SINGLE AND MULTI-FRAME MOTION DEBLURRING FOR LEGGED ROBOTS: CHARACTERIZATION USING A NOVEL FD-AROC PERFORMANCE METRIC AND A COMPREHENSIVE MOTION-BLUR DATASET

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

GÖKHAN KORAY GÜLTEKIN

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF DOCTOR OF PHILOSOPHY
IN
ELECTRICAL AND ELECTRONICS ENGINEERING

SEPTEMBER 2016

Approval of the thesis:

SINGLE AND MULTI-FRAME MOTION DEBLURRING FOR LEGGED ROBOTS: CHARACTERIZATION USING A NOVEL FD-AROC PERFORMANCE METRIC AND A COMPREHENSIVE MOTION-BLUR DATASET

submitted by **GÖKHAN KORAY GÜLTEKIN** in partial fulfillment of the requirements for the degree of **Doctor of Philosophy in Electrical and Electronics Engineering Department, Middle East Technical University** by,

Prof. Dr. Gülbin Dural Ünver
Dean, Graduate School of **Natural and Applied Sciences** _______________

Prof. Dr. Gönül Turhan Sayan
Head of Department, **Electrical and Electronics Engineering** _______________

Assoc. Prof. Dr. Afşar Saranlı
Supervisor, **Electrical and Electronics Engineering** _______________

Examining Committee Members:

Prof. Dr. Gözde Bozdağı Akar
Electrical and Electronics Engineering Dept., METU _______________

Assoc. Prof. Dr. Afşar Saranlı
Electrical and Electronics Engineering Dept., METU _______________

Assoc. Prof. Dr. Alptekin Temizel
Graduate School of Informatics, METU _______________

Assoc. Prof. Dr. Pınar Duygulu Şahin
Computer Engineering Dept., Hacettepe University _______________

Assist. Prof. Dr. Onur Özcan
Mechanical Engineering Dept., Bilkent University _______________

Date: _______________

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name: GÖKHAN KORAY GÜLTEKIN

Signature :

ABSTRACT

SINGLE AND MULTI-FRAME MOTION DEBLURRING FOR LEGGED ROBOTS: CHARACTERIZATION USING A NOVEL FD-AROC PERFORMANCE METRIC AND A COMPREHENSIVE MOTION-BLUR DATASET

Gültekin, Gökhan Koray
Ph.D., Department of Electrical and Electronics Engineering
Supervisor : Assoc. Prof. Dr. Afşar Saranlı

September 2016, 110 pages

Dexterous legged robots are agile platforms that can move on variable terrain at high speeds. The locomotion of these legged platforms causes oscillations of the robot body which become more severe depending on the surface and locomotion speed. Camera sensors mounted on such platforms experience the same disturbances, hence resulting in motion blur. This is a corruption of the image and results in loss of information which in turn causes degradation or loss of important image features. Most of the studies in the literature and the proposed performance metrics focus mainly on the visual quality of motion blurred images and its improvement. However, from the perspective of computer vision algorithms, feature detection performance is an essential factor that determines their performance. The aim of this study is to analyze and evaluate motion blur on a legged robot and the deblurring methods with a focus on feature detection. We propose a multi-frame motion deblurring method utilizing the variable motion blur in consecutive image frames captured from the camera on a legged mobile robot. For a comparison of blurred and deblurred images, we define a novel performance metric based on the feature detection accuracy. Noting that a suitable data set to evaluate the effects of motion blur and its compensation for legged platforms is lacking in the literature, we develop a comprehensive multi-sensor data set for that purpose. The data set consists of monocular image sequences collected in synchronization with a low cost MEMS gyroscope, an accurate fiber optic gyroscope and an externally measured ground truth motion data. We make use of this data set for an extensive benchmarking of prominent motion deblurring methods from the literature in terms of the proposed feature based metric.

Keywords: Motion Blur, Motion Deblurring, Corner Feature Detection, Legged Robots

ÖZ

BACAKLI ROBOTLARDA HAREKET BULANIKLIĞINI TEKİL VE ÇOĞUL KARELER KULLANARAK GİDERME: ÖZGÜN FD-AROC BAŞARIM ÖLÇÜTÜ VE KAPSAMLI BİR HAREKET BULANIKLIĞI VERİ KÜMESİ İLE NİTELEME

Gültekin, Gökhan Koray
Doktora, Elektrik ve Elektronik Mühendisliği Bölümü
Tez Yöneticisi : Doç. Dr. Afşar Saranlı

Eylül 2016 , 110 sayfa

Kabiliyetli bacaklı robotlar, yüksek hızlarda değişken arazilerde kıvrak bir şekilde hareket edebilen yapılardır. Bu yapıların hareketleri, hareket hızı ve yüzeye bağlı olarak robot gövdesinde ciddi salınımlara neden olmaktadır. Bu yapılara monte edilmiş olan kamera algılayıcıları da aynı bozanetkenlere maruz kalmakta ve dolayısıyla hareket bulanıklığına yol açmaktadır. Bu resimdeki bir bozulmadır, bilgi kaybına ve resimdeki önemli öznitelik noktalarının kaybolmasına yol açmaktadır. Literatürdeki birçok çalışma ve önerilen başarım ölçütleri genel olarak hareket bulanıklığına maruz kalmış olan resimlerin görsel kalitesi ve bunun artırılması üzerine odaklanmaktadır. Ancak, özniteliklerin tespit başarımı, bilgisayarla görme algoritmalarının başarımındaki önemli bir etkendir. Bu çalışmanın amacı, bacaklı bir robot üzerindeki hareket bulanıklığının analizi ve bulanıklık giderme yöntemlerinin öznitelik tespit başarımına odaklı olarak değerlendirilmesidir. Bacaklı bir robot üzerinden alınan ardışık resim karelerindeki farklı hareket bulanıklığından yararlanılarak çok kareli bir hareket bulanıklığı giderme yöntemi önerilmiştir. Hareket bulanıklığı ve giderilmesinin ölçümünde, öznitelik tespit başarımı temelinde yeni bir başarım ölçütü önerilmektedir. Literatürde halihazırda, bacaklı robotların hareketinden kaynaklanan hareket bulanıklığı ve bunun giderilmesi için kullanılabilecek bir veri kümesi bulunmaması nedeniyle, çoğul algılayıcı verisi içeren kapsamlı bir veri kümesi oluşturulmuştur. Veri kümesi, tekgözlü kamera resim dizileri ile eş zamanlı olarak toplanmış olan düşük maliyetli MEMS dönüölçer, hassas bir fiberoptik dönüölçer ve harici olarak ölçülmüş olan referans hareket verisinden oluşmaktadır. Bu veri kümesi kullanılarak, literatürde öne çıkan hareket bulanıklığı giderme yöntemlerinin, önerilen öznitelik temelli ölçüt ile kapsamlı olarak kıyaslaması yapılmıştır.

