ROLLING SHUTTER DEBLURRING USING PATCH-WISE APPROXIMATION

A Project Report

submitted by

AMIT KASANA

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

BACHELOR OF TECHNOLOGY

DEPARTMENT OF ELECTRICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY MADRAS.

May 2018

THESIS CERTIFICATE

This is to certify that the thesis titled **ROLLING SHUTTER DEBLURRING USING PATCH-WISE APPROXIMATION** by **Amit Kasana**, to the Indian Institute of Technology, Madras, for the award of the degree of **Bachelor of Technology**, is a bona fide record of the research work done by him under our 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.

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

Place: Chennai

Date: 10th May 2018

ACKNOWLEDGEMENTS

I take this opportunity to express my deepest gratitude to my project guide Dr. A.N. Rajagopalan for his valuable guidance and motivation throughout the project. I am very grateful to him for providing his valuable time to guide me during the project.

It is a privilege to be a student in IIT Madras. I express special thanks to all my teachers for all the academic insight obtained from them. I also acknowledge the excellent facilities provided by the institute to the students.

I am indebted to my parents and my brother for their unconditional love, support and guidance. I dedicate this thesis to them.

ABSTRACT

KEYWORDS: Rolling-shutter motion blur, motion density function, blind deblurring, efficient filter flow.

Today we see the world by eyes of technology i.e. camera. But a lot of images as they are affected by motion blur and rolling-shutter(RS) distortion. Today, most of the imaging devices like mobile phone, drone etc. are equipped with CMOS based sensors, which consume less power while delivering fast-high performance and advanced features. The RS distortion is a frequent knock on CMOS sensors. It occurs while shooting video, trying to capture a fast-moving subject or moving the camera while shooting. Unlike Global-shutter, in CCD where all the pixel is exposed at the same time, in CMOS each row of RS sensor array is exposed sequentially one-after-other, same as reading book reading a book(line by line).

In this work I have implemented a model for rolling-shutter blind motion delblurring (using single blurred image) that was proposed by R. *et al.* (2017). This project reduces computational cost, need for precise sensor information and inability to deal with wide angle systems and irregular camera trajectory.

This project investigates the removal of spatially-variant blur from photographs degraded by rolling-shutter as well as motion blur, and the removal of large occluding objects from photographs of popular places. These problems are examined in the case where the photographs are taken with standard consumer cameras with CMOS sensor, and don't have any particular information about the scene being photographed. However, It is shown that blur from camera shake is in general mostly due to the 3D motion of the camera, resulting in a rolling-shutter and motion blur that can be significantly non-uniform across the image. We model this blur using a weighted set of camera poses, which induce homographies on the image being captured. The blur in a particular image row(as image is captured row by row) is parameterized by the set of weights, which provides a compact global descriptor for the blur, analogous to a convolution kernel. In order to reduce the computational cost of our homography-based model, an efficient filter flow(EFF) was introduced based approximation on local-uniformity of the blur.

TABLE OF CONTENTS

LIST OF FIGURES

ABBREVIATIONS

3**D**	Three-dimensional
EFF	Efficient filter flow
MAP	Maximum a posterior
MDF	Motion density function
CCD	Charged Coupled Device
CMOS	Complementary metalâĂŞoxideâĂŞsemiconductor
RS	Right camera coordinate systemRolling shutter
MB	Motion blur
BMD	Bind Motion Deblurring
GS	Global Shutter
FFT	Fast Fourier Transformation
RAM	Random access memory
CPU	Central processing unit

CHAPTER 1

INTRODUCTION

With the explosion of digital photography in recent years, many of us take large numbers of digital photos with digital cameras or camera-phones and most of these cameras are equipped with CMOS camera because it consume less power while delivering fast-high performance and advanced features. When we review our photos later however, there is sometimes a divergence between what we remember seeing at the time, and what our cameras recorded. This disparity can perhaps be explained by the old adage that we see with our brains, not with our eyes. Whether our photos contain a luridly-dressed tourist that our brain had filtered out at the time, or our photos appear blurry due to camera shake that causes rolling-shutter(RS) distortion and motion blur(MB), it is not uncommon that the photos we find ourselves with do not capture what we wanted to record.

Avoiding motion during capture is not always possible. Hand shakes during hand-held camera capture, mounting cameras on a mobile platform, etc. are some instances where camera motion cannot be avoided. So, it becomes necessary to rectify images from rolling shutter distortions. Single image RS rectification however is a challenging task because RS motion-affected image itself might look correct unless prior properties of the scene are known. A straight line in the scene may look curved in the image due to the RS motion, but that does not make the image unnatural in the sense that the recorded curve in the image could in fact correspond to an actual curve in the scene. Hence, it is only logical to learn the naturalness or unnaturalness of specific scene classes such as urban scene, indoor scene, and human faces, as against commenting on the absolute nature of any type of image.

In this project, I implemented models and methods, aimed at "restoring" photographs to bring them closer to the images we hoped to record at the time of their taking, that was proposed by R. *et al.* (2017). For many people, existing photos capture important and fleeting moments that it may be impossible to recapture. As opposed to proposing new hardware or methods for taking future photographs, we are predominantly concerned with handling images that have already been captured. By using accurate models of the image formation process, and incorporating strong prior information on the images we would like to recover, we endeavor to make software post-processing the "brain" to the digital "eyes" of our cameras. Specifically, our objective is to automatically restore photographs, when these photographs contains unwanted occluders , or when they are blurry due to RSMB. Besides the emotional motivation for improving peoples' photos more concert motivations for these tasks might be that need to recover visual information from blurry image along with camera trajectory and blur kernel.

1.1 Problem Statement

The types of image degradation considered in this work lead to inherently ill-posed image restoration problems. Starting from only a degraded image(i.e., rolling shutter and motion blur), we wish to recover a good image of the same scene with underlying camera motion. If there is a loss of information, or we have more unknowns than observations, there may be a large family of valid solutions, which we must somehow choose amongst when producing the "restored" image. This is the case in the problem discussed here. For deblurring images, where the blur is unknown, we must estimate both the parameters describing the blur and the sharp image. Since the sharp image has the same number of pixels as the blurry image, we evidently have more unknowns than observations. When we wish to remove occluders from photographs, the recorded image contains no information about what is behind the occluder; it could be concealing a building, a tree, a patch of grass, or a group of people. Making the right choice and producing visually-pleasing results requires good models of the image formation process, and equally importantly, good prior information about the unknowns.

1.2 Thesis outline

This thesis will explore solutions to two problems - 1) Using a blur image and dataset of Sharpe image, face recognition and finding best match from dataset 2) single image rolling shutter motion blur correction. This thesis is organized as follows: The second chapter presents the recent works done in the field and related terminology. The third chapter presents the face recognition and matching using Siamese neural networks model that is trained for a a single image. The fourth chapter addresses the motion blur model and rolling-shutter motion blur model.

1.3 Summary of contributions

The following is the summary of the contributions of this thesis:

- In Chapter **??**, we present a novel Siamese neural networks architecture for face recognition and face matching using a blured image and clear image. Instead of providing a lots of images for training I have given single sharpe image of person and and blurred image of same person.

- In Chapter **??**, I have shown comparison of my implementation with R. *et al.* (2017) and Whyte *et al.* (2012).

