Single Image Deblurring for Underwater & Surface Scenes

A Project Report

submitted by

SUNIL KUMAR

in partial fulfilment of the requirements
for the award of the degree of

MASTER OF TECHNOLOGY

DEPARTMENT OF ELECTRICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY MADRAS.

May 2018

THESIS CERTIFICATE

This is to certify that the thesis titled *'Single Image Deblurring for Underwater & Surface Scenes '*, submitted by **Sunil Kumar**, to the Indian Institute of Technology, Madras, for the award of the degree of **Master of Technology**, is a bona fide record of the research work done by him under my supervision. The contents of this thesis, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

A.N. Rajagopalan
Research Guide
Professor
Dept. of Electrical Engineering
IIT-Madras, 600 036

Place: Chennai

Date: May 11, 2018

ACKNOWLEDGEMENTS

I express my sincere gratitude to my supervisor, Dr. A.N.Rajagopalan for introducing me to the exciting and challenging world of image processing and for his valuable discussions and motivation throughout the project. I am grateful to him for providing his valuable time to guide me during the project.

I am also thankful to my lab mates, especially Nimisha, Kuldeep, Subeesh and Anshul for their help and guidance, whenever I got stuck and for keeping the atmosphere of lab highly friendly and research oriented.

Finally, I thank to my parents, my brother Anil and friends Bijoy, Yashodhara, Vishal and Hemant for bringing moments of fun during stressful times.

ABSTRACT

KEYWORDS: Underwater Imaging; Skew; Water Surface Waveform; Image Restoration; Deblurring; Reblurring; Unsupervised Learning; Generative Adversarial Networks.

This thesis mainly comprises works on two problems, first one involving underwater imaging and second related to unsupervised motion deblurring. First one deals with reconstruction of dynamic water surface using single captured blurred image of a planner scene immerged inside water. Waveform estimation is done by using blur as cue and relating it with the slope of water surface waveform.
The above problem statement is extended to incorporate the deblurring of captured blurred image using the information of water surface waveform. From this, the point spread function estimation is done. Conventional methods assume some priors on psf and latent image and use optimization techniques to compute the psf, however, a better estimate can be easily obtained, if the information of shape of water surface is added over it.

The second problem deals with class specific motion deblurring. In this work, three classes of images, namely, checkerboard, faces and texts deblurring is handled. For this purpose, GAN are trained to generate clean images and inorder to improve the stability of GAN and to preserve the image correspondence, an additional CNN module that reblurs the generated GAN output to match with the blurred input and a module is added, which matches the gradient of GAN output with the input. This self guidance is achieved by imposing a scale-space gradient error with an additional gradient module. This makes the whole architecture completely unsupervised.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS i

ABSTRACT iii

LIST OF TABLES vii

LIST OF FIGURES ix

ABBREVIATIONS xi

NOTATION xiii

1 INTRODUCTION 1

1.1 Underwater Imaging 1

1.1.1 Summary of Contribution 1

1.2 Unsupervised Deblurring 2

1.2.1 Related Works 2

1.2.2 Summary of Contribution 4

2 Water Surface Waveform Estimation 5

2.1 Image Formation in flowing Water 5

2.2 Blur and Skew In Underwater Imaging 7

2.3 PSF 9

2.3.1 Length of PSF 9

2.3.2 Weights of PSF 11

2.4 Water Surface Waveform Estimation 12

2.5 PSF Estimation 14

2.5.1 PSF model for sinusoids 15

2.5.2 PSF estimation from the Blurred Image 17

2.6 Conclusion 20

3 Unsupervised Deblurring **21**

3.1 Loss Functions 23

3.1.1 Tried out loss functions 23

3.1.2 Final loss function 25

3.2 Network Architecture 27

3.2.1 Tried Network Architecture 27

3.2.2 Final Network Architecture 27

3.3 Experiments 29

3.3.1 Dataset Creation 29

3.3.2 Comparison Methods 30

3.3.3 Quantitative Analysis 30

3.3.4 Visual Comparisons 32

3.4 Conclusions 34

REFERENCES **39**

LIST OF TABLES

3.1 The proposed generator and discriminator network architecture. conv ↓ indicates convolution with stride 2 which in effect reduces the output dimension by half and d/o refers to dropout. 28

3.2 Reblurring CNN module architecture. 28

3.3 Quantitative comparisons on face, text, and checkerboard datasets. . 32

3.4 Quantitative comparisons on face and text on real handshake motion (48). 32

LIST OF FIGURES

1.1 Proposed network with GAN, reblur module and scale-space gradient module. 4

2.1 Ray diagram for a scaled orthographic camera to demonstrate image formation. 6

2.2 Variation of gradients with time for a periodic water surface, Time (a) $T = 0$, (b) $T = t_1$, (c) $T = t_2$ & (d) $T = t_4$. 8

2.3 Increase in the shift of pixel because of higher slope. 10

2.4 Increase in the shift of pixel because of increase in distance between surface and object/camera. 10

2.5 Slope and weight correspondence of Psf. 11

2.6 Input blurred image. 14

2.7 The estimated waveform. 15

2.8 The sine wave. 15

2.9 Input blurred image. 17

2.10 The clean image. 18

2.11 Deblurred image using traditional approach. 19

2.12 Deblurred image using new approach. 19

3.1 Scale space gradient error. 23

3.2 SML(weight 0.1) with GAN (row 1&2), SML (weight 100) with GAN (row3 &4), SML (weight 0) with GAN (row 5 & 6). 24

3.3 Effect of different cost functions. (a) Input blurred image to the generator, (b) result of unsupervised deblurring with just the GAN cost in eq. (3.3), (c) result obtained by adding the reblurring cost in eq. (3.4) with (b), (d) result obtained with gradient cost in eq. (3.5) with (c), and (e) the target output. 26

3.4 GAN network with SML module 27

3.5 Visual comparison on checkerboard deblurring. Input blurred image, deblurred results from conventional methods (19), (18) and (46), results from supervised network in (10), (37) and unsupervised network (13), Proposed network result and the GT clean image are provided in that order. 33

3.6 Visual comparisons on face deblurring. 34

ABBREVIATIONS

UW	Underwater
GAN	General Adversarial Networks
cGAN	Conditional General Adversarial Networks
CNN	Convolutional Neural Networks
AGD	Alternating gradient descent
ReLU	Rectified linear units
fc	Fully connected
GPU	Graphic Processing Unit
VAE	Variational Auto Encoder
SML	Sum Modified Laplacian

NOTATION

$\mathbf{P}$	Camera Projection Matrix
$p_{x,\tau}$	Displacement due to refraction
η	Refractive index of water
θ_r	Angle of refraction
θ_i	Angle of incidence
$s = [X \quad Y \quad Z]^T$	3D point in real world
$s = [x \quad y]^T$	Pixel location in image domain
$\mathbf{f}$	Clean image
$\mathbf{g}$	Blur image
G	Generator network
D	Discriminator network
X	Clean image domain
Y	Blur image domain
L_G	Generator loss
$\gamma_{\mathbf{adv}}$	Weight of adversarial loss
$L_{\mathbf{adv}}$	Adversarial loss
$\gamma_{\mathbf{reblur}}$	Weight of reblurring loss
$L_{\mathbf{reblur}}$	Reblurring loss
$\gamma_{\mathbf{grad}}$	Weight of gradient loss
$L_{\mathbf{grad}}$	Gradient loss

CHAPTER 1

INTRODUCTION

1.1 Underwater Imaging

Underwater imaging has been explored intensively in last few decades(1). The underwater images suffer from various types of degradations like color loss, noise, low contrast, skewing and blurring (2) and in worse case all in single image. This posts a challenge in the underwater imaging applications of coral reef monitoring, contamination of shallow water, mapping the distribution of vegetation & seabed sediments etc (3).

In this work, only blurring problem is addressed and an attempt is made to extract more cues about dynamics of water waveform to better estimate the blur kernel which would eventually improve the deblur methodologies. The images underwater are primarily affected by skew and motion blur because of dynamic refractive medium and as the wave moves, the light rays experience different amount of refraction. There are several works that aim to restore skewed scenes (4; 5; 6; 7; 8).

These methods neglects the effect of motion blur and only skew is taken in consideration. However, skewing of scenes can also be associated with motion blur when the imaging medium is dynamic in nature and camera exposure is long. If the exposure time is long, all the pixels would see the same waveform (experience same nature of skew) and the result image will be a motion blurred image. If intermediate frames are analyzed, all intermediate frames would be skewed version of clean image and they all averaged to give the final blurred image.

From that blurred image, psf of blur is estimated and used to further estimate the water surface waveform.

1.1.1 Summary of Contribution

- Detailed analysis on psf.

- Water surface waveform estimation from psf.
- Given water surface waveform refinement of estimated psf and deblurring using the refined psf.

1.2 Unsupervised Deblurring

Blind-image deblurring is a classical image restoration problem which has been an active area of research in image and vision community over the past few decades. With increasing use of hand-held imaging devices, especially mobile phones, motion blur has become a major problem to confront with. Any motion in the camera or scene while capturing the image leads to motion blur. In most cases, users only have single blurred image.

Blind-deblurring can be posed as an image-to-image translation from blur domain y to clean domain x and with proper correspondance between the images. Many recent deep learning based deblurring networks (10), estimate this mapping when provided with large sets of y_i, x_i paired training data. Even though these networks have shown promising results, the basic assumption of availability of paired data is too demanding. In many a situation, collecting paired training data can be difficult, time-consuming and expensive and in some applications even non-existent.

This limitation demands unsupervised learning (13; 14; 15) of deep networks.

1.2.1 Related Works

There is a vast literature on motion deblurring spanning both conventional and supervised deep learning techniques. Similarly, of late there are works on unsupervised image translations gaining popularity due to lack of availability of paired data.

Motion deblurring is a long-studied topic in imaging community. To avoid shot noise due to low amount of available photons in low light scenarios, the exposure time is increased. Hence, even a small camera motion is enough to create motion blur in the recorded image due to averaging of light energy from slightly different versions of the same scene. The above problem is simplified in several deblurring works (16; 17) by assuming usage of multiple frames.