Anahtar Kelimeler: Hareket Bulanıklığı, Hareket Bulanıklığının Giderilmesi, Köşe özniteliklerinin Bulunması, Bacaklı Robotlar

To my parents and little brother

Mahmut, Elife & Korcan Emre Gültekin

ACKNOWLEDGMENTS

I would like to express my deep appreciation and sincere gratitude to my supervisor Dr. Afşar Saranlı for his leading guidance, encouragement, and continuous support from beginning to the end of my PhD studies. His suggestions during our research meetings played a tremendous role in helping me to broaden my view and knowledge. He showed me different ways to approach a research problem and the need to be persistent to accomplish any goal. He let me take part in the SensoRHex project which was a great opportunity for me to extend my academic and technical skills. He also made the Rolab (Laboratory of Robotics and Autonomous Systems) a wonderful workplace which led this complicated study to end up successfully.

I wish to express my deep sense of gratitude to Dr. Uluç Saranlı for his guidance and being a source of inspiration for me in the conduct of my thesis work and our research project. I also would like to thank Dr. Kemal Leblebicioğlu for his leading advices. I should extend my sincere thanks to Dr. Emre Tuna for his helps on some of the formulations.

I would like to thank to the professors of my thesis committee; Dr. Gözde Bozdağı Akar and Dr. Alptekin Temizel for providing valuable feedback for the progress of this work. I also thank to my PhD jury members, Dr. Pınar Duygulu Şahin and Dr. Onur Özcan for their suggestions that help us to improve our work even more.

I literally consider myself a lucky person to work with the amazing group of people in Rolab. I am very thankful to Dr. Mert Ankaralı, Dr. Orkun Öğücü, Emre Ege, Ferit Üzer, Mehmet Mutlu, Deniz Duyul, Ünver Akmandor and all other members for sharing me their time and knowledge. Further thanks should also go to my dear friends Dr. Feza Carlak, Maksat Atagoziyev, Onur Çetin, Kadir Üstün, İlker Gürcan, Erdal Epçaçan, İsmail Uyanık, Ali Nail İnal, Ömür Arslan, Tolga Özaslan and Çağlar Seylan.

I would like to thank the Scientific and Technological Research Council of Turkey (TÜBİTAK) for awarding me their prestigious Doctor of Philosophy studies scholarship (2211) and supporting this study partially as part of the TÜBİTAK-1001 project granted with ID:110E120.

Finally, yet the most importantly, nothing is adequate to express my heartfelt feelings to my beloved family forever. None of this would have been even possible without the love and patience of them. I owe a great many thanks to my loving mother(Elif Gültekin), my dear father(Mahmut Gültekin) and my beloved brother(Korcan Emre

Gültekin) for their undying love, unconditional support, encouragement and their trust in me.

I apologize from the people who I should also have included their names here for their contributions but may forgot because of the rush I am in. I send my thanks to all individually.