CHAPTER 2

BACKGROUND AND RELATED WORK

Today with explosion of digital photography, use of digital camera is increased. To make camera energy-efficient, high performance and advance feature, most of imaging device are equipped with CMOS sensors. CMOS camera uses **rolling-shutter**, where CCD camera uses global-shutter. Rolling shutter is a method of image capture in which a still picture (in a still camera) or each frame of a video (in a video camera) is captured not by taking a snapshot of the entire scene at a single instant in time but rather by scanning across the scene rapidly, either vertically or horizontally. In other words, not all parts of the image of the scene are recorded at exactly the same instant. (Though, during playback, the entire image of the scene is displayed at once, as if it represents a single instant in time.) This produces predictable distortions of fast-moving objects or rapid flashes of light. This is in contrast with "global shutter" in which the entire frame is captured at the same instant.

The "rolling shutter" can be either mechanical or electronic. The advantage of this method is that the image sensor can continue to gather photons during the acquisition process, thus effectively increasing sensitivity. It is found on many digital still and video cameras using CMOS sensors. The effect is most noticeable when imaging extreme conditions of motion or the fast flashing of light. While some CMOS sensors use a global shutter, the majority found in the consumer market use a rolling shutter.

CCDs (charge-coupled devices) are alternatives to CMOS sensors, which are generally more sensitive and more expensive. CCD-based cameras often use global shutters, which take a snapshot representing a single instant in time and therefore do not suffer from the motion artifacts caused by rolling shutters,as shown in Fig. 2.1.

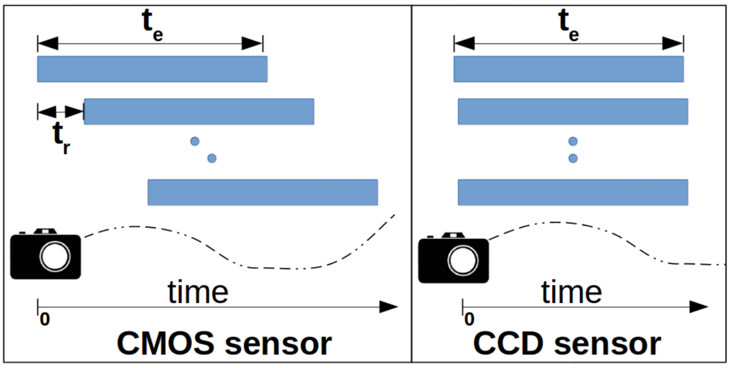


Figure 2.1: Working of rolling-shutter(left side) and global shutter(right side), where t_e) is exposer time and t_r) is read-out time.

2.1 Distortion effects of Rolling Shutter

2.1.1 Rolling shutters can cause such effects as :

Wobble This phenomenon (also known as the jello effect) appears when the camera is vibrating, in situations such as hand-held shots at telephoto settings, or when shooting from a moving vehicle. The rolling shutter causes the image to wobble unnaturally.

Skew The image bends diagonally in one direction or another as the camera or subject moves from one side to another, exposing different parts of the image at different times. Skew is a minor manifestation of the wobble phenomenon

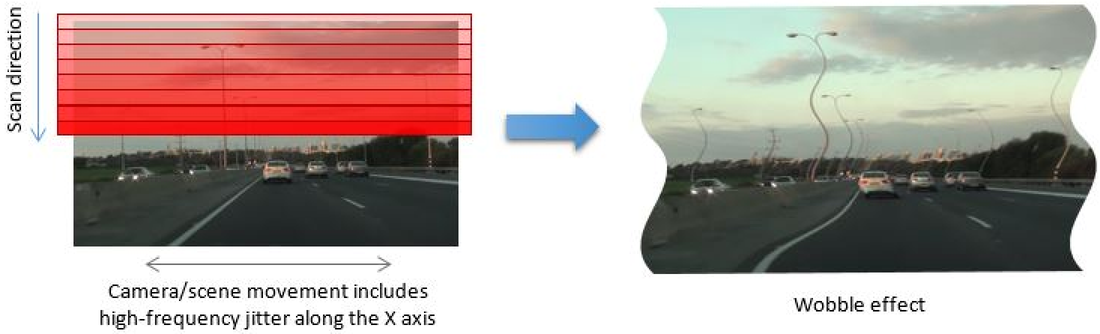

Scan direction
Camera/scene movement includes high-frequency jitter along the X axis
Wobble effect

Figure 2.2: Wobble effect caused by object or camera motion.

described above.

Spatial aliasing . Vertically adjacent pixels are sampled in violation of the sampling theorem, when the camera or object motion is too rapid. One example of this is imaging of a quickly rotating propeller. The smear of each blade is caused by the propeller rotating at the same or near the same speed that the frame is read by the camera. Viewed perpendicular to a fan spinning clockwise, the blades on the left side appear thinner than usual while the blades on the right side appear thicker, and can even appear as if they aren't connected at the center.

Temporal aliasing , including Partial Exposure. If a camera flash goes on for only part of the time of the exposure, the illumination of the flash may only be present for some rows of pixels in a given frame. For example, the top 1/3 of the picture may be brightly lit by a flash, while the bottom 2/3 of the picture is dark and unlit, as the flash was off by the time that part of the CMOS was sequenced. The difference between the two distinct parts of the frame can look odd. Similar problems can arise with fluorescent lighting, strobe effects, lightning, or any extreme situation where very fast motion or very fast bursts of light are seen in the time between when the CMOS chip sequentially records a frame.

The effects of a rolling shutter can prove difficult for visual effects filming. The process of match moving establishes perspective in a scene based on a single point in time, however this is difficult with a rolling shutter that provides multiple points in time within the same frame. Final results depend on the readout speed of the sensor and the nature of the scene being filmed; as a rule of thumb, higher end cinema cameras will have faster readout speeds and therefore milder rolling shutter artifacts than low end cameras.

Images and video that suffer from rolling shutter distortion can be improved by algorithms that do rolling shutter rectification, or rolling shutter compensation. How to do this is an active area of research.

2.2 Basic Defination & terminology

2.2.1 Efficient Filter Flow

An idea key to EFF framework was introduced by Stockham, who presented the overlap-add (OLA) method for fast convolution and correlation. TO increase computation efficiency and to reduce memory requirement EFF is used.computing each element of P is essentially equivalent to performing a sub-pixel interpolation, which is computationally expensive.Spaitally varing blur can be approximated as locally uniform.

Divide image into patches, considering these have spatially invariant blur, make computation easy.One weakness of considering separate filter per patch is that in regions with little texture or strong edges, it may not be possible to estimate a good kernel. The usual (overlap-add)OLA method chops the image into overlapping patches, damps the borders of each patch with some windowing function, convolves each patch with the same filter, and then adds the transformed patches to obtain the output image. If each patch is processed with its own filter we get a space-variant linear filter.

2.2.2 Gradient of image

We found that strong edges do not always profit kernel estimation, but instead under certain circumstance degrade it. This finding leads to a new metric to measure the usefulness of image edges in motion deblurring and a gradient selection process to mitigate their possible adverse effect.

Image derivatives help to reduce ringing artifacts by putting weight on the edges. Secondly, it lowers the condition number of the optimization problem and hence leads to faster convergence .

