Team 4 - Midterm Project Report

Tadipatri Uday Kiran Reddy
ee19btech11038@iith.ac.in

Sahukari Chaitanya Varun
ee19btech11040@iith.ac.in

L. Pranay Kumar Reddy
ai19btech11019@iith.ac.in

Abstract

With the evolution of high-resolution digital cameras, snapshots have become the most common way to record and share visual data and experiences taken through mobile phones and tablets. But as humans, it is challenging to keep a stiff hold while capturing an image, especially when the image is to be captured on the move; it introduces blur. Though there lot of external tools for frame stabilization, such as gimbal, mounting it on can be challenging at times. Hence, we propose this paper to solve this problem by processing the image with the aid of gyroscope sensed data attached with the camera device, helping to deblur the image. We aim to implement this deblurring with the help of deep convolution nets and extrapolate it with GAN's to give better quality and faster output than traditional CNN's. We can extend this work to stabilise the video feed with lower frame rates with faster computation capabilities.

1. Further literature review

With the ever-growing classical image processing problem, deblurring is also a key research area that has witnessed significant progress. There are major three types of deblurring; Burst mode assisted deblurring, non-blind deblurring and blind deblurring(also known as single image-based deblurring); Burst-mode assisted deblurring, also known as deblurring using multiple images, as it describes series of images are captured for a scene and using these the deblurred images are estimated recent works like [?], [?] focused on statistics approach to estimate blur-kernels or Point Spread Function (PSF) and applying deconvolution on the image to get the latent image. Authors of [?] use the Convolutional Neural Network(CNN) approach to deblur the image; here, the PSF approach is not adopted. In non-blind deblurring, blur-kernel or PSF available priorly these works [?], [?] render perfect images but this method is not practically feasible as the PSF is not priorly known always. In blind deblurring, alone single image is exploited to deblur the image, classically blur-kernel is estimated later a deconvolution operation is applied on the image to get the latent image. Authors of [?], [?] use a statistical approach to estimate the PSF, albeit they assume that blur is space-invariant. In simpler words, they assume the blur is linear throughout the image, albeit images experience non-linear blur. Intuitively, we see that estimating the blur kernel followed by deconvolution is not flawless and adds artefacts that may render an undesirable latent image. Smartphones or DSLR cameras are equipped with sensors like gyroscopes, accelerometers and magnetometers. Recent works [?], [?], [?], [?], [?] exploit such data to deblur the images but works [?], [?], [?], [?] take an impractical assumption that blue is space-invariant and also assume that sensor data is reliable. But these are prone to time delay, miscalibration and anomalies. Authors in [?] showed incredible results despite these assumptions; here, they built their architecture based on Encoder-Decoder CNN with a pre-computed 2-channel blur field. In [?], the authors have considered the sensor's non-idealities. The authors also compute the blur-field with sensor data but also refine it by solving an optimisation problem.

2. Deep Gyro [?] Implementation

This section covers on the implementation of deblurring of images using gyroscope-extracted data with help a deep neural network. This deblurring is based fully-convolutional neural network. For working on this problem in a supervised manner we need data.

2.1. Data for Playing

For training the network, we need a set of blurred and sharp images along with the corresponding gyro-based blur fields. But manually capturing this data data in real time is difficult. So we use gyroscope readings to generate realistic blur fields and blurred images. For this purpose, the visual-inertial dataset has been used which is a set diverse sequences in different scenes for evaluating odometry. IMU measurements in the dataset has the measurements of accelerations and angular velocities on 3 axes at 200 Hz. We extract the gyro data for generating the blur fields. Here, two different blur fields are generated referred to as ”exact” and ”noisy” , where the exact blur field is used in generating the blurred image and the noisy blur field is given to network as additional input to help in deblurring.