TABLE OF CONTENTS

ABSTRACT v

ÖZ vii

ACKNOWLEDGMENTS x

TABLE OF CONTENTS xii

LIST OF TABLES xv

LIST OF FIGURES xvi

LIST OF ABBREVIATIONS xxiii

CHAPTERS

1 INTRODUCTION 1

  1.1 Literature Survey 4

  1.2 Contributions 16

  1.3 Outline of the Thesis 17

2 PROPOSAL OF A MULTI-IMAGE MOTION DEBLURRING METHOD FOR APPLICATION TO A LEGGED MOBILE ROBOTIC PLATFORM 19

  2.1 Theory of Motion Blur Formation and the Inverse Problem . 19

    2.1.1 Motion Blur Model as a Convolution 19

2.1.2 Point Spread Function 20

2.1.3 Inverse Filtering 21

2.2 Multi-Image Motion Deblurring 22

2.2.1 Zero Response Frequencies of a Linear Motion Blur PSF 24

2.3 Motion Blur in Legged Mobile Robotic Platforms 31

2.3.1 IMU Based 6-D Spatially Varying Motion Blur PSF Computation and Simulation 32

2.4 Estimation of 2-D Nonlinear Point Spread Function 35

3 MOTION BLUR ASSESSMENT AND FEATURE BASED IMAGE QUALITY METRIC FOR COMPUTER VISION APPLICATIONS . 39

3.1 Available Metrics for Image Quality 39

3.2 Available Metrics for Image Sharpness 42

3.3 Proposed Standard Deviation Based Motion Blur Measurement of an Image Within an Image Sequence 45

3.3.1 Justification of Standard Deviation Based Motion Blur Measurement on a Legged Robot Image Sequence 49

3.4 Proposed Feature Detection Based Image Quality Metric for Computer Vision Applications 52

3.4.1 "Receiver Operating Characteristic" for Feature Detection Performance 53

3.4.2 Justification of Feature Based Metric 55

4 A COMPREHENSIVE MOTION BLUR DATASET FOR LEGGED ROBOTICS 59

4.1 Motivation and Scope of a New Motion Blur Dataset 59

4.2 1-D, Linear, Spatially Invariant Motion Blur Dataset 60

4.3 1-D Spatially Varying Motion Blur Dataset 63

4.4 Multi-Sensor Motion Blur Dataset for Legged Mobile Robotic Vision Applications 64

4.5 Offline Synchronization of Motion Data & Image Data 71

4.5.1 Optical Flow Computation From the Motion Data . 72

4.5.2 Optical Flow Computation From the Image Frames 74

4.5.3 Synchronization Using Optical Flow From Images & Camera Motion 75

5 EXPERIMENTS AND TEST RESULTS ON MOTION DEBLURRING STUDIES 79

5.1 Testing and Benchmark Comparison of Single Frame Motion Deblurring Methods 79

5.2 Multi-Frame Motion Deblurring Results 91

5.3 The Effect of Motion Blur and Deblurring on Feature Descriptors 94

5.4 Computational Complexities of the Tested Motion Deblurring Methods 96

6 CONCLUSIONS AND FUTURE STUDIES 99

6.1 Conclusions 99

6.2 Future Work 100

REFERENCES 103

APPENDICES

CURRICULUM VITAE 109

LIST OF TABLES

TABLES

Table 3.1 Comparison of three frequently used image quality metrics and FD-AROC. 57

Table 4.1 Properties of the camera speed controlled 1D linear, uniform motion blur dataset experiments. 63

Table 4.2 The sensors used in dataset and their properties. 65

Table 4.3 Properties of the legged robot motion blur dataset experiments. . . . 71

Table 5.1 Metric scores of motion deblurring methods with Harris corner detector. 89

Table 5.2 Metric scores of motion deblurring methods with FAST corner detector. 90

Table 5.3 Number of SIFT keypoints that can be detected on sharp, blurred(No-DB) and deblurred images with the given motion deblurring methods. . . . 95

Table 5.4 The mean error of matching SIFT descriptor vectors on blurred and deblurred images with the given motion deblurring methods. 96

Table 5.5 The cornerness scores of sharp, blurred and deblurred checkerboard images with the given motion deblurring methods. 97

Table 5.6 The computation times required for the tested motion deblurring methods to run on 1280x960 resolution images. 97

LIST OF FIGURES

FIGURES

Figure 1.1 SensoRHex legged mobile robot platform 1

Figure 1.2 An example motion blurred image frame captured from the camera on the SensoRHex robot while it is walking on flat laboratory ground surface. 4

Figure 1.3 Camera stabilization platform designed for SensoRHex robot. The mechanism has three linear motors and has the capability to rotate around 3 axes (roll, pitch and yaw) 5

Figure 1.4 (Left) The robot eye mechanism used for motion deblurring in [23]. (Right) Wheeled mobile robot with a side looking camera that is used for coded exposure based motion deblurring in [38]. 6

Figure 1.5 A box filter in time domain and corresponding sinc function in frequency domain which corresponds to a linear PSF. The zero response frequencies of the sinc function prevents its invertibility. 7

Figure 1.6 (Left) A typical real-world scene. (Right) The empirical distribution of gradients in the scene (blue), along with a Gaussian fit (cyan), a Laplacian fit (red) and a Hyper-Laplacian with $\alpha = 2/3$ (green) [25]. . . . 8

Figure 1.7 The steps of motion deblurring method proposed in [40]. They only manipulate the capturing part to get an invertible motion blur PSF. . . 9

Figure 1.8 The prototype camera setup with fluttered shutter mechanism and its control circuit [40]. 10

Figure 1.9 The DFT of the box filter (flat blur) is a sinc function and frequencies with sharp dips are lost. The MURA code filter is better than box filter but the variance of the frequency magnitudes is high compared to their proposed code. (b) The DFT of decoding filters show that box and MURA filters require significant amplification of certain high frequencies. [40] 10

Figure 1.10 The comparisons of obtained results. a) A short exposure photo is shown in a low light environment and its log intensity scaled form is given at (e). b) An image obtained with traditional continuous exposure with 200ms duration is shown. c) Fluttered shutter exposure using MURA code. d) Fluttered shutter exposure using proposed code. f,g,h) Deblurred results [40]. 11

Figure 1.11 Hybrid camera setup proposed by [6]. The primary camera has a high resolution and low frame rate whereas the secondary camera has a low resolution and high frame rate. 12

Figure 1.12 Noninvertible PSFs Hi(w) each having zeros and the combined operator P which does not have zeros shown on left. Comparison with coded exposure shown on right [2]. 13

Figure 1.13 Comparison of varying exposure, traditional video and coded exposure methods. Varying exposure method achieves similar performance to the coded exposure method and a better performance than the traditional video [2]. 14

Figure 1.14 The motion blurred frames captured with a static camera, parabolic motion camera and fluttered shutter camera. Their deblurring outputs are given on the bottom images [26]. 15

Figure 1.15 First figure shows the range of motion for 1D case proposed by [26] and the second figure shows the range of motion for the proposed 2D case by [3]. Third figure shows the circular motion of the sensor. The sensor is translated circularly about the optical axis. Last figure shows the sensor motion trajectory (red curve) in the space-time volume. 15

Figure 1.16 Experimental setup used in [24]. (Left) Light-weight structure with reflective spherical markers for recording camera shake with a Vicon system. (Right) Camera attached to a high-precision Stewart platform to take pictures with played back camera shakes in a controlled setting. 16

Figure 2.1 Frequency responses of 1-D linear PSFs with different blur lengths. 23