2.2.3 Conditional Number

It is used to measure how sensitive a function is to changes or errors in the input, and how much error in the output results from an error in the input. A system is said to be singular if the condition number is infinite, and ill-conditioned if it is too large. For example, the condition number associated with the linear equation Ax = b gives a bound on how inaccurate the solution x will be after approximation.

2.2.4 35 mm focal length

This means that when you shoot at this focal length you are giving your viewers a vantage point similar to if they were on the scene. The bottom line is that the standard fields of view from 35mm cameras can be duplicated by digital cameras with their smaller sensors and lenses by using wider focal length lenses.If you wanted to know the *35mm* equivalent of a *50mm* lens used on a digital camera with an *APS-C* size image sensor, you would just multiply the *50mm* focal length by *1.5. (50mm X 1.5 = 75mm)* So, in this example, the *35mm* focal length equivalent for the *50mm* lens would be *75mm*. A standard *35mm* film image is *36mm* wide : so to convert focal length into pixel *35mm_focal_length x image width / 36*.

2.3 Recent works

A large body of literature is devoted to estimation of image degradations. A full review of all the literature is outside the scope of this thesis. Instead we will discuss the relevant prior work that relates directly to: estimation of blur, segmenting dynamic objects from 3D scenes and Rolling Shutter Rectification.

Blur Estimation Estimating blur kernels in images is a very ill-posed problem. So, different assumptions on blur kernels and natural images have been incorporated. Fergus *et al.* (2006) exploit the knowledge that the gradient histogram of natural images exhibits a heavy-tailed profile and that blur kernels due to camera shake have a sparse support. Chakrabarti *et al.* (2010) estimates motion blur based on blur spectrum analysis of image patch in Fourier transform space. Levin (2007) predicts motion blur kernel using natural image statistics. Couzinie-Devy *et al.* (2013) learns a regression function to predict motion blur kernel based on some hand-crafted features. The current state-of-the-art rolling-shutter and morion blur,blind deblurring Su and Heidrich (2015) eliminates problem of device-specific constrains of and estimates times tamped ego-motion solely from image intensities. However, the method is limited to parametric ego-motion derived specifically for hand-held blur.

Different to these methods, R. *et al.* (2017) have preseted a method that can handle narow-angle as well as wide angle system. This method also computationally effective, can be sused for CCD motion blur as well as rolling shutter motion blur.

CHAPTER 3

ROLLING SHUTTER & MOTION BLUR MODELING

3.1 Motion Blur

3.1.1 Introduction

A common problem faced while capturing photographs is the occurrence of motion blur due to camera shake. The extent of blurring at a point in an image varies according to the camera motion as well as the depth value resulting in space-variant blur. Recovering the clean image from known camera motion i.e. blur kernel is called *non-blind deblurring*, while of both the clean image and underlying camera motion from a single motion blurred image is called *blind motion deblurring*.

3.1.2 Motion blur model

As the problems discussed in this thesis, we begin with a general forward model of image degradation. There are four important components in the image degradation process:

- The (observed) degraded image *g* which is the image output by the camera's sensor.

- The (latent) sharp image f: This is the underlying, ideal, sharp image of the scene, which we would like to recover.
- The degradation operator(also known as blur kernel) h that acts on f, and which describes how light from the sharp image f is distributed in the observed image g. Depending on the situation,h may be known in advance (e.g. from optical properties of the camera) or maybe unknown (e.g. arbitrary camera motion during camera shake).
- The random noise n that perturbs the recorded image after the sharp image has been degraded by h.

So, traditionally a blurry image can be modeled as combination of noisy image(n) and clear image(f) convolved with kernel(h). Mathematically can be represented as follow:

$$g = f * h + n \tag{3.1}$$

And in case of non-blind deblurring model can be given as Fig. 3.1.

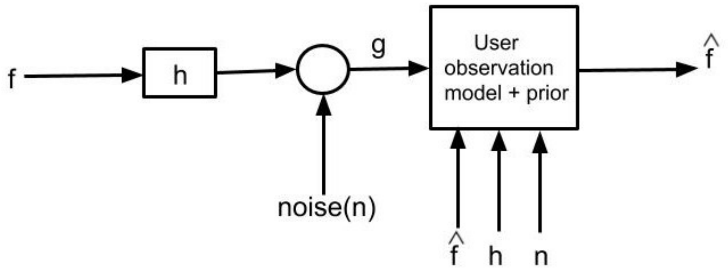


Figure 3.1: Perceptron model

As proposed in Whyte *et al.* (2012), a blurry image or video frame g can be modeled as weighted sum of images observed over a number of camera poses during exposer time. We assume form now that the only significant motion of the camera is a rotation about its optical center, and that the scene being photographed static. As in most cases of camera shake, the rotation of the camera during exposure has a significantly larger effect than its translation Levin *et al.* (2009): Typical consumer cameras have a field of view of about 50°, and rotating the camera by only a small angle during exposure e.g. $1^\circ = \frac{1}{50} * 50^\circ$ cause an image blur whose size in the image approximately follows the same proportion, i.e., a 20 pixel blur on a 1000 pixel-wide image. Translation of the camera, on the other hand, will cause a blur whose size is inversely proportional to the depth of the scene, and reaching the same 20 pixel blur for an object at a depth of 2m would require translating the camera by about 4cm. The rotation of 1°represents significantly smaller motion of the camera and, in most cases, camera rotation can be assumed to be the only significant source of camera shake blur.

Under the pinhole model of a camera, all views seen by the camera are projectively equivalent, excluding boundary effects. This means that the image at one camera orientation is related to the image at any other by a 2D projective transformation, or homography. For an uncalibrated camera, this is a general 8-parameter homography, but in the case of a camera with known internal parameters, the homography **H** is parameterized by the 3×3 rotation matrix **R** describing the rotation of the camera Hartley and Zisserman (2004).

$$\mathbf{H} = \mathbf{KRK}^{-1} \tag{3.2}$$

where **K** is the camera's internal calibration matrix.

The matrix **R** requires only 3 parameters, and we adopt here the "angle-axis" representation, in which a rotation is described by the angle θ moved about an axis a (a unit-norm 3-vector), summarized by the vector $\theta = \theta_a = (\theta_x, \theta_y, \theta_z)$. We fix our 3D coordinate frame to have its origin at the camera's optical center, with the XY -plane aligned with the camera sensor's coordinate frame and the Z-axis parallel to the camera's optical axis. **R** is given by the matrix exponential.

$$\begin{bmatrix}\theta\end{bmatrix}_X = \begin{bmatrix} 0 & -\theta_Z & \theta_Y \\ \theta_Z & 0 & -\theta_X \\ -\theta_Y & \theta_X & 0 \end{bmatrix} \tag{3.3}$$

Having defined the type of image transformation we expect, we now assume that when the shutter of the camera opens, there is a sharp image $f : \mathbb{R}^2 \to \mathbb{R}$ of a static scene that we would like to capture. The camera's sensor accumulates photons while the shutter is open, and outputs an observed image $g : \mathbb{R}^2 \to \mathbb{R}$. In the ideal case, each point on the sensor would see a single scene point throughout the exposure, giving us a sharp image. However if, while the shutter is open, the camera undergoes a sequence of rotations, parametrized by $\boldsymbol{\theta}$(t), each ray from the static scene will trace a sequence of points on the image. For each point x in the observed blurred image we can trace the record of rays **x**'(t)

$$x'(t) \sim \mathbf{H}_t x \tag{3.4}$$

where $\mathbf{H}_t$ is the homography induced by the rotation $\boldsymbol{\theta}$(t), and $\sim$ denotes equality up to scale. The observed image g is thus the integral over the exposure time T of all the projectively- transformed versions of f , plus some observation noise ε:

$$g(x) = \int_0^T f(\mathbf{H}_t x)dt + \varepsilon \tag{3.5}$$

where, with a slight abuse of notation,$\mathbf{H}_t x$ denotes inhomogeneous coordinates of a point in *f*.