Deblurring from single blur frame is actually an ill-posed problem. To overcome the

ill-posedness, most of the existing algorithms (18; 19; 20) rely on image heuristics and assumptions on the sources of the blur. The most widely used image heuristics are sparsity prior, the unnatural l_0 prior (19) and dark channel prior (18). Assumptions on camera motion are imposed in the form of kernel sparsity and smoothness of trajectory. These heuristics are used as priors and iterative optimization schemes are deployed to solve for camera motion and latent clean frame from a single-blurred input. Even though these methods are devoid of any requirement of paired data, they are highly dependent on the optimization techniques and prior selection.

There have been several attempts to solve this problem using deep learning frameworks. (10; 11; 12) have shown the effectiness of deep networks for the task of blind deblurring. These methods work end-to-end and skip the need for the camera motion estimation and directly provide the clean frame when fed with the blurred image thus overcoming the tedious task of prior selection and parameter tuning. But the main disadvantage with existing deep-learning works is that they require close supervision warranting large amounts of paired (blur, clean) datasets for training.

Unsupervised learning The recent trend in deep learning is to use unpaired data to achieve domain transfer. With the seminal work of Goodfellow (21), GANs have been used in multiple areas of image-to-image translations. The key to this success is the idea of an adversarial loss that forces the generated images to be indistinguishable from real images thus learning the data domain. Conditional GANs (cGAN) (22; 23; 24) have made progress recently for cross-domain image-to-image translation in supervised settings. The goal remains the same in unsupervised settings too i.e; to relate the two domains. One way to approach the problem is by enforcing a common representation across the domains by using shared weights with two GANs as in (15; 25; 26). The fundamental objective here is to use a pair of coupled GANs, one for the source and one for the target domain, whose generators share their high-layer weights and whose discriminators share their low-layer weights. In this manner, they are able to generate invariant representations which can be used for unsupervised domain transfer. Apart from these methods, there are neural style transfer networks (27; 28; 29) that is also used for image-to-image translation with unsupervised data. The idea here is to combine the 'content' features of one image with the 'style' of another image (like famous paintings).

GAN is used in proposed network to learn a strong class-specific prior on clean data.

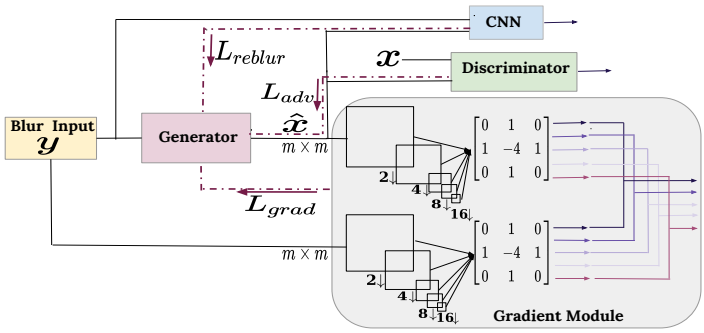

Figure 1.1: Proposed network with GAN, reblur module and scale-space gradient module.

The discriminator thus captures the semantic domain knowledge of a class but fails to capture the content, colors, and structure properly. These are usually corrected with supervised loss functions in regular networks which is not practical in the unsupervised setting. Hence, self-guidance is introduced using the blurred data itself. Proposed network is trained with unpaired data from clean and blurred domains. A comprehensive diagram of the network is provided in Fig 1.1

1.2.2 Summary of Contribution

This is collaborative work done by me and one of my lab mate. For completeness entire work is mentioned in this report. My contributions to this work are as follows:

- This is the first attempt at unsupervised learning for the task of deblurring.
- Tried various cost functions:
  - Implemented Sum modified laplacian (SML) module in the GAN architecture and analyzed its effect without GAN loss.
  - Analyzed the effect of SML module with GAN loss.
  - Analyzed the effect of Gradient module with GAN loss.
- Dataset creation for text, faces and checkerboard deblurring.
- Implemented codes of various networks (SML, SML+GAN,GAN+Gradient) with different loss functions in torch and analyzed effects of hyper-parameters like weights given to loss functions, batch size etc before arriving at the final architecture and parameter setting.

CHAPTER 2

Water Surface Waveform Estimation

2.1 Image Formation in flowing Water

Fig 2.1 represents the ray diagram of image formation. Let $I_g(\mathbf{x})$ be the clean image of the planner scene. Assuming no attenuation in wave, each video frame $I(\mathbf{x},\tau)$ is a distorted version of $I_g(\mathbf{x})$ related by following equations:

$$I(\mathbf{x},\tau) = I(\mathbf{x}+\mathbf{w}(\mathbf{x},\tau)) \tag{1}$$

where $\mathbf{w}(\mathbf{x},\tau)$ is the warping function at each pixel $\mathbf{x}$ and time τ. Let η be the refractive index of medium and θ_i and θ_r be the angle of incidence and refraction, respectively, of the light ray reaching the camera from scene point B. Assuming the water fluctuations with respect to the level h_0 are small. The displacement $p_{x,\tau}$ that a light ray experiences can be calculated as:

$$||p_{x,\tau}||_2 = h_0 \tan(\theta_r - \theta_i) \tag{2}$$

$$||p_{x,\tau}||_2 = h_0[\frac{(\tan\theta_r - \tan\theta_i)}{1+\tan\theta_r \tan\theta_i}] \tag{3}$$

For, small water waves, θ_r and θ_i will be small (9), hence, $\cos\theta_r \approx 1$,$\cos\theta_i \approx 1$ and $1+\tan\theta_r \tan\theta_i \approx 1$. Using these,

$$||p_{x,\tau}||_2 = h_0(\sin\theta_r - \sin\theta_i) \tag{4}$$

$$||p_{x,\tau}||_2 = h_0 \sin\theta_r(1 - \frac{\sin\theta_i}{\sin\theta_r}) \tag{5}$$

From Snell's law $\eta\sin\theta_i = \sin\theta_r$ as $\eta_{air} = 1$. Hence:

$$||p_{x,\tau}||_2 = h_0 \sin\theta_r(1 - \frac{1}{\eta}) \tag{6}$$

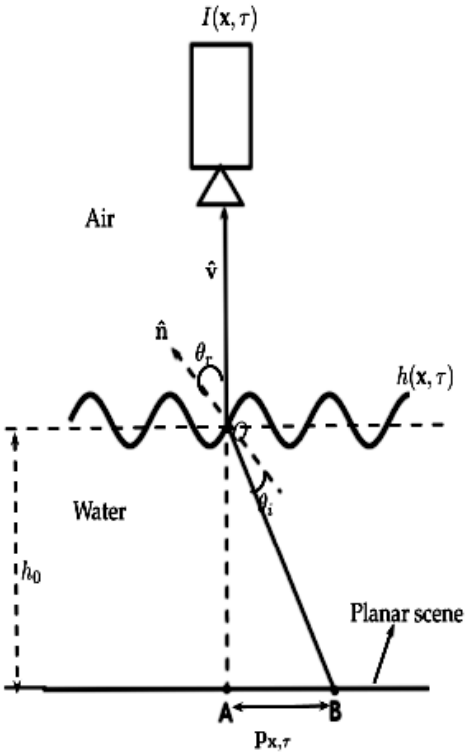

Figure 2.1: Ray diagram for a scaled orthographic camera to demonstrate image formation.

Taking $\alpha = h_0(1-\frac{1}{\eta})$,

$$||p_{x,\tau}||_2 = \alpha\sqrt{1-\cos^2\theta_r} \tag{7}$$

From Fig 2.1, $\cos\theta_r = (\hat{v}.\hat{\eta})$, where $\hat{v} = [0 \quad 0 \quad 1]^T$ and $\hat{\eta}$ can be calculated as

$$\hat{n} = \frac{t_x \times t_y}{||t_x \times t_y||_2} \tag{8}$$

where '.'and '×'represent dot and cross product, while $t_x = [1 \quad 0 \quad h_x]^T$ and $t_y = [0 \quad 1 \quad h_y]^T$ are the tangential vectors defined on water surface at point O, and h_x and h_y denote the partial derivatives of $h(x,\tau)$ with respect to spatial coodinates. If ∇ denotes the gradient of the water surface then $\nabla = [h_x \quad h_y]^T$. Using these values of t_x and t_y in Eq 8:

$$\hat{n} = \frac{[-h_x \quad -h_y \quad 1]^T}{\sqrt{1+h_x^2+h_y^2}} \tag{9}$$

Substituting Eq 9 in Eq 7:

$$||p_{x,\tau}||_2 = \alpha \frac{\sqrt{h_x^2 + h_y^2}}{\sqrt{1 + h_x^2 + h_y^2}} \tag{10}$$

$$||p_{x,\tau}||_2 = \alpha \frac{||\nabla h||_2}{\sqrt{1 + h_x^2 + h_y^2}} \tag{11}$$

For small waves, $||\nabla h||_2 << 1$ and Eq 11 simplifies to

$$||p_{x,\tau}||_2 \approx \alpha ||\nabla h||_2 \tag{12}$$

$$||p_{x,\tau}||_2 \approx \alpha \nabla h(x,\tau) \tag{13}$$

From Fig 2.1, when the water surface is still, the camera will image scene point A. However, in flowing water, the light ray which reaches the camera above the water surface will make a finite angle with the normal $\hat{n}$ defined on the surface of the water at point O. From Snell's law, there will be refraction at the water surface due to which instead of A, the camera will actually see scene point B. The amount of translation that a pixel x experiences on the image plane is given as:

$$w(x,\tau) = \mathbf{P} p_{x,\tau} \tag{14}$$

where **P** is the projection matrix of the camera.