Figure 2.2 (Left) Frequency responses of two different PSFs. (Right) Illustration of the windowed filtering of PSFs based on maximum frequency responses. Dotted line shows the equivalent combined filter frequency response. 24

Figure 2.3 Frequency response graph of two PSFs whose magnitudes (8 and 16) are integer multiple of one another. Graph shows the common and individual zero response frequencies as calculated above. 29

Figure 2.4 Frequency response graph of two PSFs whose magnitudes (10 and 25) are not integer multiple of each other but have at least one common divisor. Graph shows the common and individual zero response frequencies as calculated above. 30

Figure 2.5 Frequency response graph of two PSFs whose magnitudes (9 and 11) are relatively prime with each other. Graph shows the common and individual zero response frequencies as calculated above. 31

Figure 2.6 Coordinate axes definition for camera motion. Image plane is placed on a plane parallel to the x-y plane and with a perpendicular distance f from the origin. Optical axis is along the z axis. 32

Figure 2.7 The resultant velocity field when translation along x and y axis are applied is shown on top graph and translation along z axis is shown on bottom graph. The PSFs corresponding to these motions are obtained and used to simulate the motion blur. The resultant simulated images are given next to the velocity graphs. Note that, the latter figure has spatially varying motion blur PSF. 34

Figure 2.8 The resultant velocity field when rotation around either of pitch, yaw and roll axes are applied is shown top, center and bottom figures respectively. The PSFs corresponding to these motions are obtained and used to simulate the motion blur. The resultant simulated images are given next to the velocity graphs. Note that, all the figures have spatially varying motion blur PSFs, however the effect is seen dramatically on the bottom figure. 36

Figure 2.9 (Top) 2-D nonlinear pixel motion trajectory using a set of motion data which are projected onto image plane using (2.50). (Bottom) The corresponding 2-D nonlinear PSF. 37

Figure 3.1 Blurring quantity of an image sequence captured from the camera on the SensoRHex robot using the variance measure. 43

Figure 3.2 Blurring quantity of an image sequence captured from the camera on the SensoRHex robot using the L1 norm of the 1st order image derivatives as a measure. 44

Figure 3.3 Blurring quantity of an image sequence captured from the camera on the SensoRHex robot using the L2 norm of the 1st order image derivatives as a measure. 45

Figure 3.4 Blurring quantity of an image sequence captured from the camera on the SensoRHex robot using the L1 norm of the image Laplacian as a measure. 46

Figure 3.5 Variation on blur magnitude (pixels) versus standard deviation of the synthetically generated checker board pattern. 47

Figure 3.6 Flow chart of the proposed motion blur detection method. 49

Figure 3.7 Standard deviation vs. time of the 50 consecutive images captured from a camera mounted on a legged mobile robot platform. 50

Figure 3.8 First randomly chosen image taken from camera mounted on a legged robot platform and its blown-up version. The experimental blur magnitude is calculated as 7.9 pixels while the ground truth value is 8.9 pixels. 50

Figure 3.9 Second randomly chosen image taken from camera mounted on a legged robot platform and its blown-up version. The experimental blur amount value is 1.69 pixels while the original value is 2.23 pixels. 51

Figure 3.10 Comparison of experimental and real values of blur magnitude for a randomly selected image frame index which are taken from images captured from a camera mounted on a legged mobile robot platform. 52

Figure 3.11 (a)Corners detected on sharp image. (b)Corners detected on blurred image. (c)Corners detected on deblurred image. 53

Figure 3.12 An example ROC graph of three arbitrary detectors. The one represented with blue star markers has the highest performance, second one is the green plusses and the red circles are the worst. 56

Figure 4.1 Accurate speed and position controlled one dimensional translating stage with a Pointgrey Flea2 camera mounted on top. This stage is used to demonstrate controllable camera speed resulting in controllable motion blur (i.e. PSF). 61

Figure 4.2 The velocity profile of the speed controlled setup when a 10cm/sec step input speed is applied. The graph on the right shows the magnification of ripples. 61

Figure 4.3 Images obtained by the controlled camera motion setup. The exposure is kept constant at 100ms for all images. The four images are all captured at the same cart position while the cart is moving at different speeds. The cart speeds are 2cm/sec, 5cm/sec, 7cm/sec and 11cm/sec for images numbered from (a) to (d) respectively. 62

Figure 4.4 The scenes of office(top), pebble stone(bottom left) and prismatic objects(bottom right). 64

Figure 4.5 The scenes of "dry leaves"(upper left), "green leaves"(upper right), "Forest Ground"(lower left) and "pine cones"(lower right). 65

Figure 4.6 The setup for obtaining spatially varying motion blur caused by the zooming effect. The camera is rotated 90 degrees clockwise around its yaw axis and attached to the moving cart. The cart is moved from far to near the checkerboard pattern at different speeds to simulate zooming based motion blurring effect. 66

Figure 4.7 Spatially varying motion blurred images obtained by the above setup. The exposure is kept constant at 100ms for all images. The images are captured at same cart speed of 13cm/sec with 2cm distance intervals each during the approaching motion. 66

Figure 4.8 The general overview of the experiment area. There are 6 OptiTrack cameras mounted on the ceiling. The checkerboard pattern is placed vertical to the ground plane and the optical axis of the camera. Three PCs are used; one is for controlling of the robot; one of them is used for real-time data capturing from the sensors; and the other one is optitrack server which computes 6D position & orientation and sends packets via UDP protocol through network to the data capturing PC. 67

Figure 4.9 The PointGrey Flea2 camera (models 14s3c and 08s2c), MicroStrain IMU, OptiTrack s250e motion tracking camera and its reflective marker are shown respectively 67

Figure 4.10 (Top) Camera mounting part for mounting both to the FOG, IMU and the body frame of SensoRHex. (Bottom) Mounting structure for assembling FOG and IMU together. The whole structure is also used for mounting to the camera. 68

