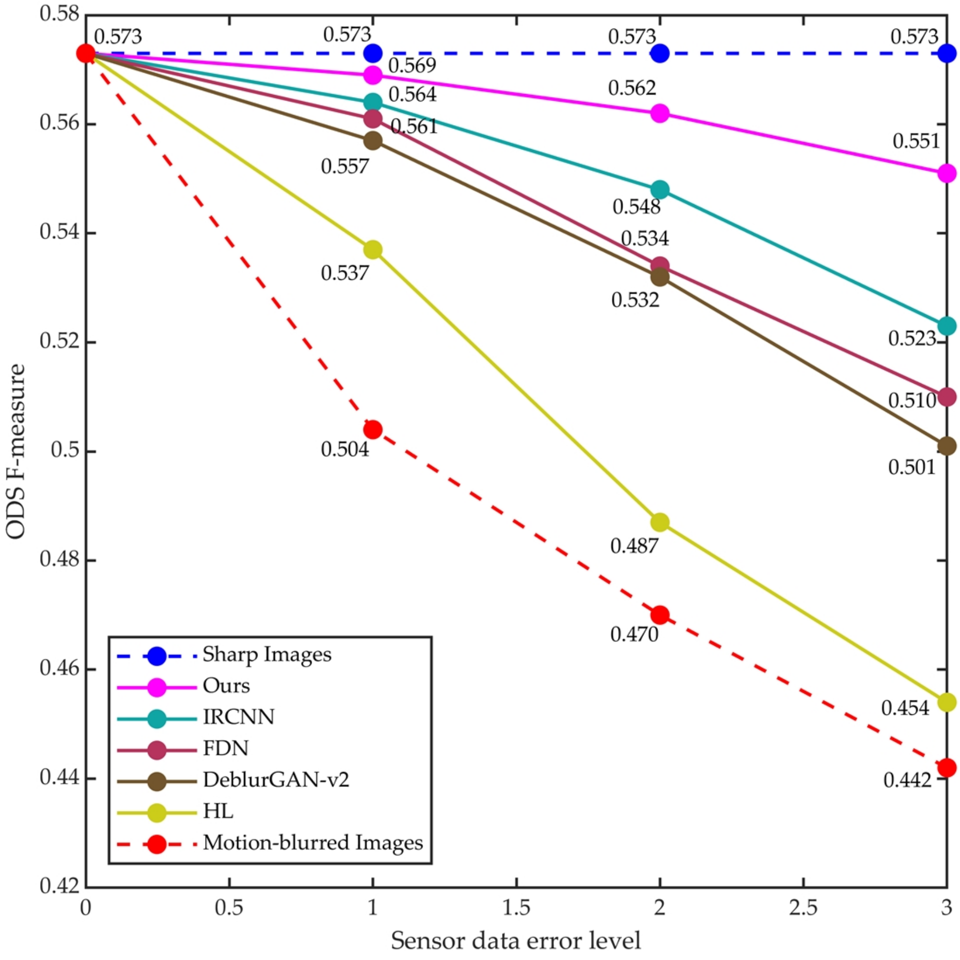

Figure 7. Robustness test results of our method and other methods on synthetic dataset.

5. Conclusions

In this paper, we propose an approach that ensures the efficacy of the deblurring process while coupling it with the edge detection task, thereby achieving higher edge detection precision. We utilize inertial sensors to aid in the deblurring process and address the impact of sensor data errors through a NN-based iterative optimization scheme. During the iterative process, we incorporate rich edge information to adapt the network's optimization objective to edge detection tasks. Experimental results show that our proposed method achieves higher accuracy and robustness on a synthetic dataset, demonstrating the effectiveness of our method for edge detection of motion-blurred images.

In our future work, we are committed to advancing our research by constructing an image acquisition platform that incorporates inertial sensor data. By capturing real-world motion-blurred image data along with inertial sensor data during exposure time, we will compare the results of deblurring and edge detection with other methods in terms of size measurement accuracy. This evaluation will allow us to better validate and showcase the efficacy of our algorithm in practical scenarios and enable comprehensive comparisons with currently advanced deblurring methods.