2.2 Blur and Skew In Underwater Imaging

As explained in section 2.1 that due to dynamic nature of water surface, the scene point **s** will transformed to s_τ=**s**+$p_{x,\tau}$, where $p_{x,\tau} = [p_{x,\tau} \quad p_{y,\tau} \quad 0]^T$.
Using the model of pinhole camera, the point $s = [X \quad Y \quad Z]^T$ in 3D world corresponds to pixel **x**, where **x**=$[x \quad y]^T$ image coordinateds. The relation between **x** and **s** is given by $x = \frac{fX}{Z}$ and $y = \frac{fY}{Z}$, here, f is the focal length of camera. Mathematically,

$$\widetilde{x} = \mathbf{K}[\mathbf{I}|\mathbf{0}]\widetilde{s} \tag{15}$$

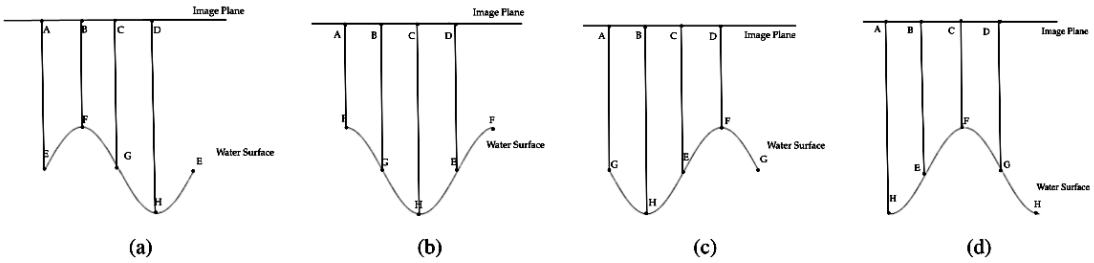

Figure 2.2: Variation of gradients with time for a periodic water surface, Time (a) $T = 0$, (b) $T = t_1$, (c) $T = t_2$ & (d) $T = t_4$.

where

$$\mathbf{K} = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$\mathbf{I}$ is 3×3 identity matrix, $\mathbf{0}$ is 3×1 zero vector, and, $\widetilde{x}$ and $\widetilde{s}$ are homogeneous coordinates of $\mathbf{x}$ and $\mathbf{s}$, respectively. Assuming point $\mathbf{s}$ is on plane with normal $\mathbf{n}$ and at a distance d form the camera image plane. And since, the water surface is dynamic, the normal η will be time varying. And if exposure time is short, an skewed image will be obtained. But if capture time sufficiently large, light rays from multiple scene points will reach same pixel and the averaging effect during the exposure window will lead to motion blurring. The mathematical relationship can be explained using Fig 2.2.

Assuming a sinusoidal wave, at time T=0, A,B,C and D pixels corresponds to points E,F,G and H, respectively. At T=t_1, the pixels A,B,C and D will see surface points F,G,H and E respectively. At T=t_2, these pixels will see G,H,E and F respectively and so on. Therefore, at any time, the pixels have different slopes and hence, different traslations. This leads to skewing effect. The final captured image will be average of all these skewed images.

Let $\mathbf{f}$ and $\mathbf{g}$ be the images captured under still and dynamic water. Then, the image $\mathbf{g}$ is the average of all skewed versions of $\mathbf{f}$ during the exposure time T_e.

Mathematically,

$$g(\mathbf{x}) = \frac{1}{T_e} \int_0^{T_e} f(\mathbf{x} + \mathbf{w}(\mathbf{x}, \tau)) d\tau \tag{16}$$

2.3 PSF

The point spread function (PSF) describes the response of an imaging system to a point source. PSF can be seen as a system's impulse response, the PSF being the impulse response of a focused optical sytem.

In the case of a planner scene immerged inside water or say inside a water tank under dynamic condition, the psf refers to the blur kernel of the image. The point spread function contains the information about the shift of central pixel because of variation in slopes of the water surface.

The PSF has two parameters namely, length of PSF (2k+1) and weights (w_i, i=-k,-k+1,..,-1,0,1,..,k-1,k). Following is a typical 1D psf of length 11:

w_5	w_4	w_3	w_2	w_1	w_0	w_{-1}	w_{-2}	w_{-3}	w_{-4}	w_{-5}

A detailed analysis of information that can be extracted from these paratmeters about the water wavefrom is presented in following sections:

2.3.1 Length of PSF

The length of psf depends upon the maximum slope in the water surface waveform and distances between the water surface and camera and between the water surface and the object. If these distances are kept constant, then the length of psf is entirely function of shape of water surface of waveform.

Effect of slope variation:

Simple geometry can give relation between angle of incidence and angle of plane: $\theta_i = \theta$. From Snell's law:$\sin\theta_r = \frac{\sin\theta_i}{\eta}$

Since, sin is increasing function in $[0, \frac{\pi}{2}]$, so, angle of refraction θ_r will be more for higher slope. This idea is clear from the Fig 2.3 also.

Effect of changing the distances between water surface and camera/object:

From the Fig 2.4, it is clear that with increase in the distance between the water surface and camera/object, the spread of psf increases.

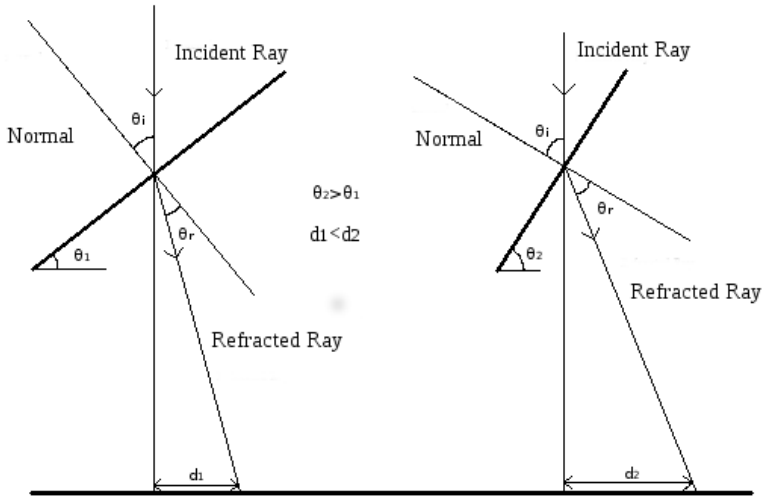

Figure 2.3: Increase in the shift of pixel because of higher slope.

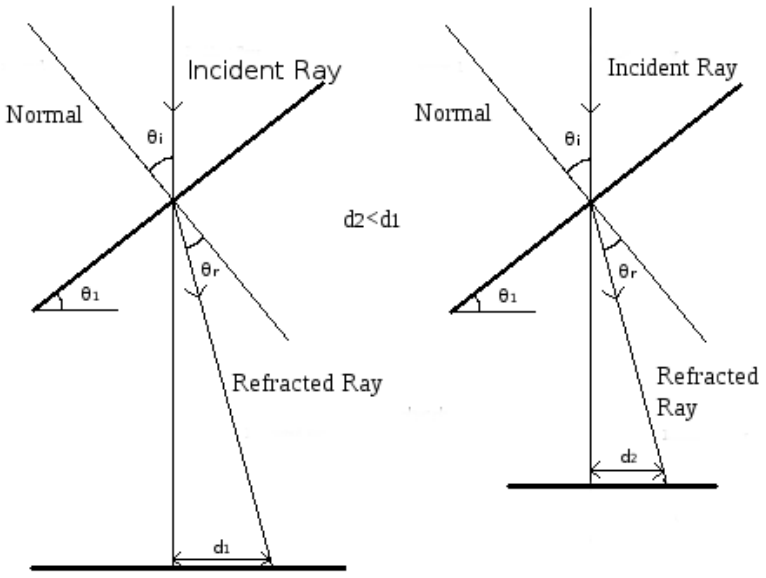

Figure 2.4: Increase in the shift of pixel because of increase in distance between surface and object/camera.

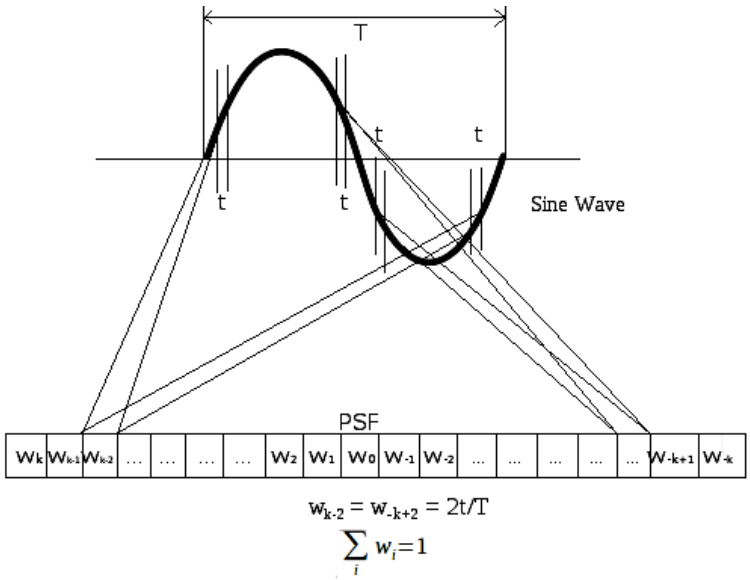

Figure 2.5: Slope and weight correspondence of Psf.

2.3.2 Weights of PSF

The weights of psf reflect the information about the time for which a particular slope appears in the water waveform. Since, psf for image is represented using pixels, therefore, because of discretization, one pixel doesn't correspond to one particular slope but to a range of slopes and the weight value in that pixel block represents the fraction of total time that slopes' range appears in the water waveform. Fig 2.5 shows the way following which psf can be computed from the waveform. This approach is used in section 2.5 to compute psf from the shape of water waveform.

2.4 Water Surface Waveform Estimation

This problem deals with reconstruction of water surface waveform using a single blurred image. The idea is to use blind deconvolution to obatin the psf estimate for the image. And relating the length and weights of psf with with the slopes and time for which those slopes are seen in the waveform, an estimate of water surface waveform is done.