Figure 4.11 The manufactured mounting parts and assembly of the camera, FOG and IMU. The whole structure is designed such that it can also be assembled to the body frame of SensoRHex with a couple of auxiliary parts that are also designed and 3D printed. 69

Figure 4.12 The camera, FOG & IMU structure assembled onto the body frame of SensoRHex. The camera is accurately placed on the center of the front frame. 11 reflective markers are rigidly mounted asymmetrically to the various places of the body frame via 3D printed mounting parts. 70

Figure 4.13 Sample image frames taken from the SensoRHex motion blur dataset together with their magnified views below them. 71

Figure 4.14 The graphical user interface (GUI) designed for manual data analysis of multi-sensor motion blur data set. The complete rotational motion data are shown together with the image frame selected by clicking on the desired time instance. The motion data that correspond to the frame capturing period of selected frame is also plotted on the small figure windows seen on the left side. A preview of homogenously selected frames from the whole sequence is displayed for a quick overview. It is also possible to zoom in and out to the motion data and the image frames. 72

Figure 4.15 An example motion blurred frame (frame:193 at t=13.79s) taken from the collected dataset and the corresponding optical flow vectors computed from the gyro data. 73

Figure 4.16 Euler angles from OptiTrack, FOG and IMU. Gyro based Euler angles are computed by the integration of gyro rates. The two red bars shows the frame capturing period (13.79s to 14.46s) of the example frame (f:193) shown in Fig. 9. Small figure on the left shows the gyro rates taken during the frame capturing period of the selected frame. 74

Figure 4.17 Euler rates captured from the FOG, IMU and OptiTrack. Small figure on the left shows the gyro rates taken during the frame capturing period of the example frame (f:193) shown in Fig. 9 75

Figure 4.18 Selected consecutive motion blurred image frames for optical flow computation and the corresponding optical flow vector field computed by the method given in [32]. 75

Figure 4.19 Optical flow averages computed from gyro versus image sequences. The starting of motion can easily be identified from both curves. It is observed that the gyro and camera data are not synchronized. 76

Figure 4.20 Optical flow averages computed from gyro versus image sequences. The gyro and camera data are synchronized by adding the required time shift and adjusting the actual frame rate of camera from 15 fps to 14 fps. . 77

Figure 5.1 An example motion blurred frame from Sp02 dataset and its zoomed in view. 80

Figure 5.2 ROC analysis of Harris(left) and FAST(right) feature detectors under different motion deblurring methods using the linear uniform 20 pixel motion blurred images in Checkerboard dataset. 80

Figure 5.3 ROC analysis of Harris(left) and FAST(right) feature detectors under different motion deblurring methods using the linear uniform 20 pixel motion blurred images in Office Scene dataset. 81

Figure 5.4 An example motion blurred frame from Sp05 dataset and its zoomed in view. 81

Figure 5.5 ROC analysis of Harris(left) and FAST(right) feature detectors under different motion deblurring methods using the linear uniform 50 pixel motion blurred images in Checkerboard dataset. 82

Figure 5.6 ROC analysis of Harris(left) and FAST(right) feature detectors under different motion deblurring methods using the linear uniform 50 pixel motion blurred images in Office Scene dataset. 82

Figure 5.7 ROC analysis of Harris(left) and FAST(right) feature detectors under different motion deblurring methods using the SensoRHex motion blur dataset. 83

Figure 5.8 (a) 20 pixels motion blurred checkerboard image. (b-h) Deblurring results obtained using the methods written on top of the frames given above. 85

Figure 5.9 (a) 50 pixels motion blurred checkerboard image. (b-h) Deblurring results obtained using the methods written on top of the frames given above. 86

Figure 5.10 (a) 20 pixels motion blurred Office Scene image. (b-h) Deblurring results obtained using the methods written on top of the frames given above. 87

Figure 5.11 (a) 50 pixels motion blurred Office Scene image. (b-h) Deblurring results obtained using the methods written on top of the frames given above. 88

Figure 5.12 ROC analysis of Harris(left) and FAST(right) feature detectors under multi-frame deblurring method, best performing single frame motion deblurring method and no deblurring applied images in the linear uniform 20 pixel motion blur dataset. 92

Figure 5.13 ROC analysis of Harris(left) and FAST(right) feature detectors under multi-frame deblurring method, best performing single frame motion deblurring method and no deblurring applied images in the linear uniform 50 pixel motion blur dataset. 93

Figure 5.14 ROC analysis of Harris(left) and FAST(right) feature detectors under multi-frame deblurring method(with linear and nonlinear PSF estimation), two of the best performing single frame motion deblurring methods and no deblurring applied images in the SensoRHex motion blur dataset. . 94

LIST OF ABBREVIATIONS

ROC	Receiver Operating Characteristic
AROC	Area Under ROC
FD-ROC	Feature Detection Based Receiver Operating Characteristic
FD-AROC	Feature Detection Based Area Under ROC
FOG	Fiber Optic Gyroscope
IMU	Inertial Measurement Unit
PSF	Point Spread Function
DB	Deblurring
SID	Single Image Deblurring
MID	Multi Image Deblurring
MAE	Mean Absolute Error
MSE	Mean Square Error
NRMSE	Normalized Root Mean Squared Error
PSNR	Peak Signal to Noise Ratio
VSNR	Visual Signal to Noise Ratio
SSIM	Structural Similarity Index
UQI	Universal Quality Measure
PQR	Picture Quality Rating
MOS	Mean Opinion Score
VIF	Visual Information Fidelity
UDP	User Datagram Protocol
TP	True Positives
FP	False Positives
TN	True Negatives
FN	False Negatives
TPR	True Positive Rate
FPR	False Positive Rate
MEMS	Micro Electro-Mechanical Systems

MURA	Modified Uniform Redundant Arrays
FAST	Features From Accelerated Segment Test
SIFT	Scale Invariant Feature Transform
SURF	Speeded Up Robust Features

CHAPTER 1

INTRODUCTION

Hexapod robots are highly mobile robotic platforms that can move on very rough terrain surfaces with high speeds. The nature of these legged platforms causes high mechanical vibrations on the robot body even during locomotion on a flat surface. Moreover, if it moves on a rough terrain with high speed, these vibrations increases even more. One such family of robots with outstanding rugged terrain locomotion performance share the RHex morphology [46]. Our work utilizes the SensoRHex experimental hexapod platform, which preserves the morphology but is a hardware and software improvement on the original RHex design. Fig. 1.1 shows the SensoRHex robot with and without its skin.