In general, a single blurry image has no temporal information associated with it, so it is convenient to replace the temporal integral in (3.5) by a weighted integral over the set of all possible rotations Γ

$$g(x) = \int_{\Gamma} f(\mathbf{H}_{\theta} x) \omega_{\boldsymbol{\theta}} dt + \varepsilon \tag{3.6}$$

where the weight function $\omega(\boldsymbol{\theta})$ corresponds to the time the camera spends at the orientation

3.2 RSMB Model

Although motion blur and rolling shutter deformations are closely coupled artifacts(as both are cause by relative motion of camera and object) in images taken with CMOS imaging sensors, the two phenomena have so far mostly been treated separately, with deblurring algorithms being unable to handle rolling shutter wobble, and rolling shutter algorithms being incapable of dealing with motion blur. In this section, we have discussed the generative model for RSMB. As mentioned earlier, the entire image in CCD imaging sensor experiences same ego motion, so motion blurred image **B** $\epsilon \mathbb{R}^{M \times N}$ is generated by integrating the images seen by the camera along its trajectory during the exposure duration $[0, t_e]$ Su and Heidrich (2015). It is given by following equation:

$$\mathbf{B} = \frac{1}{t_e} \int_0^{t_e} \mathbf{L}^{\mathbf{P}(t)} dt \tag{3.7}$$

where $\mathbf{P}(t)$ represents the general 6D camera pose at time instant t_0, $\mathbf{L}^{\mathbf{P}(t)}$ is the latent image $\mathbf{L}$ transformed according to pose $\mathbf{P}(t)$, and t_e is shutter speed. As we have already discussed, each RS sensor-row experience different ego-motion due to staggered exposure window (Fig. 2.1). Unlike CCD, here we cannot associate a global warp for the entire latent image L, but need to consider each row separately. Image row $\mathbf{B}_i$ (where *i* indicate *i*th row) of an RS blurred image $\mathbf{B} = [\mathbf{B}_1^T \mathbf{B}_2^T \dots \mathbf{B}_M^T]^T$ is given by:

$$\mathbf{B} = \frac{1}{t_e} \int_{(i-1).t_r}^{(i-1).t_r + t_e} \mathbf{L}_i^{\mathbf{P}(t)} dt \ \ : i \, \epsilon \, \{1, 2, \dots, M\} \tag{3.8}$$

where $\mathbf{L}_i^{\mathbf{P}(t)}$ is the ith row of the transformed image $\mathbf{L}^{\mathbf{P}(t)}, t_e$ is the shutter speed or row-exposure time in CMOS camera, and t_r is the inter row delay(time gap between starting time of current row and last row). All the current RS deblurring methods use a discretized form of Eq. 3.8 as the forward model, and we refer to this as temporal model. By considering weighted integration of the transformed image-row over camera pose Eq. 3.8 can be written as following:

$$\mathbf{B} = \int_{\mathbf{P}} \omega'_p.(\mathbf{p}) \mathbf{L}_i^{\mathbf{P}} d\mathbf{p} \ \ : i \, \epsilon \, \{1, 2, \dots, M\} \tag{3.9}$$

where $\mathbf{P}$ is the pose-space and ω'_p is the weight corresponding to the transformed row $\mathbf{L}^{\mathbf{P}_0}$. Unlike existing RS deblurring works, the second model is also employed by R. *et al.* (2017) and by discretization the pose-space in Eq. 3.9, new equation can be written as follow:

$$\mathbf{B} = \sum_{\mathbf{p} \, \epsilon \, \mathbb{P}} \omega'_p.(\mathbf{p}) \mathbf{L}_i^{\mathbf{P}} d\mathbf{p} \ \ : i \, \epsilon \, \{1, 2, \dots, M\} \tag{3.10}$$

here discretization step-size is taken such that there is less than one pixel displacement between two adjacent poses. Where $\mathbb{P}$ is the discretized pose-space $\mathbf{P}$, and the discrete weight $\omega_i(p_0)$ is the summation of all the continuous-weight $\omega'_i(p)$ for all $\mathbf{p}$ that lie int the half step-size neighborhood of pose $\omega'_i(p)$. We identify the weights $\omega_i(p)$ as the motion density function (MDF), as in (Gupta *et al.*, 2010)

3.3 RS Deblurring Computational Approach

In this section we are going to discuss formulation of maximum a posteriori (MAP) framework for latent image and kernel estimation, EEF for rolling-shutter blur, optimization for ego-motion estimation, latent image estimation etc.

3.3.1 RS Deblurring

For estimation of both the latent image and the block-MDFs, R. *et al.* (2017) formulate a maximum a posteriori (MAP) framework. In this section,new prior for MDF to enable RS motion is introduced. A direct MAP framework for unknown $\theta = \{\mathbf{L}, \mathbf{w}_i \,:\, 1 \leqslant i \leqslant n_b\}$ is given as

$$\hat{\theta} = \min_{\theta} \sum_{i=1}^{n_b} \|\mathbf{B}_i - \sum_{\mathbf{p}\,\epsilon\,\mathbb{P}} \omega_i(\mathbf{P}).\mathbf{L}_i^{\mathbf{p}}\|_2 + \lambda_1 \|\nabla \mathbf{L}\|_1 + \lambda_2 \sum -i = 1^{n_b} \|\mathbf{W}_i]\|_1 \tag{3.11}$$

where $\mathbf{w}_i$ is the vector containing weights $\omega_i(\mathbf{P})$ for poses $\mathbf{p} \,\epsilon\, \mathbb{P}$ and $\nabla\mathbf{L}$ is the gradient of $\mathbf{L}$. Here assumption is made that is the optimal block-size r_b is known(size of r_b is such that any block of contiguous rows with size less than or equal to r_b will have substantial camera-pose overlap). The first term in the objective is data fidelity that enforces forward blur model of Eq. 3.10. To reduce ill-posedness, R. *et al.* (2017) to enforce a sparsity prior on the image-gradient as explained in 2.2.2 and used in Whyte *et al.* (2012). A sparsity prior is also imposed on the MDF weights since a camera can transit over only few poses in $\mathbb{P}$ during exposure. In the literature on CCD deblurring (i.e., $n_b = 1$) it is well-known that the objective in Eq. 3.11 is biconvex, i.e., it is individually convex with respect to the latent image and MDF, but non-convex overall and convergence to a local minima is ensured with alternative minimization of MDF and latent image Whyte *et al.* (2012). However, RS sensors introduce a different challenge if Eq 3.11 is directly considered.