Mathematical Desciption

Lets take following as the known psf:

w_k	w_{k-1}	...	..	w_1	w_0	w_{-1}	..	...	w_{-k+1}	w_{-k}

The length of psf is $2k+1$ and w_i are the weights with corresponding slopes represented by s_i. From equations 13,14

$$\omega(x,\tau) = \mathbf{P}\alpha h_x$$

Considering 1D water waveforms for simplicity, however, the equations remain true for 2D water waveforms also. Since, $\mathbf{P}$ and α are constants, the maximum warping corresponds to maximum slope as explained in section 2.3.1. Let t be pixel length (assuming pixels are square in shape), therefore, maximum slope is given by:

$$\mathbf{P}\alpha s_{max} = (k-1)t \tag{17}$$

Hence,

$$s_{max} = s_k = \mathbf{P}^{-1}\frac{(k-1)t}{\alpha} \tag{18}$$

Similarly, maximum negative slope would be:

$$s_{-max} = s_{-k} = -\mathbf{P}^{-1}\frac{(k-1)t}{\alpha} \tag{19}$$

Since, the distribution of slope is linear, all s_i can now be computed easily.

$$\Delta s = \frac{s_k - s_{-k}}{2k+1} \tag{20}$$

$$s_{k-1} = s_k - \Delta s \tag{21}$$

$$s_{k-2} = s_{k-1} - \Delta s \tag{22}$$

$$s_{-k+1} = s_{-k+2} - \Delta s \tag{23}$$

and so on. Let T be total exposure time and T_i be time corresponding slope range is present in the waveform. Therefore:

$$\sum_{i=-k}^{k} T_i = T \tag{24}$$

The T_i are related to weights of psf by following relations:

$$T_i = \frac{w_i}{\sum_{j=-k}^{k} w_j} T \tag{25}$$

Waveform Reconstruction

Neglecting the spatial parameter as the same waveform is seen at all pixels and naming it as $y(t)$:

$$y(t) = \begin{cases} s_k t & 0 \le t \le T_k \\ s_k T_k + s_{k-1} t & T_k \le t \le T_{k-1} \\ \vdots & \vdots \\ \sum_{i=k}^{-k+1} s_i T_i + s_{-k} t & T_{-k+1} \le t \le T_{-k} \end{cases} \tag{26}$$

Results

The input blurred image Fig 2.6:

The computed psf:

0.13	0.13	0.08	0.07	0.07	0.06	0.07	0.07	0.08	0.13	0.12

In the case of sinusoidal wave (in general any symmetric waveform), only half of the wave can be generated using the psf as the psf for 2^{nd} half is exactly same as the 1^{st}. Since, psf is already normalized, detecting repetition is not possible. However, this can be easily resolved by making assumption that water waveform has to differentiable at each point, which is totally logical as non-differentiable waveforms don't occur in normal conditions. They may occur in turbulant flows and some haphazard motion of water waves but that can be safely neglected as those cases are rare and not considered in this work.

Figure 2.6: Input blurred image.

Estimated Waveform

This method generates one of many possible waveforms that would result in the blurred image and more on this ambiguity is discussed in the section 2.6. The x-axis represents time, and it depends upon fps and exposure time of camera. The y-axis information is irrelevant as discussed in section 2.2, amplitude of wave is considered to be small and waveform is assumed as just slopes at surface of water with negligible height.

2.5 PSF Estimation

The psf estimation from single blurred image is ill-posed problem. The blurred image obtained by camera is convolution of ground image with the psf. Assuming **g** is the blurred image of **f**. So, the following holds neglecting presence of noise:

$$\mathbf{g} = \mathbf{f} * k \tag{27}$$

Hence, there may be different combinations of **f** and **k** which will give same **g** and estimating the correct **f** and **k** is difficult without any additional information. Usually some prior are assumed on **f** and **k**, which reduces this complexity to some extent but not

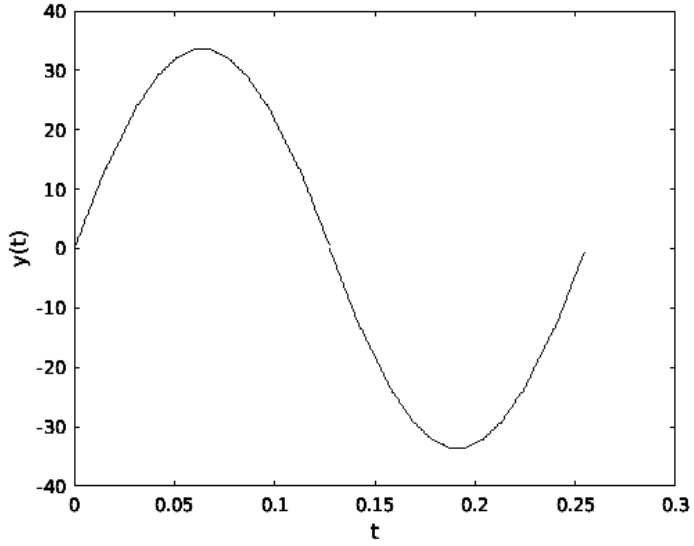

Figure 2.7: The estimated waveform.

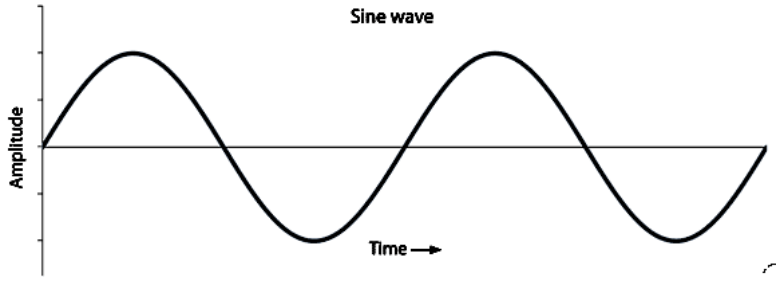

Figure 2.8: The sine wave.

completely.

If the information about shape of waveform is considered in the psf estimation, the estimation can be highly improved. The challenge in this how to get the information about the shape of water waveform. This has been done using optical flow. However, use of optical flow increases the complexity of solution.

In general, water waveform created as result of wind, follows trochoid shape, which is closely related to sinusoidal wave for smaller amplitude. In this report, sinusoidal waveform is assumed, however, similar psf model can be developed for trochoids and other waveforms.

2.5.1 PSF model for sinusoids

$$y(t) = Asin(\omega t) \tag{28}$$

$$y'(t) = A\omega cos(\omega t) = A' cos(t) \tag{29}$$

Here, the effect of ω can be modeled by modifying the amplitude.

Maximum slope=$\tan^{-1}$A

The center of psf corresponds to zero slope. On each side of zero slope pixel, there are k pixels. Therefore, there are (k+0.5) pixel on both sides of zero slope. Therefore:

$$(k + 0.5)x = \tan^{-1} A \tag{30}$$

$$x = \frac{2\tan^{-1} A}{2k + 1} \tag{31}$$

This associates each psf block with range of angles. For i^{th} pixel angle range will be from $\frac{(2i+1)\tan^{-1}A}{2k+1}$ to $\frac{(2i+3)\tan^{-1}A}{2k+1}$ for i=0,1,...k-1.

Exactly similar expressions will hold for negative slopes and i=-1,-2,...-k.

Computation of Weights:

For i^{th} block:

$$\tan(\frac{(2i + 1)\tan^{-1} A}{2k + 1}) = A cos(t) \tag{32}$$

$$T_i = \cos^{-1}[\frac{1}{A}\tan(\frac{(2i + 1)\tan^{-1} A}{2k + 1})] \tag{33}$$

$$T_{i+1} = \cos^{-1}[\frac{1}{A}\tan(\frac{(2i + 3)\tan^{-1} A}{2k + 1})] \tag{34}$$

$$\Delta T_i = |T_i - T_{i+1}|$$

$$\Delta T_i = \cos^{-1}[\frac{1}{A}\tan(\frac{(2i + 1)\tan^{-1} A}{2k + 1})] - \cos^{-1}[\frac{1}{A}\tan(\frac{(2i + 3)\tan^{-1} A}{2k + 1})] \tag{35}$$

The weights w_i of psf are ΔT_i multiplied by some constant, however, estimating the constant is not required as psf is itself normalized. Hence, given the length $2k + 1$ of psf and amplitude of wave, using above equations, psf of sinusoidal wave can be estimated. Following is psf of sine wave for length of 11 (k=5) and amplitude 1.

0.231	0.085	0.062	0.052	0.047	0.045	0.047	0.052	0.062	0.085	0.231

Figure 2.9: Input blurred image.

2.5.2 PSF estimation from the Blurred Image

As explained before, the blur has information about the pixel movement across the frames and that information relates to slope of water waveform.

Algorithm:

1. Divide the blurred mxn image into say N smaller sub images.
2. Estimate the psf for all sub images and the complete image using tradional approach to get N+1 psf.
3. Use K-mean clustering to group similar psf into single group and it is assumed group with higher element has psf closer to actual psf.
4. Start with some amplitude A for sinusoidal wave, compute the psf.
5. Minimize the error between the psf from the group and sinusoidal wave. Gradient descent is used for updating the A, such that error reduces.
6. Finally, the last step is to stop the loop, if further improvement is not possible or is smaller than some threshold. Since the obtained psf is coming from shape of wave itself, it must be better estimate. This can be verified by computing Kernel Similarity (47) with the actual kernel.

Results:

The input blurred image Fig 2.9:

Figure 2.10: The clean image.

Since, it is synthetic experiment, original image is also known Fig 2.10:

Psf estimate by tradional approach:

0.144	0.217	0.087	0.038	0.023	0	0	0	0	0	0
0	0	0	0	0	0	0.039	0.088	0.217	0.146	

Psf estimate by new approach:

0.140	0.060	0.047	0.041	0.037	0.035	0.033	0.032	0.031	0.030	0.030
0.030	0.031	0.032	0.033	0.035	0.037	0.041	0.047	0.060	0.140	

Since, this is synthetic experiment, ground psf is also known:

0.075	0.094	0.055	0.047	0.039	0.039	0.035	0.035	0.031	0.031	0.031
0.031	0.031	0.035	0.035	0.039	0.039	0.047	0.055	0.090	0.075	

Comparsion of ground truth psf with both methods show the significance of adding shape of wave information in the psf estimation. The closeness of psf by new approach to ground truth psf can be seen.