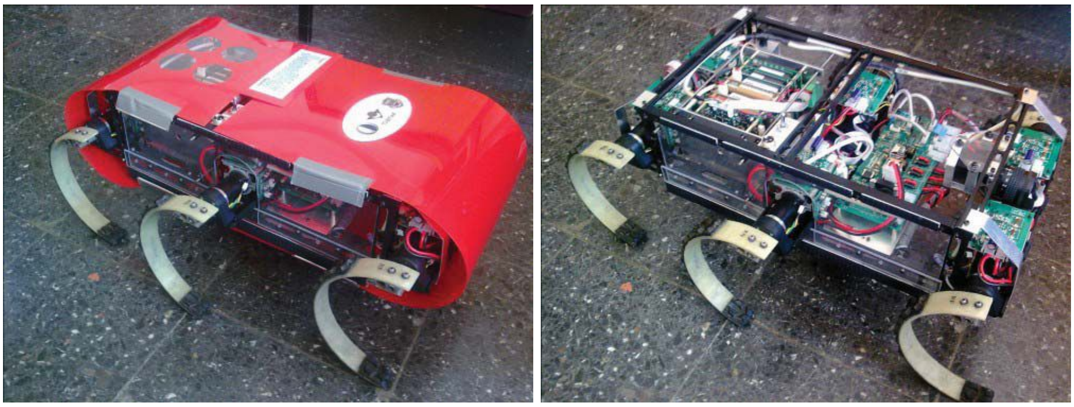


Figure 1.1: SensoRHex legged mobile robot platform

During its operation in an environment, the robot requires to gather some information about the state of itself and the surroundings. Among various sensors, cameras are very commonly utilized since vision provides a rich variety of information by the use of image processing techniques available. For this purpose, there exists a camera mounted to the front of our legged robot platform SensoRHex as seen in Fig. 1.1.

An ideal image is the snapshot of a scene and every point in the scene corresponds to a single point on the image. However, the cameras are not capable of capturing the image of a scene in an instant of time. The measuring element of the camera (mostly image sensors in digital cameras) requires enough light to fall on them to be able to measure the scene brightness with enough accuracy. Therefore, the measuring element should be exposed to the light rays coming from the scene for some time period. During this time period, the scene and the camera should be standing still to be able to capture a sharp image of the scene. If this assumption does not hold, then each point on the scene does not fall onto a single point on the image but spreads into different locations causing a motion blur to appear in the captured image. Motion blur is defined as the apparent streaking of rapidly moving objects in a still image. It is caused by the relative motion between the camera and the scene during the exposure period of the camera. If the frequency spectrum of a sharp image (having negligibly small or no motion blur) is analyzed it can be observed that it includes high frequency components. However, when this image is exposed to motion blur, it is observed that motion blur acts like a lowpass filter and filters high frequency components out. Although there are studies that intentionally add motion blur to generate visually pleasant videos or utilize the motion blur effect (e.g. for generating super-resolution image or to gather information about the motion, etc.), it is often an unwanted effect that degrades the performance of most of the computer vision algorithms. Therefore, it is one of the major problems in computer vision literature to eliminate or decrease the effect of motion blur in images and video sequences.

The camera mounted on a legged robot which is moving with a high speed and on a rough terrain is mostly exposed to high mechanical vibrations. These vibrations during a video capture prevent obtaining sharp images from the camera. This results in a corruption or loss of information by means of a blurring on the image. An example image captured during the walking motion of a hexapod robot that includes motion blur is shown in Fig. 1.2. As can be noticed easily from the figure, the captured image is neither visually pleasant for human perception for remote controlling by a human operator nor suitable for gathering high frequency components of scene such as edges, corners etc.

Computer vision often relies on image processing algorithms that can extract so called *image features* that are meaningful for higher level visual processing and image understanding. Widely used image features include those that are suitable for human interpretation such as corners, lines and angles in the image as well as important features that are multi-dimensional and that do not carry such direct meaning. Such features include the popular Harris Corner Detector [16], Features from Accelerated Segment Test (FAST) [44, 45], Scale-Invariant Feature Transform (SIFT) [45] and Speeded Up Robust Features (SURF) [5]. The non-uniform degradation of the image and the loss of high-frequency components have a significant impact on the reliable extraction of these features and hence the performance of subsequent image analysis.

The recovery of the original image content from the motion-blur corrupted version has important applications. More traditionally, these applications involved improving visual appearance of the image such as in digital photography. A more recent focus is on computer vision and mobile robot vision in particular. Legged robotics is a particularly challenging area because of ever-presence of severe camera motion. It is reasonable to expect that compensation of motion-blur with the aim of partially recovering the original image data would also improve the performance of the aforementioned feature detection methods. Such compensation based on a single motion-blurred frame which essentially is an inverse filtering problem, unfortunately is an ill-posed problem. The zeros of the "blur filter" results in a loss of image information that cannot be recovered. Single-frame deblurring methods, the focus of the present paper, concentrate on using apriori information and make assumptions in order to reconstruct the image with minimal loss and processing artifacts. Measuring the performance of these reconstruction algorithms requires performance measures that can quantify the success of the algorithm with respect to a given application. The literature discuss various such "performance metrics" but the present paper argues that these are mostly focused on human perceptual improvement and/or raw distortion in the image and are not necessarily well suited for automated computer vision applications since they do accurately reflect the performance of the feature extraction process.