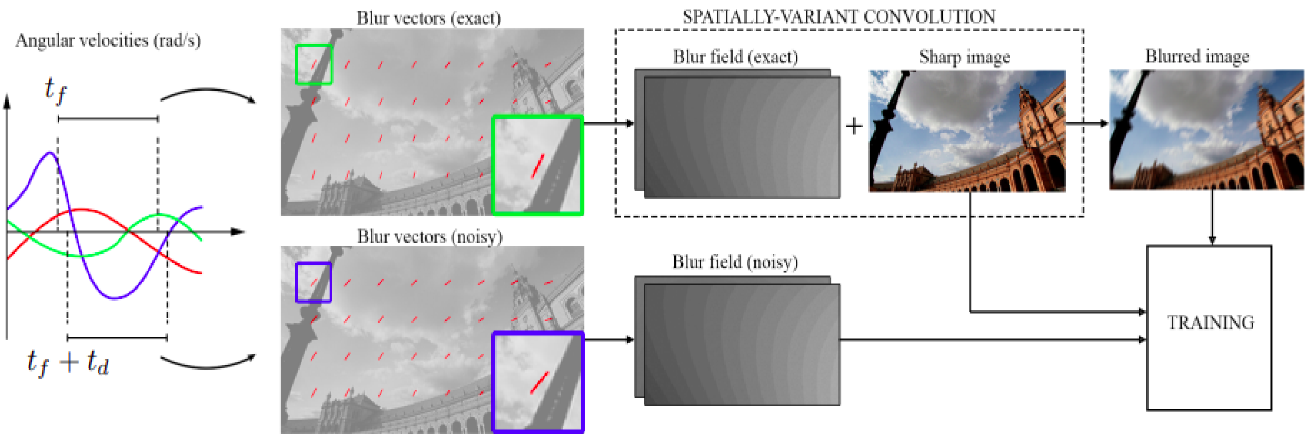


Figure 1. Data generation schematic picture

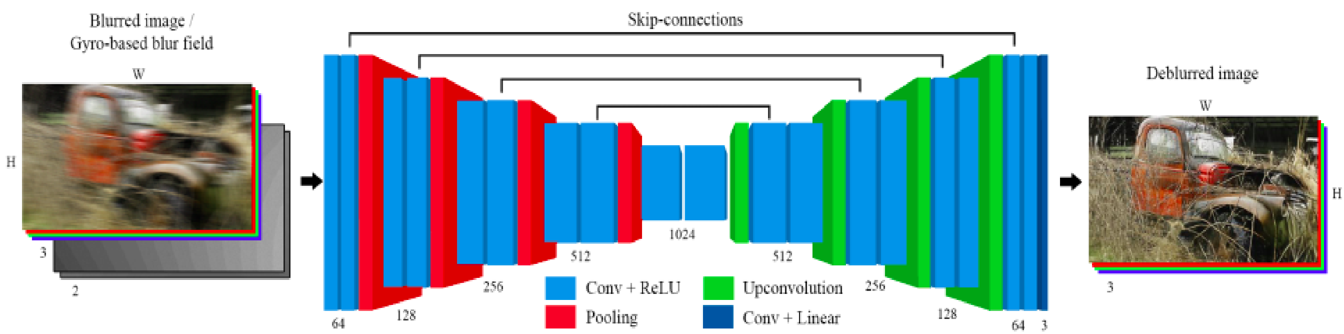


Figure 2. Network Architecture

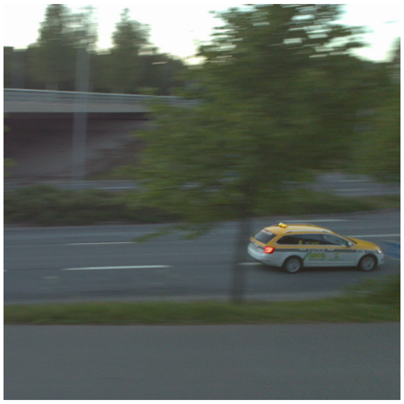
The calibration parameters of the camera device which include camera intrinsics, camera readout time, IMU-camera temporal offsets, IMU-to-camera rotation is taken into consideration so that if we downsample the picture, the camera intrinsics also have to be scaled accordingly and help in computation of blur field. Along with them, the time stamps and exposure times data is also used for computing the blur field. For the purpose of further training of our model, we try to inculcate the DAVIS 240C dataset for pose estimation, visual odometry and SLAM.

2.2. Generating Blur Fields

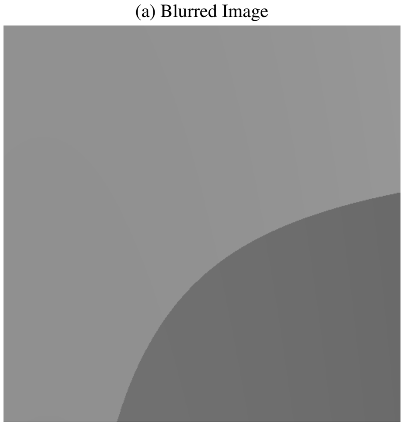
As described in PPR, the same approach is taken for the blur estimation. To make the model more robust, the exposure times and read out times are slightly but randomly tweaked to simulate the misalignment between the camera and the gyroscope. A small uniformly random delay is added when computing the noisy blur field. We first rotate the gyroscope measurements from IMU frame to the camera frame. The gyroscope reading are then upsampled to facilitating readings for the time difference between two consecutive row exposures. The rotation of camera is estimated by integrating the gyroscope readings and futher fid the quaternions to rotation matrices. Then the blur fields are computed based on the horizontal and vertical components of the blur by finding the projections as discussed in the PPR. These blur vectors are returned as a gray scale image for the neural network.

2.3. Network Architecture

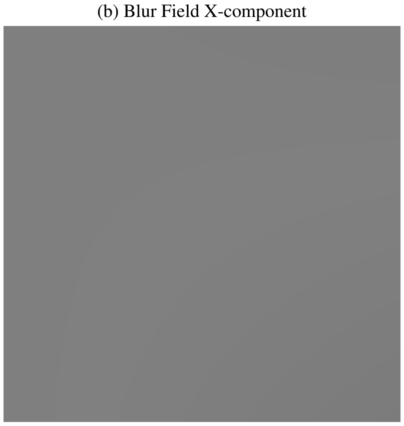
This network architecture is similar to that of encoder-decoder network which is generally used in image translation problems. The input of the network consists of blurred RGB image and a gyro-based blur field. They pass through series of convolutional and downsampling layers, until the lowest resolution is reached and then after the bottlenecke, the low resolution image is expanded to full resolution image with the help of upsampling layers and skip connections, which helps in information exchange between the encoder and decoder network. As shown the network has 5 convolution layers for downsampling and 5 for upsampling. The output of each convolution layer is padded accordingly.