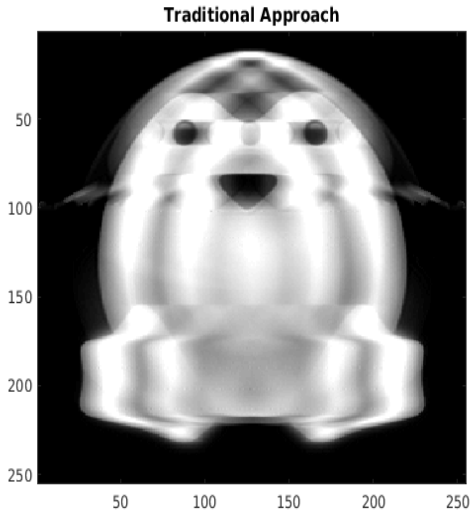

Figure 2.11: Deblurred image using traditional approach.

Figure 2.12: Deblurred image using new approach.

Kernel similarity with ground psf of tradional approach: 0.8230

Kernel similarity with ground psf of new approach: 0.9158

The kernel similarity matrix is used to measure the closeness of a kernel with ground truth kernel, where 0 represents no similarity and 1 represents same kernel.

Further the estimated psf can be used for deblurring the image.

Deblurring using tradional approach Fig 2.11:

Deblurring using new approach Fig 2.12:

Clearly, the new algorithm performs better than existing approach as it adds the information about shape of water wave to the existing psf estimation method which improves the psf estimation significantly.

This is shown for sine wave only, however, similar approach can be extended for other waveforms and even to non-periodic waveforms.

2.6 Conclusion

The first problem of estimation of water surface waveform using the psf has multiple solutions. Using the approach mentioned in section 2.4, one of those solutions is estimated. This ambiguity appears impossible resolve as there may be multiple intermediate frames whose average can give the same blurred image.

Psf contains the information about the slopes present in the waveform for e.g say slopes from -$\frac{\pi}{4}$ to $\frac{\pi}{4}$ are in the waveform but psf doesn't contain the information about time like which slope is appearing at what time or which slope is coming after which slope or about the repetition of slopes.

The second problem of estimation of psf requires prior knowledge of water surface waveform. Apart from this constraint, the solution seems to improve psf estimation significantly.

CHAPTER 3

Unsupervised Deblurring

In image deblurring, the transformation from blur to clean domain is a many-to-one mapping while clean to blur is the vice versa depending on the extent and nature of blur. Thus, it is difficult to capture the domain knowledge with the existing architectures of CoGAN, DualGAN, CycleGAN. Also, the underlying idea in all these networks is to use a pair of GANs to learn the domains, but usually training GANs is highly unstable (38) and thus using two GANs simultaneously escalates in stability issues in the network. Instead of using a second GAN to learn the blur domain, we use a CNN network for reblurring the output of GAN and a gradient module to constrain the solution space. A detailed description of each module is provided below.

GAN proposed by Goodfellow (21) consists of two networks (a generator and a discriminator) that compete to outperform each other. Given the discriminator D, the generator tries to learn the mapping from noise to real data distribution so as to fool D. Similarly, given the generator G, the discriminator works as a classifier that learns to distinguish between real and generated images. The function of learning GAN is a min-max problem with the cost function

$$E(D,G) = \max_{D}\min_{G} \underset{x\sim P_{data}}{E}[\log D(x)] + \underset{z\sim P_z}{E}[\log(1-D(G(z)))]. \tag{3.1}$$

where z is random noise and x denotes the real data. This work was followed by conditional GANs (cGAN) (39) that use a conditioning input in the form of image (23), text, class label etc. The objective remains the same in all of these i.e, the discriminator is trained to designate higher probability to real data and lower to the generated data. Hence, the discriminator acts as a data prior that learns clean data domain similar to the heuristics that are used in conventional methods. In the proposed network, the input to generator G is a blurred image $y \in Y$ and the generator maps it to a clean image $\hat{x}$ such that the generated image $\hat{x} = G(y)$ is indistinguishable from clean data (where clean data statistics are learned from $\tilde{x}$s $\in X$).

Self-supervision by reblurring (CNN Module) The goal of GAN in proposed deblurring framework is to reach an equilibrium where P_{clean} and $P_{generated}$ are close. The

alternating gradient update procedure (AGD) is used to achieve this. However, this process is highly unstable and often results in mode collapse (40). Also, an optimal G that translates from $Y \rightarrow X$ does not guarantee that an individual blurred input y and its corresponding clean output x are paired up in a meaningful way, i.e, there are infinitely many mappings G that will induce the same distribution over $\hat{x}$ (13). This motivated the use of reconstruction loss ($||\hat{x} - x||^2$) and perceptual loss ($||\Phi_i((\hat{x}) - \Phi_i(x)||^2$, where Φ_i represents VGG module features extracted at the i^{th} layer) along with the adversarial loss in many supervised learning works (10; 22; 23; 24; 41), to stabilize the solution and help in better convergence. Since all these require ground truth, which is unavailable in this case, we use the blurred image y itself as a supervision to guide in deblurring. Adding such a module ensures that the deblurred result has the same color and texture comparable to the input image thereby constraining the solution to the manifold of images that captures the actual input content.

Gradient matching Module With a combined network of GAN and CNN modules, the generator learns to map to clean domain along with color preservation. Now, to enforce the gradients of the generated image to match its corresponding clean image, we use a **gradient module** in the network as given in Fig 1.1. Gradient matching resolves the problem of over-sharpening and ringing in the results. However, since the reference image is unavailable, determining the desired gradient distribution to match with is difficult. For this, a heuristic from (43) that takes advantage of the fact that shrinking a blurry image y by a factor of α results in a image y^{α} that is α times sharper than y is used. Thus, the blurred image gradients are used at different scales to guide the deblurring process. At the highest scales, the gradients of blurred and generated output match the least but improve while going down in scale space. A visual diagram depicting this effect is shown in Fig 3.1 where the gradients of a blurred and clean checker-board at different scales are provided. Observe that, at the highest scale, the gradients are very different and as we move down in scale the gradients start to look alike and the L_1 error between them decreases.

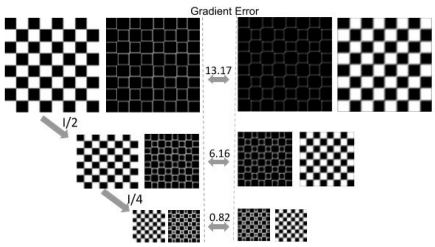

Figure 3.1: Scale space gradient error.

3.1 Loss Functions

3.1.1 Tried out loss functions

The feature which differentiate a blur image from clean is sharpness. A clean image is much sharper than the blurred image. In order to classify any image as clean or blurred, we have various focus operators like Gradient energy, Gaussian derivative, 3D gradient, Energy of Laplacian, Modified Laplacian, Histogram entropy etc. Generally, the objective is to find an operator that behaves in a stable and robust manner over variety of images and for this purpose, the commonly used focus operator is sum modified laplacian (SML).

SML

Mathematically, the SML focus operator on an image I is computed as:

$$\nabla^2 I = |\frac{\partial^2 I}{\partial x^2}| + |\frac{\partial^2 I}{\partial y^2}| \tag{3.2}$$

We incorported SML module in GAN architecture. First we tried in supervised framework. Given a blurred image, the generator output and clean image, we trained generator to minimize the loss between the SML value of clean image and generator output. This helped generator to learn features of clean image. However, it is supervised as availability of clean image of corresponding blurred input image is assumed in this case. The results with different weights fo SML on deblurring is given below:

From Fig 3.2, it's clear that SML module with GAN is able to perform deblurring. The output images are sharp and visually appreciable. However, it suffer from problem of color preservation. With too high weight or too low weight to SML cause severe color problems in the ouput.

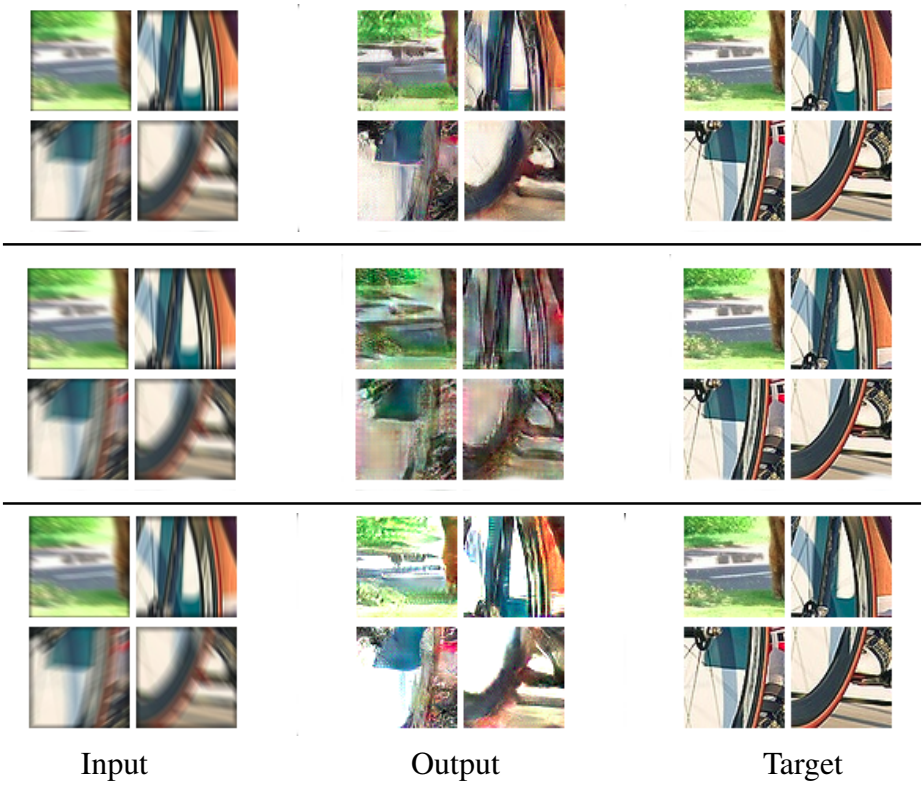

Figure 3.2: SML(weight 0.1) with GAN (row 1&2), SML (weight 100) with GAN (row3 &4), SML (weight 0) with GAN (row 5 & 6).