For an RS blured image, multiple solutions are possible for the latent image-MDF pair in each individual image block. They satsfy the forward model in Eq 3.10 and consistent with the image and MDF prior in Eq 3.11, as information regarding direction has lost.

3.3.2 Model and Optimization

State-of-the-art CCD-BMD methods (Whyte *et al.* (2012), Xu *et al.* (2013), Cho and Lee (2009)) work by alternative minimization (AM) of MDF and latent image over a number of iterations in a scale-space manner, i.e., AM proceeds from coarse to fine image-scale(i.e., image size is changed in every step) in order to handle large blurs. This requires generation of blur numerous times. Efficiency of the blur-ring process is a major factor that governs computational efficiency of a method. In this section, firstly pose-space model and how it allows for an efficient process for RS blurring (analogous to CCD-EFF (Hirsch *et al.* (2011) Hirsch *et al.* (2010))) is discussed.

Efficient Filter Flow for RS blur

Here approximation is made that motion blur in individual small image patches is space invariant convolution with different blur kernels. It is represented as:

$$\mathbf{B} = \sum_{k=1}^{R} \mathbf{C}_k^{\dagger}.\{a^{(k,b(k))} \ * \ ((\mathbf{C}_k.\mathbf{L}))\} \tag{3.12}$$

where R is the total number of overlapping patches in latent image $\mathbf{L}$, $b(k)$ is a function which gives the index of the block to which the major portion of the k^{th} patch belongs(i.e., $b(k) \;\epsilon\; \{\mathbf{1}, \mathbf{2}, \ldots \mathbf{n_b}\;\}$), $\mathbf{C_{k.L}}$ is a linear operation which extracts the *k*th patch from$\mathbf{L}$ and $\mathbf{C_k^\dagger}$ inserts the patch back to its original position with a windowing operation. $a^{(k,b(k)}$ represents the blur kernel which when convolved with the *k*th latent image-patch creates blurred patch. Considering *b(k)* as **j**, we can write

$$a^{(k,b(k))} \;=\; \sum_{\mathbf{k=1p}\;\epsilon\;\mathbb{P}} \omega_{\mathbf{j}}(\mathbf{p}).\delta_k(\mathbf{P}) \tag{3.13}$$

where $\delta_k(\mathbf{P})$ is a shifted impulse obtained by transforming with pose **p** an impluse center at the *k*th patch-center. Intuitively, the blur kernel at patch *k* due to an arbitrary MDF is the superposition of the $\delta_k(\mathbf{P})$s generated by it. Since $\delta_k(\mathbf{P})$ is independent of the latent image and the MDF, it needs to be computed *only once*, and be subsequently used to create the blur kernel in patch *k* for any image. Thus, given a latent image **L** and MDF of each block, our blurring process first computes kernels in R patch-centres using the precomputed $\delta_k(\mathbf{P})$ Eq. 3.13, convolves them with their corresponding latent-image patches and combines them to form the RS blurred image Eq. 3.12. We carry out convolution using the efficient FFT. Note that the CCD-EFF is a special case of Eq 3.12. under identical MDFs $(\mathbf{w}_i \;=\; \mathbf{w}\forall\mathbf{i})$ or the single block case $(n_b = 1)$.

Ego-Motion Estimation

The objective of this step is to estimate the ego-motion at iteration $d + 1(i.e., \mathbf{w}^{d\,+\,1})$ given the latent image estimate at iteration $d(i.e., \mathbf{L}(d))$ We frame our MDF objective function in the gradient domain for faster convergence and to reduce illconditionness (Hirsch *et al.* (2011), Whyte *et al.* (2012), Xu *et al.* (2013)). We give it as

$$\mathbf{w}^{d+1} = \arg\min_{\mathbf{w}} ||\mathbf{Fw} - \nabla \mathbf{B}||_2^2 + \alpha||\mathbf{Gw}||_2^2 + \beta||\mathbf{w}||_1 \tag{3.14}$$

where the information of the gradient of **L(d)** is embedded in blur matrix **F**, ∇**B** is the gradient o **B**, and $||\mathbf{Gw}||^2||$ is the perior we introduced for RS blur. We further simplify the objective in Eq. 3.14 by separating out the sparsity prior as a constraint and taking the derivative. This yields

$$\mathbf{w}^{d+1} = \arg\min_{\mathbf{w}} ||(\mathbf{F^TF} + \alpha\mathbf{G}^T\mathbf{G})\mathbf{w} - \mathbf{F}^t\nabla\mathbf{B}||_2^2 \tag{3.15}$$

Suppose a blurred image of size $M \times N$ and n_b number of MDFs f length *l* (i.e., block size $r_b = M/n_b$). Then the dense matrix F in Eq. 3.15 is n_b times larger compared to the CCD case. This escalates the memory requirement and computational cost for RS deblurring; i.e., a naive approach to create $\mathbf{F}^T\mathbf{B}$ (with size $n_b.l \times n_b.l$) is to form a large matrix **F** of size $MN \times n_b.l$ (where $MN >> n_b$) and perform large-matrix multiplication. We avoid this problem by leveraging the block-diagonal structure of **F**, and thus for $\mathbf{F}^T\mathbf{F}$, that is specific for RS blur. The jth column of the *j*th block-matrix $\mathbf{F}_i of \mathbf{F}$ (of size $r_bN \times l$) is formed by transforming $\nabla\mathbf{L}(d)$ with the pose of $\omega_i(j)$, and vectorizing its ith block. For this, we employ the RS-EFF. Since each $\mathbf{F}_i$ can be generated independently, we bypass creating **F** , and instead directly arrive at the block-diagonal matrix $\mathbf{F}^T\mathbf{F}$ one diagonal-block at a time, with the *j*th block as $\mathbf{F}_j^T\mathbf{F}_j$. A similar operation is also done for $\mathbf{F}^T\nabla\mathbf{B}$. Since **G** is sparse, $\mathbf{G}^T\mathbf{G}$ in Eq. 3.15 can be computed efficiently Yuster and Zwick (2005)

Latent Image Estimation

Given the ego-motion at iteration $d \ + \ 1 (i.e., \mathbf{w}^{d\ +\ 1})$, this step estimates the latent image $\mathbf{L}(d \ + \ 1)$ Since ego-motion estimation is based on image gradients (Eq. 3.15), only only the latent-image gradient information needs to be correctly estimated, as pointed out in Hirsch *et al.* (2011), Whyte *et al.* (2012),Cho and Lee (2009). This eliminates the use of computationally expensive image priors in the alternative minimization step. We obtain the latent image by inverting the forward blurring process in Eq. 3.12, i.e.,

$$\mathbf{L}(d+1) \ = \ \arg\min_{\mathbf{w}} \sum_{k=1}^{R} \mathbf{C}_k^{\dagger}.\mathbb{F}^{-1}\left(\frac{1}{\mathbb{F}(a^{(k,b(k))})} \odot \mathbb{F}(\mathbf{C}_k.\mathbf{B})\right) \tag{3.16}$$

where, $(a^{(k,b(k))}$ is generated using $\mathbf{w}^{d\ +\ 1}$, $\mathbb{F}$ and $\mathbb{F}^{-1}$ are the forward and inverse DFT, respectively, and $\odot$ is a point-wise multiplication operator that also suppresses unbounded-values. We combine patches using Bartlett- Hann window that tapers to zero near the patch boundary. It has 70% overlap for patches that span adjacent blocks (to eliminate the effect of sudden changes in MDFs), and 50% for the rest. It is important to note that explicit block-wise segregation of blurred image is employed only for MDF estimation (to create $\mathbf{F}^T\mathbf{F}$ in Eq. 3.15), and and not for latent im age estimation where the estimated MDFs are utilized only to project PSFs in overlapping patches, akin to CCD-BMD Hirsch *et al.* (2011), Whyte *et al.* (2012),Hirsch *et al.* (2010). From a computational perspective, this is equiv- alent to extracting each patch of the blurred image, deconvolving it with the corresponding blur kernel (created using Eq. 3.13 with FFT acceleration, and combining the decon- volved patches to form the updated latent image. For the final iteration (in the finest level), instead of FFT inversion (as in