Author Contributions: Conceptualization, L.T. and P.W.; methodology, L.T.; software, L.T.; validation, L.T., K.Q. and Y.Z.; formal analysis, L.T., K.Q. and Y.Z.; investigation, L.T.; resources, L.T.; data curation, L.T.; writing—original draft preparation, L.T.; writing—review and editing, P.W.; visualization, L.T.; supervision, P.W.; project administration, L.T.; funding acquisition, P.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Aquino, A.; Gegundez-Arias, M.E.; Marin, D. Detecting the optic disc boundary in digital fundus images using morphological, edge detection, and feature extraction techniques. *IEEE Trans. Med. Imaging* **2010**, *29*, 1860–1869. [PubMed]
2. Maninis, K.K.; Pont-Tuset, J.; Arbelaez, P.; Van Gool, L. Convolutional Oriented Boundaries: From Image Segmentation to High-Level Tasks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *40*, 819–833. [PubMed]
3. Rasche, C. Rapid contour detection for image classification. *IET Image Process.* **2018**, *12*, 532–538. [CrossRef]
4. Zitnick, C.L.; Dollár, P. Edge boxes: Locating object proposals from edges. In Proceedings of the Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, 6–12 September 2014; Springer: Berlin/Heidelberg, Germany, 2014; Volume 8693, pp. 391–405.
5. Fergus, R.; Singh, B.; Hertzmann, A.; Roweis, S.T.; Freeman, W.T. Removing camera shake from a single photograph. In *Acm Siggraph 2006 Papers*; ACM Digital Library: New York, NY, USA, 2006; pp. 787–794.
6. Shan, Q.; Jia, J.; Agarwala, A. High-quality motion deblurring from a single image. *ACM Trans. Graph.* **2008**, *27*, 1–10.
7. Xu, L.; Jia, J. Two-phase kernel estimation for robust motion deblurring. In Proceedings of the Computer Vision–ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Greece, 5–11 September 2010; Springer: Berlin/Heidelberg, Germany, 2010; Volume 6311, pp. 157–170.
8. Whyte, O.; Sivic, J.; Zisserman, A. Deblurring shaken and partially saturated images. *Int. J. Comput. Vis.* **2014**, *110*, 185–201.
9. Joshi, N.; Kang, S.B.; Zitnick, C.L.; Szeliski, R. Image deblurring using inertial measurement sensors. *ACM Trans. Graph.* **2010**, *29*, 1–9.
10. Joshi, N.; Zitnick, C.L.; Szeliski, R.; Kriegman, D.J. Image deblurring and denoising using color priors. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; IEEE: Piscataway, NJ, USA, 2009; pp. 1550–1557.
11. Budd, J.; van Gennip, Y.; Latz, J.; Parisotto, S.; Schönlieb, C.-B. Joint reconstruction-segmentation on graphs. *arXiv* **2022**, arXiv:2208.05834 2022.
12. Cai, J.-F.; Ji, H.; Liu, C.; Shen, Z. Blind motion deblurring from a single image using sparse approximation. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; IEEE: Piscataway, NJ, USA, 2009; pp. 104–111.
13. Hu, Z.; Yuan, L.; Lin, S.; Yang, M.-H. Image deblurring using smartphone inertial sensors. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 1855–1864.
14. Hee Park, S.; Levoy, M. Gyro-based multi-image deconvolution for removing handshake blur. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 3366–3373.
15. Šindelář, O.; Šroubek, F. Image deblurring in smartphone devices using built-in inertial measurement sensors. *J. Electron. Imaging* **2013**, *22*, 011003. [CrossRef]
16. Zhang, Y.; Hirakawa, K. Combining inertial measurements with blind image deblurring using distance transform. *IEEE Trans. Comput. Imaging* **2016**, *2*, 281–293.
17. Mustaniemi, J.; Kannala, J.; Särkkä, S.; Matas, J.; Heikkila, J. Gyroscope-aided motion deblurring with deep networks. In Proceedings of the 2019 IEEE Winter Conference on Applications of Computer Vision (WACV), Waikoloa, HI, USA, 7–11 January 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 1914–1922.
18. Nan, Y.; Ji, H. Deep learning for handling kernel/model uncertainty in image deconvolution. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 2388–2397.
19. Adler, J.; Lunz, S.; Verdier, O.; Schönlieb, C.-B.; Öktem, O. Task adapted reconstruction for inverse problems. *Inverse Probl.* **2022**, *38*, 075006. [CrossRef]
20. Yang, C.; Wang, W.; Feng, X. Joint image restoration and edge detection in cooperative game formulation. *Signal Process.* **2022**, *191*, 108363. [CrossRef]
21. Hartley, R.; Zisserman, A. *Multiple View Geometry in Computer Vision*; Cambridge University Press: Cambridge, UK, 2003.

22. Zhang, K.; Zuo, W.; Chen, Y.; Meng, D.; Zhang, L. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE Trans. Image Process.* **2017**, *26*, 3142–3155. [CrossRef] [PubMed]
23. Canny, J. A computational approach to edge detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **1986**, *6*, 679–698.
24. Arbelaez, P.; Maire, M.; Fowlkes, C.; Malik, J. Contour detection and hierarchical image segmentation. *IEEE Trans. Pattern Anal.* **2010**, *33*, 898–916. [CrossRef] [PubMed]
25. Zhang, S.; Zhen, A.; Stevenson, R.L. A Dataset for Deep Image Deblurring Aided by Inertial Sensor Data. *Electron. Imaging* **2020**, *32*, 379-1–379-6. [CrossRef]
26. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980 2014.
27. Vasu, S.; Maligireddy, V.R.; Rajagopalan, A. Non-blind deblurring: Handling kernel uncertainty with CNNs. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 3272–3281.
28. Krishnan, D.; Fergus, R. Fast image deconvolution using hyper-Laplacian priors. *Adv. Neural Inf. Process. Syst.* **2009**, 1033–1041.
29. Kupyn, O.; Martyniuk, T.; Wu, J.; Wang, Z. Deblurgan-v2: Deblurring (orders-of-magnitude) faster and better. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 8878–8887.
30. Kruse, J.; Rother, C.; Schmidt, U. Learning to push the limits of efficient fft-based image deconvolution. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 4586–4594.
31. Zhang, K.; Zuo, W.; Gu, S.; Zhang, L. Learning deep CNN denoiser prior for image restoration. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 3929–3938.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.