Then, we tried without GAN. The same architecture was used, however the GAN loss was neglected and only SML loss was considered. However, it didn't work. The generator was producing some random images which had same SML value as of clean image but it wasn't learning the content of the image. Hence, we dropped the idea of using SML alone.

We also tried to make the whole GAN + SML architecture unsupervised. For this we computed the histogram for SML values of clean and blurred images. And found a threshold over the SML values to categorize an image as clean or blur based on the average SML. However, this also didn't work as expected and we came up with a better modules to solve the problem of unsupervised deblurring.

In short, we started with SML module in supervised framework. It was performing fine, doing decent work at deblurring. However, since our final aim was to make the entire architecture unsupervised, we moved to its unsupervised version. We further tried with SML in GAN framework and VAE framework before finally arriving to the final network consisting of a GAN module, a CNN module for reblurring and a gradient module for matching the gradients.

3.1.2 Final loss function

A straightforward way for unsupervised training is by using GAN. Given large unpaired data $\{x_i\}_{i=1}^{M}$ and $\{y_j\}_{j=1}^{N}$ in both domains, train the parameters (θ) of the generator to map from $y \rightarrow x$ by minimizing the cost

$$L_{\text{adv}} = \min_{\theta} \frac{1}{N} \sum_{i} \log(1 - D(G_\theta(y_i))) \tag{3.3}$$

Training with adversarial cost alone can result in color variations or missing finite details (like eyes and nose in faces or letters in case of texts) in the generated outputs but the discriminator can still end up classifying it as real instead of generated data. This is because discriminating between real and fake does not depend on these small details (see Fig 3.3(b), the output of GAN alone wherein eyes and colors are not properly reconstructed).

With the addition of the reblurring module, the generator is more constrained to match the colors and textures of the generated data (see Fig 3.3(c)). The generated clean image from generator $\hat{x} = G(y)$ is again passed through the CNN module to obtain back the blurred input. Hence the reblurring cost is given as

$$L_{\text{reblur}} = ||y - \text{CNN}(\hat{x})||_2^2 \tag{3.4}$$

Along with the above two costs, the gradients are forced to match at different scales (s) using the gradient cost defined as

$$L_{\text{grad}} = \sum_{s \in \{1,2,4,8,16\}} \lambda_s |\nabla y_{s\downarrow} - \nabla \hat{x}_{s\downarrow}| \tag{3.5}$$

where ∇ denotes the gradient operator. A Laplacian operator $\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$ is used to calculate the image gradients at different scales and λ_s values are set as [0.0001, 0.001, 0.01, 0.1, 1] for $s = \{1,2,4,8,16\}$, respectively. Adding the gradient cost removes unwanted ringing artifacts at the boundary of the image and smoothens the result. It is evident from the figure that with inclusion of supporting cost functions corresponding to reblurring and gradient, the output (Fig 3.3(d)) of the network becomes comparable

(a) (b) (c) (d) (e)

Figure 3.3: Effect of different cost functions. (a) Input blurred image to the generator, (b) result of unsupervised deblurring with just the GAN cost in eq. (3.3), (c) result obtained by adding the reblurring cost in eq. (3.4) with (b), (d) result obtained with gradient cost in eq. (3.5) with (c), and (e) the target output.

with the ground truth (GT) image (Fig 3.3(e)). Hence, we traine the generator network with a combined cost function given by

$$L_G = \gamma_{\text{adv}} L_{\text{adv}} + \gamma_{\text{reblur}} L_{\text{reblur}} + \gamma_{\text{grad}} L_{\text{grad}} \tag{3.6}$$

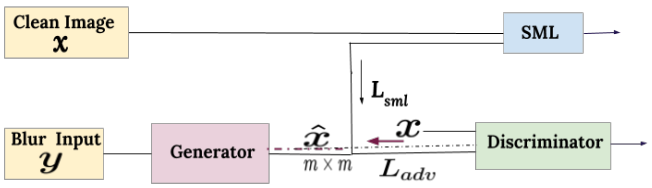

Figure 3.4: GAN network with SML module

3.2 Network Architecture

3.2.1 Tried Network Architecture

GAN with SML

Fig 3.4 presents the network architecture of GAN along with SML for supervised training. The clean image sml value is replaced by pre defined sml value classifying images as clean and blur to train the model in unsupervised manner. The impacts and results obtained by this network architecture is already discussed in section 3.1.1.

GAN with Gradient Module

As the GAN network with SML module didn't perform upto expectation, we switched to a gradient module. The idea as explained in section 3.1.2 was, as we scale down blur images, the extent of blur reduces and the gradients of blur and clean image starts to match. Hence, we introduced this gradient matching module which downscales the GAN output and computes the gradient losses between the down scaled versions of GAN output by factors of 1,2,4,8 and 16 and blur image down scaled by same factors correspondingly. This helped in better learning of GAN for the task of deblurring. Results are given in section 3.1.2

3.2.2 Final Network Architecture

We use similar architecture for generator and discriminator as proposed in (24), which has shown good performance for blind-superresolution, with slight modification in the feature layers. The network architecture of GAN with filter sizes and the number of feature maps at each stage is provided in Table 3.1. Each convolution (conv) layer in the

Table 3.1: The proposed generator and discriminator network architecture. conv ↓ indicates convolution with stride 2 which in effect reduces the output dimension by half and d/o refers to dropout.

Module	**Generator**										**Discriminator**					
Layers	conv	conv	conv	conv	conv	conv	conv	conv	conv	conv	conv↓	conv↓	conv↓	conv↓	conv ↓	fc
Kernel Size	5	5	5	5	5	5	5	5	5	5	4	4	4	4	4	-
Features	64	128	128	256 d/o (.2)	256 d/o (.2)	128	128	64	64	3	64	128	256	512	512	-

Table 3.2: Reblurring CNN module architecture.

Module	**CNN**					
Layers	conv	conv	conv	conv	conv	tanh
Kernel Size	5	5	5	5	5	
Features	64	64	64	64	3	

generator is followed by batch-normalization and non-linearity using Rectified Linear Unit (ReLU) except the last layer. A hyper tangential (Tanh) function is used at the last layer to constrain the output to $[-1, 1]$. The discriminator is a basic 6-layer model with each convolution followed by a Leaky ReLU except the last fully connected (fc) layer which is followed by a Sigmoid. Convolution with stride 2 is used in most layers to go down in dimension and the details of filter size and feature maps are provided in Table 3.1. The reblurring CNN architecture is a simple 5-layer convolutional module provided in Table 3.2. We operate the gradient module on-the-fly for each batch of data using GPU based convolution with the Laplacian operator and downsampling depending on the scaling factor with 'nn'modules.

Torch is used for training and testing with the following options: ADAM optimizer with momentum values $\beta_1 = 0.9$ and $\beta_2 = 0.99$, learning rate of 0.0005, batch-size of 32 and the network was trained with the total cost as provided in eq. (3.6). The weights for different costs were initially set as γ_{adv}=1, γ_{grad}=.001 and γ_{reblur}=0.01 to ensure that the discriminator learns the clean data domain. After around 100K iterations the adversarial cost was weighted down and the CNN cost was increased so that the clean image produced corresponds in color and texture to the blurred input. Hence, we readjusted the weights, as γ_{adv}=.01, γ_{grad}=.1 and γ_{reblur}=1 and reduced the learning rate to .0001 to continue training. Apart from these, to stabilize the GAN, during training, we use drop-out of 0.2 at the fourth and fifth convolution layers of the generator and a smooth labeling of real and fake labels following (38).

3.3 Experiments

The experiments section is arranged as follows: (i) training and testing datasets, (ii) comparison methods, (iii) quantitative results, metrics used and comparisons, and (iv) visual results and comparisons.

3.3.1 Dataset Creation

For all classes, we use 128 × 128 sized images for training and testing. The dataset generation for training and testing of each of these classes is explained below. Note that we traind the network for each of these classes separately.

Camera Motion Generation: In the experiments, to generate the blur kernels required for synthesizing the training and test sets, we use the methodology described by Chakrabarthi in (44). The blur kernels are generated by randomly sampling six points in a limited size grid (13 × 13), fitting a spline through these points, and setting the kernel values at each pixel on this spline to a value sampled from a Gaussian distribution with mean 1 and standard deviation of 0.5, then clipping these values to be positive, and normalizing the kernel to have unit sum. We use a total of 100K kernels for creating the dataset.

Face Dataset: We use the aligned CelebA face dataset (45) for creating the training data. CelebA is a large-scale face attributes dataset of size 178 × 218 with more than 200K aligned celebrity images. 200K images were selected from it and resized to 128 × 128 and divided into two groups. Then one group is blurred.

Text Dataset: For text images, we used training dataset of Hradiš et al. (37) which consists of images with both defocus blur generated by anti-aliased disc and motion blur generated by random walk. They have provided a large collection of 66K text images of size 300×300. We cropped the images to 128×128 and grouped into blur and clean domains containing 33K images each.

Checkerboard Dataset: We take a clean checkerboard image of size 256 × 256 and applied random rotations and translations to it and cropped out 128 ×128 (avoiding boundary pixels) to generate a set of 100K clean images. Then, we partitioned the clean images into two sets of 50K images each to ensure that there are no corresponding pairs available during training. To one set synthetic motion blur is applied to create

the blurred images by convolving with linear filters and the other set is kept as such.