(a) Blurred Image



(b) Blur Field X-component



(c) Blur Field Y-component

(d) Example for generation of blur field

The stride is 2 in 2× 2 max pooling in downsampling and the upconcolution layers have 2×2 convolution that halves the number of feature channels.

2.4. Training and Results

The model uses Adam optimizer for fitting the data. ReLu-activation used for all layers except the last layer which has linear activation. The original model was trained on 100k images. So we use the weights of the pre-trained model to test for the initial phase. The results are shown in

Figure 4. Results replicated from the paper(blurred on left, deblurred on right)

Figure 4

In Figure 5, few snapshots of real world are taken and deblurred.

3. MCG-GAN Network

3.1. Dataset for Model

We used CIFAR-10 dataset for training our model. The CIFAR-10 dataset consists of 60000 32x32 colour images in 10 classes, with 6000 images per class. For testing the model, we picked 10 random images from each class of the dataset. So, the size of our training input is 100 32x32x3 images. The size of training input is small as the deep convolutional GAN takes lot of time to run for higher dataset sizes.

Figure 5. Simulated Examples(blurred on left, deblurred on right)

3.2. Network Architecture

Our GAN[1] includes 1 generator and 1 discriminator. The generator takes an input of dimension noise-dim(noises to make image look blurry), and through multiple sets of dense layers followed by LeakyReLu activations, it outputs a flat version of the image(unblurred version).
We added noise to normal images instead of taking blurred images we created for now to test the validity of model. As the model is performing well now, the generator of final DCG-GAN will take blurred image and motion blur data and work to return an image which can fool the discriminator to believe it's a real image. Which in our case means, creating a deblurred image from given blurred image.
This model starts with a small image, which is slowly scaled up with Conv2DTranspose layers. Growing GAN images is an incredible technique developed by NVIDIA, and the generator architecture above mimics a greatly simplified version of it. For the final layer we are using tanh activation function as it gives better results and normalizes our image. The optimizer we use is also very important, in our model we decided to use Adam optimizer as Adam combines the best properties of the AdaGrad and RMSProp algorithms to provide an optimization algorithm that can handle sparse gradients on noisy problems.

In Figure **??** we have 100 images after adding noises. Which in our case means before training and after training these images for 320 epochs we got Figure **??**.

References

[1] Miika Aittala and Fredo Durand. Burst image deblurring using permutation invariant convolutional neural networks. In *The European Conference on Computer Vision (ECCV)*, September 2018.

[2] Jian-Feng Cai, Hui Ji, Chaoqiang Liu, and Zuowei Shen. Blind motion deblurring using multiple images. *Journal of Computational Physics*, 228:5057–5071, 08 2009.

[3] Rob Fergus, Barun Singh, Aaron Hertzmann, Sam T. Roweis, and William T. Freeman. Removing camera shake from a single photograph. *ACM Trans. Graph.*, 25(3):787–794, July 2006.

[4] Michael Hirsch, Christian J. Schuler, Stefan Harmeling, and Bernhard Schölkopf. Fast removal of non-uniform camera shake. In *2011 International Conference on Computer Vision*, pages 463–470, 2011.

[5] Zhe Hu, Lu Yuan, Stephen Lin, and Ming-Hsuan Yang. Image deblurring using smartphone inertial sensors. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1855–1864, 2016.

[6] Seowon Ji, Jun-Pyo Hong, Jeongmin Lee, Seung-Jin Baek, and Sung-Jea Ko. Robust single image deblurring using gyroscope sensor. *IEEE Access*, 9:80835–80846, 2021.

[7] Janne Mustaniemi, Juho Kannala, Simo Särkkä, Jiri Matas, and Janne Heikkilä. Gyroscope-aided motion deblurring with deep networks, 2018.

[8] T M Nimisha, Akash Kumar Singh, and A N Rajagopalan. Blur-invariant deep learning for blind-deblurring. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 4762–4770, 2017.

[9] Uwe Schmidt, Carsten Rother, Sebastian Nowozin, Jeremy Jancsary, and Stefan Roth. Discriminative non-blind deblurring. In *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pages 604–611, 2013.

[10] Ondrej Sindelar and Filip roubek. Image deblurring in smartphone devices using built-in inertial measurement sensors. *Journal of Electronic Imaging*, 22, 2013.

[11] Subeesh Vasu, Venkatesh Reddy Maligireddy, and A. N. Rajagopalan. Non-blind deblurring: Handling kernel uncertainty with cnns. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3272–3281, 2018.

[12] Qing Wang, Jun Tan, Tianzhang Xing, Feng Chen, and Jinping Niu. Sid: Sensor-assisted image deblurring system for mobile devices. *IEEE Access*, 7:146607–146619, 2019.

[1] Drive folder link containing training model