Secondly, the benchmarking of motion deblurring methods requires synthetic and/or real test data with reliable ground truth information. There are many studies in the literature on motion blur and methods of deblurring. Some of these, namely *blind methods* require only motion blurred image(s) whereas *informed* methods require both the motion blurred image(s) as well as information about the corresponding motion and/or the resultant blurring function (also called the point-spread function-PSF). Proposed methods are generally tested with a custom generated limited test set which often may not fully disclose the performance which can be highly dependent on the chosen data. To be able to make a fair comparison of the methods, they should be tested on the same data set which covers scenarios of the application area. For example, for the application area of legged robotic platforms, the test data set collected from a wheeled vehicle may not reflect the true performance of tested algorithms on the intended application area. Legged locomotion platforms, including robotic ones have some particular motion characteristics that are reflected to the resulting motion blur. This effects algorithm performance but also, may sometimes be exploited for the development of application specific optimizations.

Although there are many studies and data sets available on motion blur, to the best of our knowledge, there is no multi-modal data set that can be used to evaluate the performances of blind and non-blind motion deblurring algorithms within the legged robotics application area. One of the aims of this study was to build a comprehensive motion blur data set, composed of multiple on-board inertial sensors supplementing the camera sensor as well as synchronized data from external sensors for ground truth information. It is our hope that this data set is not only suitable for the objective comparison of current and future algorithms on motion deblurring for legged mobile robotic but also for other vision research for cameras mounted on legged platform.

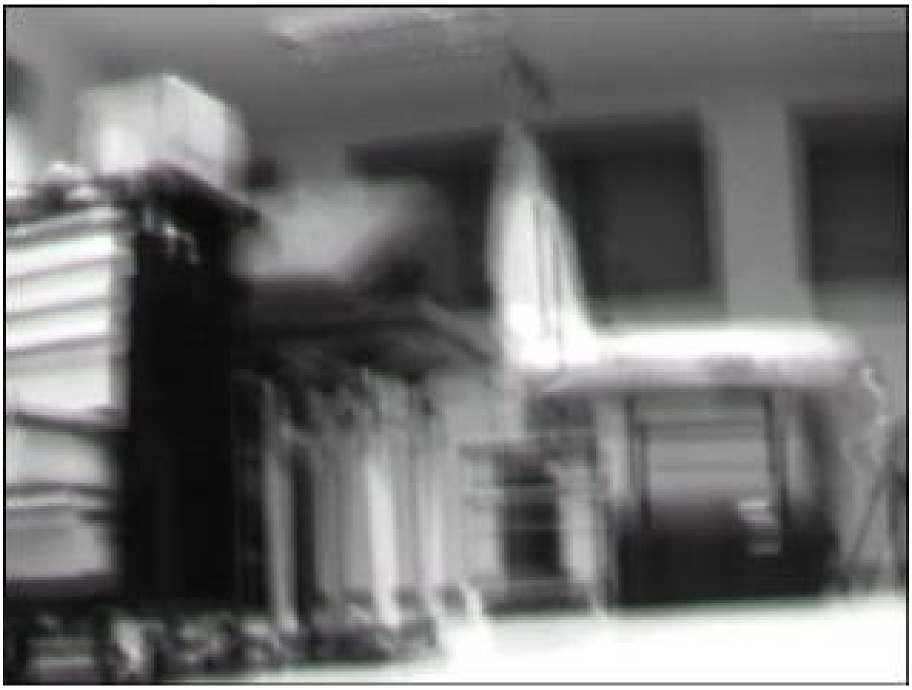


Figure 1.2: An example motion blurred image frame captured from the camera on the SensoRHex robot while it is walking on flat laboratory ground surface.

1.1 Literature Survey

Motion blur term is as old as the invention of the camera. Therefore, a large body of literature is available on motion blur because of the wide application area of cameras. While some of the studies try to eliminate motion blur, some other utilizes motion blur to gather information. Since our focus is on eliminating the motion blur and getting a sharper image, we mainly present studies in literature focusing on this subject while briefly giving examples from other studies which utilize motion blur such as generating super-resolution image in [1], motion estimation in [10], 3D shape estimation in [11] and vehicle speed detection in [30].

As a sensor to gather information about the environment, camera is frequently used in robotics applications [15, 17, 18, 30]. However, vision sensors are considerably sensitive to shocks and instantaneous movements especially at high speed and accelerations. An experimental study on the six-legged SensoRHex robot platform revealed that the performance of the feature detectors (i.e., edge and corner detection algorithms) degrades with increasing velocity values. Obtained results showed that at various normalized velocity values which measured as 0.1, 0.4, 0.8 and 1.0, respectively, every 2, 16, 28 and 60 of 100 frames cannot be included to feature detection algorithms due to the intolerable motion blur effect [52]. As these studies suggest, motion blur is an unwanted effect for the robotics applications.

There are different techniques proposed in literature to eliminate motion blur. Stabilization methods try to eliminate motion blur during the capturing process by canceling out the relative speed difference between the camera and the scene objects. This is generally done by stabilizing the camera mechanically so that the camera and the scene are standing still relative to each other during the exposure. This has the ability to achieve both a fixed camera line-of-sight while concurrently reduce motion blur. However the main disadvantage of the electro-mechanical approach is the complexity of the control problem and the requirement of a mechanical structure that is generally larger and heavier than the camera itself. To prevent camera motion, some newer cameras optically stabilize their lenses where adaptive optical elements controlled by inertial sensors compensate for camera motion to reduce blur [8, 37].