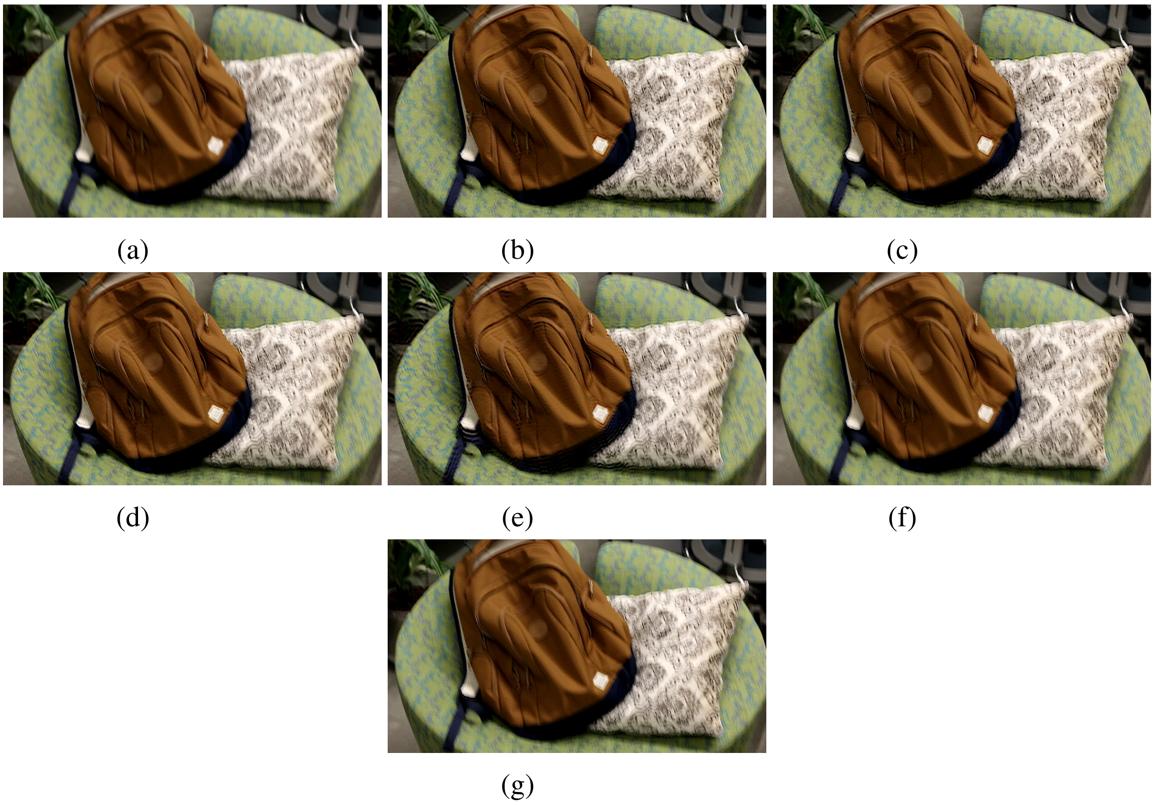
Eq. 3.16 we adopt Richardson-Lucy deconvolution Lucy. (1974) which considers natural image-priors.

CHAPTER 4

Experimental Results

4.1 Implementation Details

We implemented our algorithm in $MATLAB$. We empirically set 7 scales, each with 7 conjugate gradient iterations in kernel estimation. The blurred image in the ith scale is formed by downscaling the input image by a factor of $\sqrt{}(1/2)^{(i-1)}$, where i varies form 7 to 1. Weights for gradient data terms for image pixel values. $omega = (1/2)^q$, where q is order of operator i.e, weight for 1st order image derivatives will be $1/2$, weight for 2nd order image derivatives will be $1/4$. To start the alternative minimization, the coarsest scale MDFs are initialized with Kronecker delta. For ego-motion estimation, we consistently used the RS-prior regularization (α in Eq. 3.15) in level i as 2^{7i} (so that the RS prior can cope with the increasing image size). We have used regularization weights for latent image estimation $\alpha = 0.0005$, Regularization weight for kernel estimation $\beta = 0.5$. For latent image estimation, we used 6 patches along the shorter dimension and 8 along the longer dimension, such that each image-patch is square. For the Richardson-Lucy deconvolution, we have used 50 iteration in final step. All results shown here are obtained using number of block 3,