3.3.2 Comparison Methods

The deblurring results are compared with three classes of approaches, (a) State-of-art conventional deblurring approaches which use prior based optimization, (b) Supervised deep learning based end-to-end deblurring approaches, and (c) latest unsupervised image-to-image translation approaches.

Conventional Single image deblurring: The comparison is done with the state-of-the-art conventional deblurring works of Pan et al. (18) and Xu et al. (19) that are proposed for natural images. In addition to this, for face deblurring, the deblurring work in (36) is used that is designed specifically for faces. Similarly for text, comparison is done with the method in (46) that uses prior on text for deblurring. Quantitative results are provided by running their codes on created test dataset. **Deep supervised deblurring**: In deep learning, for quantitative analysis on all classes,we do comparison with end-to-end deblurring work of (10) and additionaly for text and checkerboard comparison with (37). The work in (10) is a general dynamic scene deblurring framework and (37) is proposed for text deblurring alone. All these methods use paired data for training and hence are supervised. Besides these for visual comparisons on face deblurring, we also compared with (30) on their images since the trained model was not available.

Unsupervised image-to-image translation: We train the cycleGAN (13) network for deblurring task. We trained the network from scratch for each class separately and quantitative and visual results are reported for each class in the following sections.

3.3.3 Quantitative Analysis

For quantitative analysis, the test sets are created for which the ground truth is available to report the metrics mentioned below. For text dataset, we used the test set provided in (37). And for checkerboard, we used synthetic motion parametrized with $\{l, \theta\}$. For faces, test sets are created using the kernels generated from (44).

Quantitative Metrics: We use PSNR (in dB), SSIM and Kernel Similarity Measure(KSM) values for comparing the performance of different state of art deblurring algorithms on all the classes. For texts, apart from these metrics, we used Character Error Rate (CER)

to evaluate the performance of various deblurring algorithms.

CER (37) is defined as $\frac{i+s+d}{n}$, where, n is total number of characters in the image, i is the minimal number of character insertions, s is the number of substitutions and d is the number of deletions required to transform the reference text into its correct OCR output. We use ABBYY FineReader 11 to recognize the text and its output formed the basis for evaluating the mean CER. Smaller the CER value, better the performance of the method.

Kernel Similarity Measure: In general practice, the deblurring efficiency is evaluated through PSNR, SSIM metric or with visual comparisons. These commonly used measures (MSE) are biased towards smooth outputs due to 2-norm form. Hence, Hu et al. (47) proposed KSM to evaluate deblurring in terms of the camera motion estimation efficiency. KSM effectively compare estimated kernels ($\hat{K}$) evaluated from the deblurred output with the ground truth (K). It is computed as $S(K,\hat{K}) = \max_\gamma \rho(K,\hat{K},\gamma)$ where $\rho(.)$ is the normalized cross-correlation function given by ($\rho(K,\hat{K},\gamma) = \frac{\sum_\tau (K(\tau).K(\hat{\tau+\gamma}))}{||K||.|\hat{K}||}$) and γ is the possible shift between the two kernels. The larger the value, the better the kernel estimate and indirectly the better the deblurring performance.

Results and Comparisons: Table 3.3 summarizes the quantitative performance of various competitive methods along with proposed network results for all the three classes. A set of 30 test images from each class is used to evaluate the performance reported in the table. It is very clear from the results that the unsupervised network performs on par with competitive conventional methods as well as supervised deep networks. Conventional methods are highly influenced by parameter selection. The results could perhaps be improved further by fine-tuning the parameters for each image but this is a time-consuming task. Though deep networks perform well for class-specific data, their training is limited by the lack of availability of large collections of paired data. It can be seen from Table 3.3 that proposed network (without data pairing) is able to perform equally well when compared to the class-specific supervised deep method (37) for text deblurring. The network even outperform the dynamic deblurring network of (10) in most cases. The cycleGAN (13)(though unsupervised) struggles to learn the blur and clean data domains. It can be noted that, for checkerboard, cycleGAN performed better than the unsupervised network in terms of PSNR and SSIM. This is because checkerboard had simple linear camera motion. Because blur varied for text and faces (general camera motion) the performance of cycleGAN also deteriorated (refer to the reported

values).

Table 3.3: Quantitative comparisons on face, text, and checkerboard datasets.

		Face dataset			Text dataset				Checkerboard dataset		
	Method	**PSNR**	**SSIM**	**KSM**	**PSNR**	**SSIM**	**KSM**	**CER**	**PSNR**	**SSIM**	**KSM**
Conventional Methods	Pan et al. (46)	-	-	-	16.19	0.7298	0.8628	0.4716	11.11	0.3701	0.7200
	Pan et al. (18)	19.38	0.7764	0.7436	17.48	0.7713	0.8403	0.3066	13.91	0.5618	0.7027
	Xu et al. (19)	20.28	0.7928	0.7166	14.22	0.5417	0.7991	0.2918	8.18	0.2920	0.6034
	Pan et al.(36)	22.36	0.8523	0.7197	-	-	-	-	-	-	-
Deep learning Methods	Nah et al. (10)	24.12	0.8755	0.6229	18.72	0.7521	0.7467	0.2643	18.07	0.6932	0.6497
	Hradiš et al. (37)	-	-	-	24.28	0.9387	0.9435	0.0891	18.09	0.6788	0.6791
Unsupervised technique	Zhu et al. (13)	8.93	0.4406	0.2932	13.19	0.5639	0.8363	0.2306	21.92	0.8264	0.6527
	Proposed	22.80	0.8631	0.7536	23.22	0.8792	0.9376	0.126	20.61	0.8109	0.7801

Real Handshake Motion: In addition, to test the capabilities of the trained network on real camera motion, we do testing for face and text classes using the real camera motion dataset from (48). Camera motion provided in (48) contains 40 trajectories of real camera shake by humans who were asked to take photographs with relatively long exposure times. These camera motions are not confined to translations, but consist of non-uniform blurs, originating from real camera trajectories. The efficiency of the proposed network in deblurring images affected by these real motions is reported in Table 3.4. Since long exposure leads to heavy motion blur which is not within the scope of this work, short segments of the recorded trajectory is used to introduce small blurs. Table 3.4 shows the PSNR, SSIM between the clean and deblurred images and KSM between the estimated and original motion. The handshake motion in (48) produces space-varying blur in the image and hence a single kernel cannot be estimated for the entire image. Patches (32×32) from the image are used and space-invariant blur is assumed over the patch to extract the kernel and KSM is computed. This was repeated on multiple patches and an average KSM is reported for the entire image. The KSM, PSNR, and SSIM are all high for both the classes signifying the effectiveness of proposed network to deal with real camera motions.

3.3.4 Visual Comparisons

The visual results of the network and competitive methods are provided in Figs. 3.5 and 3.6. Fig. 3.5 contains the visual results for text and checkerboard data. Comparisons

Table 3.4: Quantitative comparisons on face and text on real handshake motion (48).

Class	**PSNR** in (dB)	**SSIM**	Kernel Similarity **KSM**
Text	21.92	0.8968	0.8811
Face	21.40	0.8533	0.7794

are provided with (19; 18) and (46). The poor performance of these methods can be attributed to the parameter setting. Most of these results have ringing artifacts. Now, to analyse the performance of the network over supervised networks, we do comparison with the dynamic deblurring network of (10) and class-specific deblurring work of (37). From the visual results it can be clearly observed that even though the method in (10) gave good PSNR in Table 3.3 it is visually not sharp and some residual blur remains in the output. The supervised text deblurring network (37) result for checkerboard was sharp but the squares were not properly reconstructed. The inefficiency of cycleGAN to capture the clean and blur domains simultaneously is reflected in the text results. On the contrary, the proposed unsupervised network produces sharp and legible (see the patches of texts) results in both these classes. Proposed network outperforms existing conventional methods and at the same time works on par with the text-specific deblurring method of (37).

Visual results on face deblurring are provided in Fig. 3.6. Here too comparison with

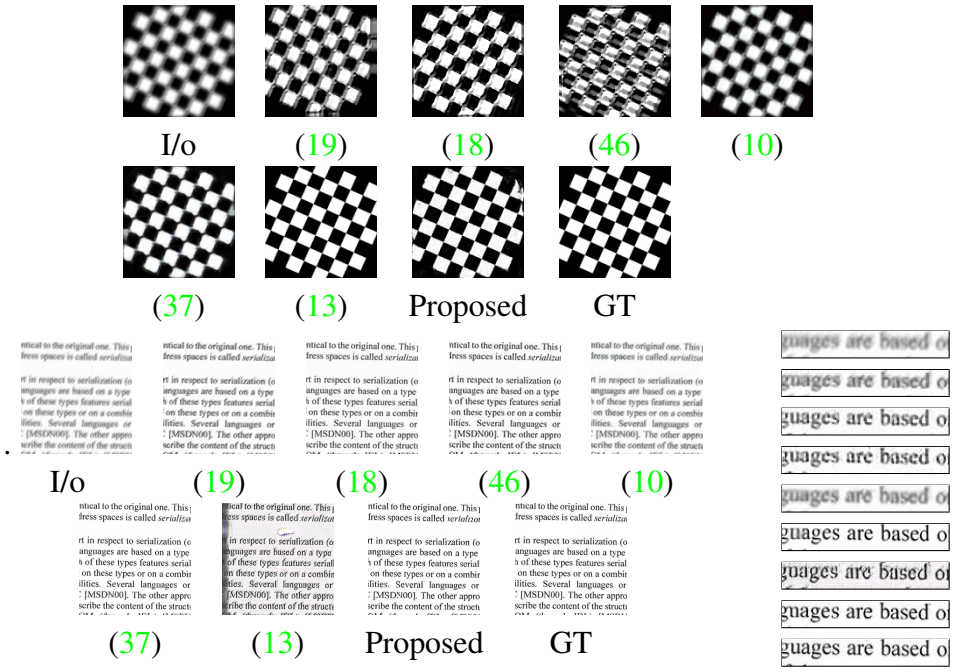

Figure 3.5: Visual comparison on checkerboard deblurring. Input blurred image, deblurred results from conventional methods (19), (18) and (46), results from supervised network in (10), (37) and unsupervised network (13), Proposed network result and the GT clean image are provided in that order.

conventional methods (19; 18) as before and the exemplar-based face-specific deblurring method of (36) is done. Though these results are visually similar to the GT, the effect of ringing is high with default parameter settings. The results from deep learning work of (10) is devoid of any ringing artifacts but are highly oversmoothened. Simi-

Figure 3.6: Visual comparisons on face deblurring.

larly, CycleGAN (10) fails to learn the domain properly and the results are quite different from the GT. On the other-hand, results of proposed network are sharp and visually appealing. While competitive methods failed to reconstruct the eyes of the lady in Fig. 3.6 (second row), proposed method reconstructs the eyes and produces sharp outputs comparable to GT.