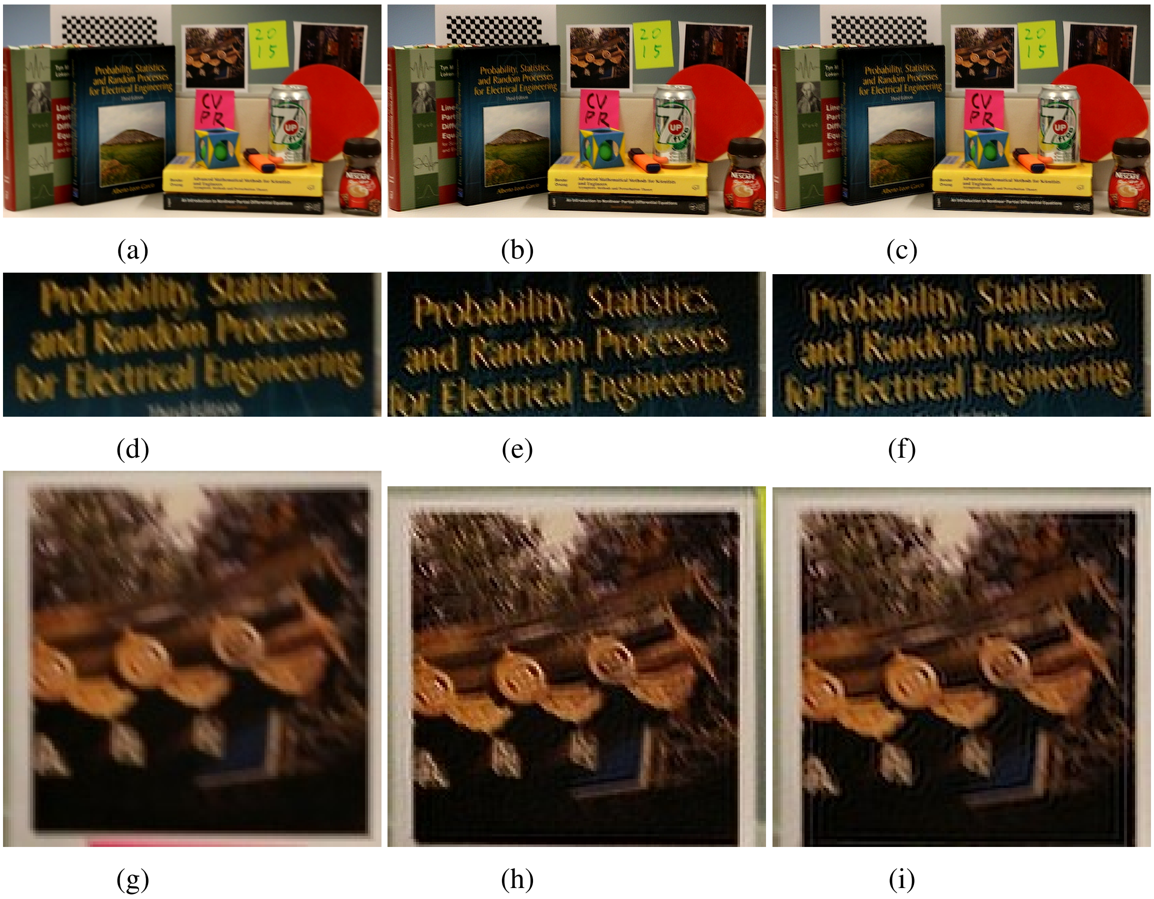


(a) (b) (c)

(d) (e) (f)

(g)

Figure 4.1: Experimental results - (a) Input RSMB distorted image (b) Output of RL deconv with nb = 3, (c) Output of direct deconv with nb = 3 (d) Output of RL deconv with nb = 2, (e) Output of direct deconv with nb = 2 (f) Output of RL deconv with nb = 1, (g) Output of direct deconv with nb = 1

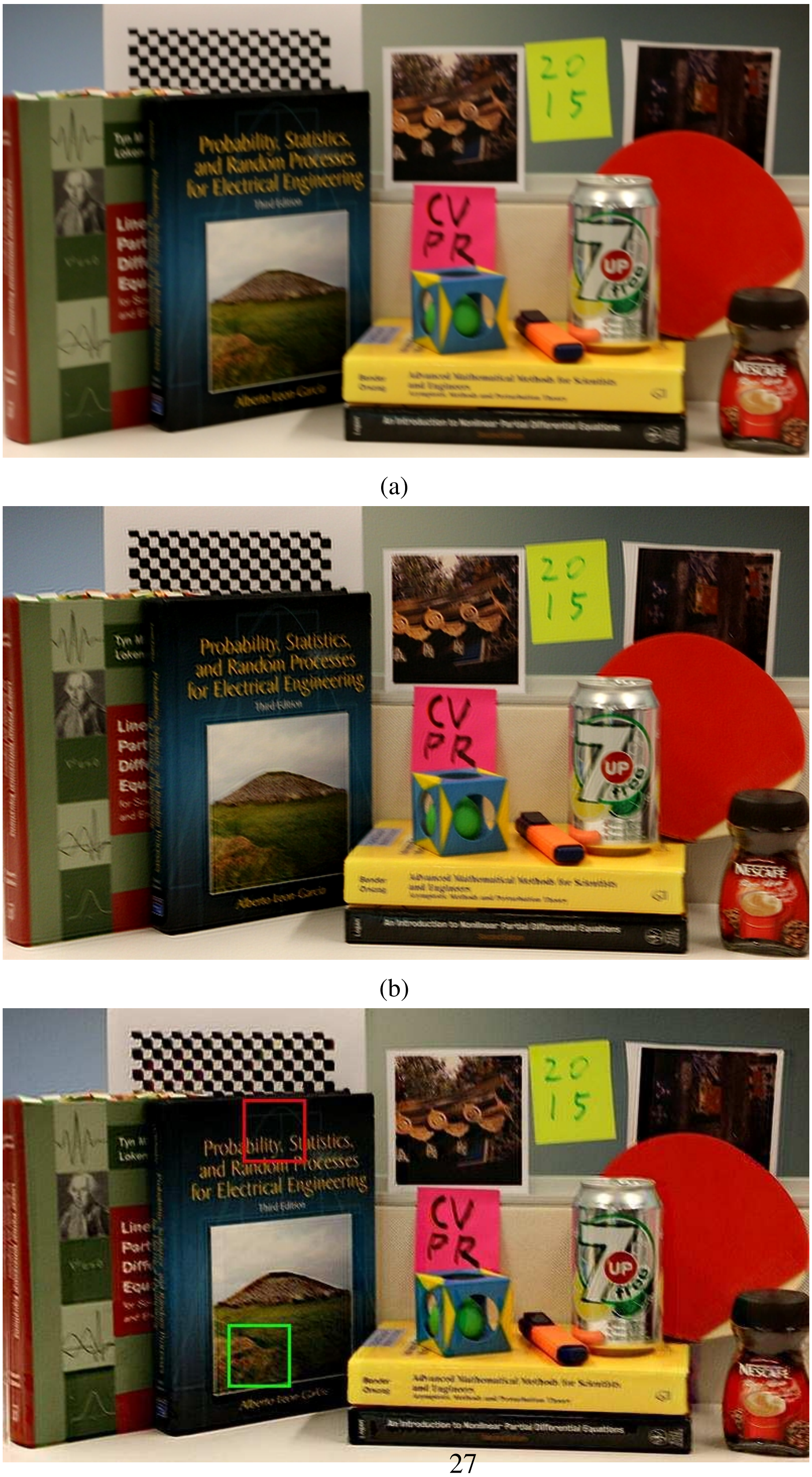


(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Figure 4.2: Experimental results - (a) Input RSMB distorted image (b) Output of RL deconv with nb = 3, (c) Output of direct deconv with nb = 3 (d)-(i) Crossponding patches