3.4 Conclusions

We propose a deep unsupervised network for deblurring class-specific data. The proposed network does not require any supervision in the form of corresponding data pairs. A reblurring cost and scale-space gradient cost are introduced that are used to self-supervise the network to achieve stable results. The performance of proposed network was found to be at par with existing supervised deep networks on both real and synthetic datasets. The method paves the way for unsupervised image restoration, a domain where availability of paired dataset is scarce.

REFERENCES

[1] Seemakurthy, K., Rajagopaln, A.N.: Deskewing of Underwater Images. In: IEEE Transactions on Image Processing (TIP). (2015)

[2] Schettini, R., Corchs, S.: Underwater image processing: state of the art of restoration and image enhancement methods. In: EURASIP Journal on Advances in Signal Processing (EURASIP). (2010)

[3] Turlaev, D., Dolin, L.: On observing underwater objects through a wavy water surface: A new algorithm for image correction and laboratory experiment. In: Atmospheric and Oceanic Physics. (2013)

[4] Shefer, R., Malhi, M., Shenhar, A.: Waves distortion correction using cross correlation. In: Technical Report, Israel Institute of Technology. (2001)

[5] Murase, H.: Surface shape reconstruction of a nonrigid transparent object using refraction and motion. In: IEEE transactions on pattern analysis and machine intelligence (PAMI). (1992)

[6] Efros, A., Isler, V., Shi, J., Visontai, M.: Advances in Neural Information Processing Systems. In: Pages 393-400. (2005)

[7] Wen, Z., Lambert, A., Fraser, D., Li, H.: Bispectral analysis and recovery of images distorted by a moving water surface. In: Applied optics, Optical Society of America. (2010)

[8] Wen, Z., Lambert, A., Fraser, D.: Bicoherence: a new lucky region technique in anisoplanatic image restoration. In: Applied optics, Optical Society of America. (2009)

[9] Tian, Y., Narasimhan, S.G.: Seeing through water: Image restoration using model-based tracking. In: 2009 IEEE 12th International Conference on intelligence Computer Vision (IEEE). (2009)

[10] Nah, S., Kim, T.H., Lee, K.M.: Deep multi-scale convolutional neural network for dynamic scene deblurring. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (July 2017)

[11] Nimisha, T., Singh, A.K., Rajagopalan, A.: Blur-invariant deep learning for blind-deblurring. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV). (2017)

[12] Kupyn, O., Budzan, V., Mykhailych, M., Mishkin, D., Matas, J.: Deblurgan: Blind motion deblurring using conditional adversarial networks. arXiv preprint arXiv:1711.07064 (2017)

[13] Zhu, J.Y., Park, T., Isola, P., Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. arXiv preprint arXiv:1703.10593 (2017)

[14] Yi, Z., Zhang, H., Tan, P., Gong, M.: Dualgan: Unsupervised dual learning for image-to-image translation. arXiv preprint (2017)

[15] Liu, M.Y., Breuel, T., Kautz, J.: Unsupervised image-to-image translation networks. In: Advances in Neural Information Processing Systems. (2017) 700–708

[16] Su, S., Delbracio, M., Wang, J., Sapiro, G., Heidrich, W., Wang, O.: Deep video deblurring for hand-held cameras. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2017) 1279–1288

[17] Ma, Z., Liao, R., Tao, X., Xu, L., Jia, J., Wu, E.: Handling motion blur in multi-frame super-resolution. In: Proceedings of the IEEE Computer Vision and Pattern Recognition (CVPR). (2015) 5224–5232

[18] Pan, J., Sun, D., Pfister, H., Yang, M.H.: Blind image deblurring using dark channel prior. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2016) 1628–1636

[19] Xu, L., Zheng, S., Jia, J.: Unnatural l0 sparse representation for natural image deblurring. In: Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on, IEEE (2013) 1107–1114

[20] Gupta, A., Joshi, N., Zitnick, C.L., Cohen, M., Curless, B.: Single image deblurring using motion density functions. In: European Conference on Computer Vision, Springer (2010) 171–184

[21] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Advances in neural information processing systems. (2014) 2672–2680

[22] Ledig, C., Theis, L., Huszár, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z., et al.: Photo-realistic single image super-resolution using a generative adversarial network. arXiv preprint (2016)

[23] Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. arxiv (2016)

[24] Xu, X., Sun, D., Pan, J., Zhang, Y., Pfister, H., Yang, M.H.: Learning to super-resolve blurry face and text images. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2017) 251–260

[25] Aytar, Y., Castrejon, L., Vondrick, C., Pirsiavash, H., Torralba, A.: Cross-modal scene networks. IEEE transactions on pattern analysis and machine intelligence (2017)

[26] Liu, M.Y., Tuzel, O.: Coupled generative adversarial networks. In: Advances in neural information processing systems. (2016) 469–477

[27] Gatys, L.A., Ecker, A.S., Bethge, M.: Image style transfer using convolutional neural networks. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2016) 2414–2423

[28] Johnson, J., Alahi, A., Fei-Fei, L.: Perceptual losses for real-time style transfer and super-resolution. In: European Conference on Computer Vision, Springer (2016) 694–711

[29] Gatys, L.A., Ecker, A.S., Bethge, M., Hertzmann, A., Shechtman, E.: Controlling perceptual factors in neural style transfer. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2017)

[30] Chrysos, G., Zafeiriou, S.: Deep face deblurring. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). (2017)

[31] Rengarajan, V., Balaji, Y., Rajagopalan, A.: Unrolling the shutter: Cnn to correct motion distortions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2017) 2291–2299

[32] Anwar, S., Phuoc Huynh, C., Porikli, F.: Class-specific image deblurring. In: Proceedings of the IEEE International Conference on Computer Vision. (2015) 495–503

[33] Anwar, S., Porikli, F., Huynh, C.P.: Category-specific object image denoising. IEEE Transactions on Image Processing **26**(11) (2017) 5506–5518

[34] Teodoro, A.M., Bioucas-Dias, J.M., Figueiredo, M.A.: Image restoration with locally selected class-adapted models. In: IEEE International Workshop on Machine Learning for Signal Processing (MLSP). (2016) 1–6

[35] Ulyanov, D., Vedaldi, A., Lempitsky, V.S.: Deep image prior. CoRR **abs/1711.10925** (2017)

[36] Pan, J., Hu, Z., Su, Z., Yang, M.H.: Deblurring face images with exemplars. In: European Conference on Computer Vision, Springer (2014) 47–62

[37] Hradiš, M., Kotera, J., Zemcík, P., Šroubek, F.: Convolutional neural networks for direct text deblurring. In: Proceedings of BMVC. Volume 10. (2015)

[38] Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., Chen, X.: Improved techniques for training gans. In: Advances in Neural Information Processing Systems. (2016) 2234–2242

[39] Mirza, M., Osindero, S.: Conditional generative adversarial nets. arXiv preprint arXiv:1411.1784 (2014)

[40] Goodfellow, I.: Nips 2016 tutorial: Generative adversarial networks. (2016)

[41] Xie, J., Xu, L., Chen, E.: Image denoising and inpainting with deep neural networks. In: Advances in neural information processing systems. (2012) 341–349

[42] Ignatov, A., Kobyshev, N., Vanhoey, K., Timofte, R., Van Gool, L.: Dslr-quality photos on mobile devices with deep convolutional networks. In: the IEEE Int. Conf. on Computer Vision (ICCV). (2017)

[43] Michaeli, T., Irani, M.: Blind deblurring using internal patch recurrence. In: European Conference on Computer Vision, Springer (2014) 783–798

[44] Chakrabarti, A.: A neural approach to blind motion deblurring. In: European Conference on Computer Vision, Springer (2016) 221–235

[45] Liu, Z., Luo, P., Wang, X., Tang, X.: Deep learning face attributes in the wild. In: Proceedings of International Conference on Computer Vision (ICCV). (2015)

[46] Pan, J., Hu, Z., Su, Z., Yang, M.H.: Deblurring text images via l0-regularized intensity and gradient prior. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2014) 2901–2908

[47] Hu, Z., Yang, M.H.: Good regions to deblur. In: European Conference on Computer Vision, Springer (2012) 59–72

[48] Köhler, R., Hirsch, M., Mohler, B., Schölkopf, B., Harmeling, S.: Recording and playback of camera shake: Benchmarking blind deconvolution with a real-world database. In: European Conference on Computer Vision, Springer (2012) 27–40

[49] Lai, W.S., Huang, J.B., Hu, Z., Ahuja, N., Yang, M.H.: A comparative study for single image blind deblurring. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2016) 1701–1709

[50] Punnappurath, A., Rajagopalan, A.N., Taheri, S., Chellappa, R., Seetharaman, G.: Face recognition across non-uniform motion blur, illumination, and pose. IEEE Transactions on image processing **24**(7) (2015) 2067–2082

LIST OF PAPERS BASED ON THESIS

1. Nimisha T.M., Sunil Kumar, A.N. Rajagopalan:, **'Unsupervised Class-Specific Deblurring'**. *European Conference on Computer Vision (ECCV)*(2018) [submitted].