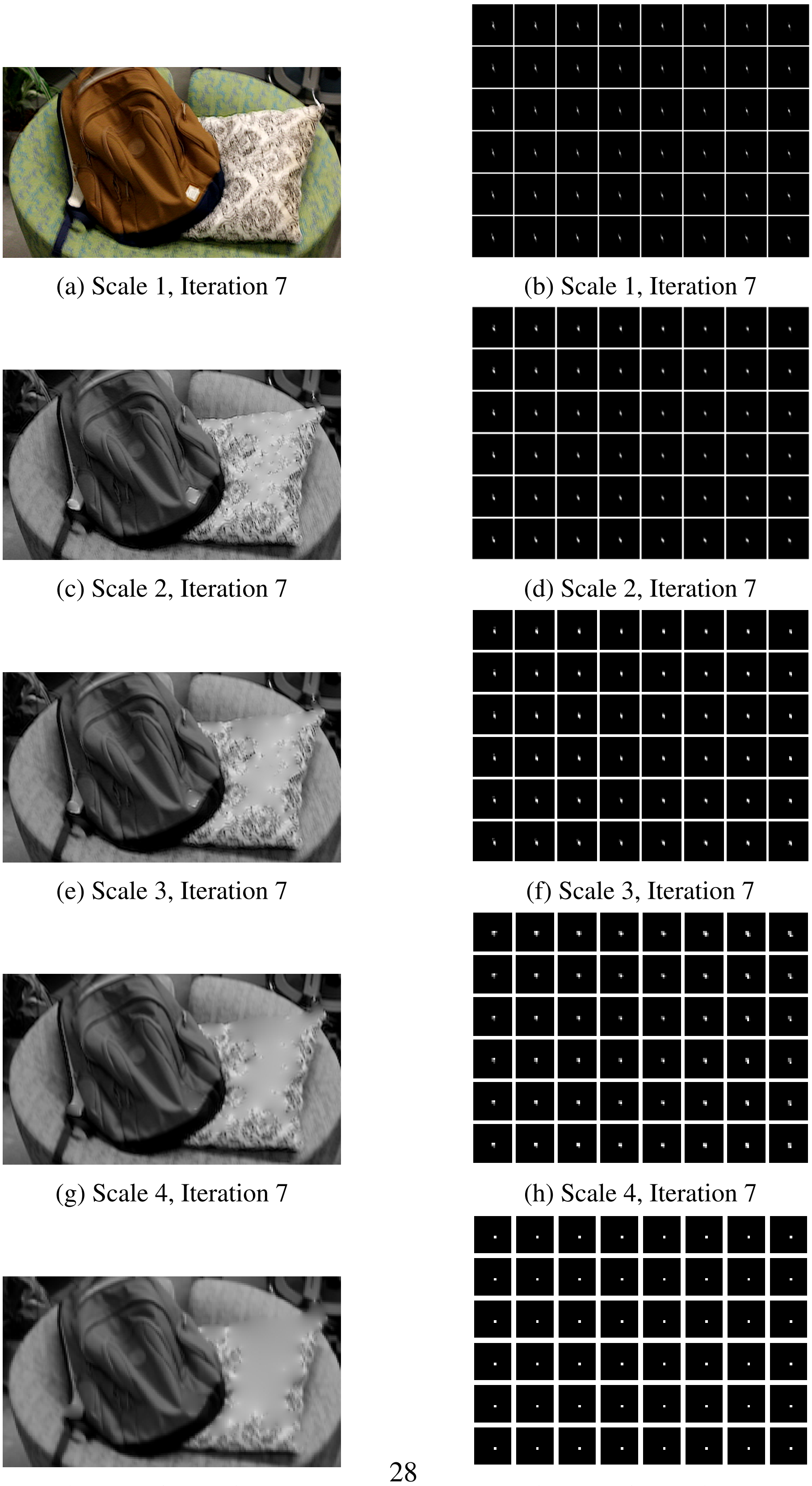


(a)

(b)

(c)

Figure 4.3: Experimental results - (a) Input RSMB distorted image (b) Our output result (c) Output from R. *et al.* (2017)



(a) Scale 1, Iteration 7

(b) Scale 1, Iteration 7

(c) Scale 2, Iteration 7

(d) Scale 2, Iteration 7

(e) Scale 3, Iteration 7

(f) Scale 3, Iteration 7

(g) Scale 4, Iteration 7

(h) Scale 4, Iteration 7

(i) Scale 5, Iteration 7

(j) Scale 5, Iteration 7

Figure 4.4: Iteration-by-iteration results of the alternative minimization of blockwise MDFs and latent image

4.2 Conclusion

In this chapter, we have shown epremental result and effect of nb on the output image. As we are suposed to reproduce result same as paper R. *et al.* (2017), we have shown our result and result from R. *et al.* (2017) paper. With cahnge in number of block, deblurred output will also change. We have noticed that after nb=3, output remain same and with heigher computational cost also increase so nb=3 is a best choice almost for all image.

REFERENCES

1. **Chakrabarti, A.**, **T. E. Zickler**, and **W. T. Freeman**, Analyzing spatially-varying blur. *In CVPR*. IEEE Computer Society, 2010. ISBN 978-1-4244-6984-0. URL `http://dblp.uni-trier.de/db/conf/cvpr/cvpr2010.html#ChakrabartiZF10`.

2. **Cho, S.** and **S. Lee** (2009). Fast motion deblurring. *In ACM Transactions on Graphics (TOG)*, **28**, 145.

3. **Couzinie-Devy, F.**, **J. Sun**, **K. Alahari**, and **J. Ponce**, Learning to estimate and remove non-uniform image blur. *In CVPR*. IEEE, 2013. URL `http://dblp.uni-trier.de/db/conf/cvpr/cvpr2013.html#Couzinie-DevySAP13`.

4. **Fergus, R.**, **B. Singh**, **A. Hertzmann**, **S. T. Roweis**, and **W. T. Freeman** (2006). Removing camera shake from a single photograph. *ACM Trans. Graph*, **25**, 787–794.

5. **Gupta, A.**, **N. Joshi**, **C. L. Zitnick**, **M. Cohen**, and **B. Curless**, Single image deblurring using motion density functions. *In Proceedings of the 11th European Conference on Computer Vision: Part I*, ECCV'10. Springer-Verlag, Berlin, Heidelberg, 2010. ISBN 3-642-15548-0, 978-3-642-15548-2. URL `http://dl.acm.org/citation.cfm?id=1886063.1886078`.

6. **Hartley, R. I.** and **A. Zisserman** (2004). Multiple view geometry in computer vision. *CPU*, **2nd**.

7. **Hirsch, M.**, **C. J. Schuler**, **S. Harmeling**, and **B. SchÃűlkopf** (2010). Efficient filter flow for space-variant multiframe blind deconvolution. *In IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 607–614.

8. **Hirsch, M.**, **C. J. Schuler**, **S. Harmeling**, and **B. SchÃűlkopf** (2011). Fast removal of non-uniform camera shake. *In IEEE International Conference on Computer Vision (ICCV)*, 463–470.

9. **Levin, A.**, Blind motion deblurring using image statistics. *In* **B. Schölkopf**, **J. C. Platt**, and **T. Hoffman** (eds.), *Advances in Neural Information Processing Systems 19*. MIT Press,

2007, 841–848. URL `http://papers.nips.cc/paper/3085-blind-motion-deblurring-using-image-statistics.pdf`.

10. **Levin, A.**, **Y. Weiss**, **F. Durand**, , and **W. T. Freeman** (2009). Understanding and evaluating blind deconvolution algorithms. *CVPR*.

11. **Lucy., L. B.** (1974). An iterative technique for the rectification of observed distributions. *The astronomical journal*, 79–745.

12. **R., M. M. M.**, **A. N. Rajagopalan**, and **G. Seetharaman** (2017). Going unconstrained with rolling shutter deblurring. *Conference on Computer Vision and Pattern Recognition (CVPR),IEEE*.

13. **Su, S.** and **W. Heidrich** (2015). Rolling shutter motion deblurring. *Conference on Computer Vision and Pattern Recognition (CVPR),IEEE*, 1529–1537.

14. **Whyte, O.**, **J. Sivic**, **A. Zisserman**, , and **J. Ponce** (2012). Non-uniform deblurring for shaken images. *International journal of computer vision (IJCV)*, **98(2)**, 168–186.

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

16. **Yuster, R.** and **U. Zwick** (2005). Fast sparse matrix multiplication. *ACM Transactions on Algorithms (TALG)*, **1(1)**, 2–13.