The camera stabilization mechanisms are also used in robotics applications. A camera stabilization platform is designed for SensoRHex robot which is capable of stabilizing the disturbances along angular rotations around 3 axes [7, 43]. The mechanism has three linear motors and which enables the platform to make roll, pitch and yaw motions. The solid drawing of the camera stabilization platform is given in Fig. 1.3. However, this kind of platforms can stabilize the camera motion only within a small range and are not effective when scene motion exists.

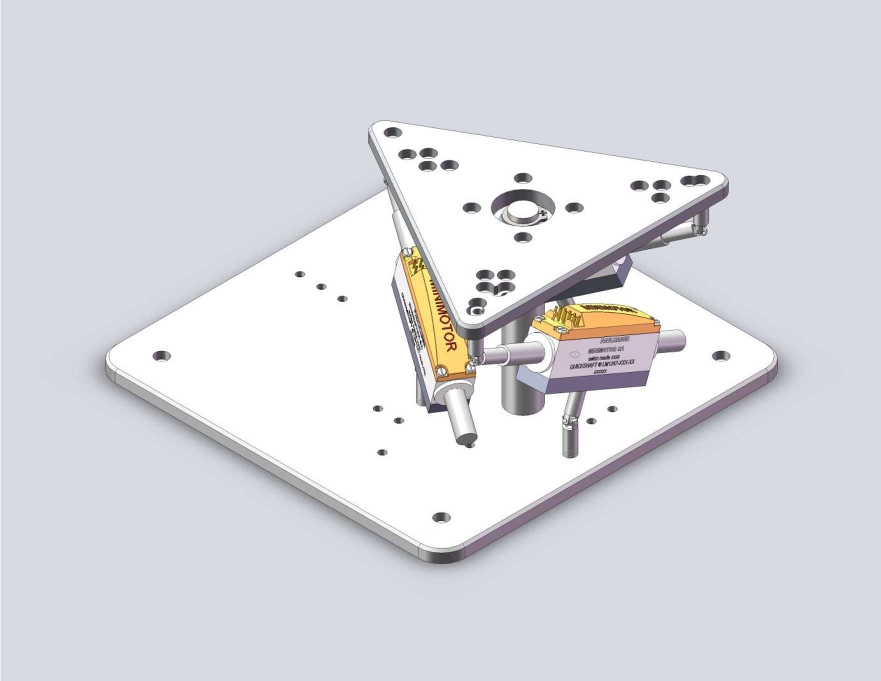


Figure 1.3: Camera stabilization platform designed for SensoRHex robot. The mechanism has three linear motors and has the capability to rotate around 3 axes (roll, pitch and yaw)

Since motion blur is a significant problem in robotics, there are studies that addresses the motion deblurring in robotics. Motion deblurring for a fast moving robot eye is studied in [23]. The mechanism is shown in Fig. 1.4. Instead of using any sensor, they estimate the PSF by using the system dynamics and input commands. The experimental results are evaluated using an average pixel difference based metric. [38] presents a coded exposure based motion deblurring for a wheeled mobile robot with a side looking camera on it which is shown in Fig. 1.4. They estimate the blur kernel by a simple template matching between adjacent frames.

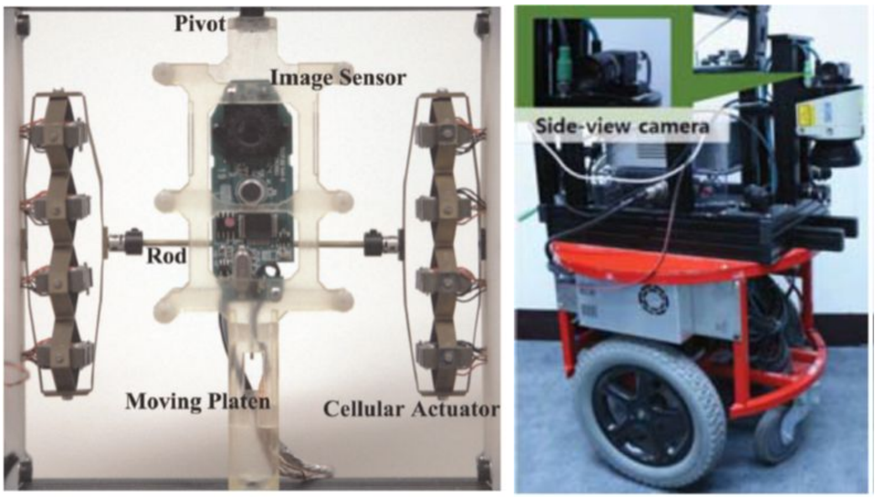


Figure 1.4: (Left) The robot eye mechanism used for motion deblurring in [23]. (Right) Wheeled mobile robot with a side looking camera that is used for coded exposure based motion deblurring in [38].

The other group of methods present post processing techniques to deblur a motion blurred image by the use of post image processing methods. The post processing methods can be classified into two groups: Deconvolution methods and hardware based approaches. The difference between these two groups is that the deconvolution methods use only the motion blurred image without manipulating the capturing process of the image where hardware based approaches manipulates the camera during the image capturing duration in order to modify the motion blurring process and motion blur characteristics of the captured image. Studies in these groups are explained below in detail.

Deconvolution methods; are a major family of methods that do not modify the image capture process but rely on purely algorithmic methods to rectify motion-blurred images. Different from camera stabilization approaches and hardware based post processing methods, deconvolution methods do not need a special mechanism and therefore can be utilized in a wide range of space-constrained or ruggedized applications. They are divided into two as blind and non-blind deconvolution methods based on the availability of Point Spread Function (PSF). The non-blind methods assumes

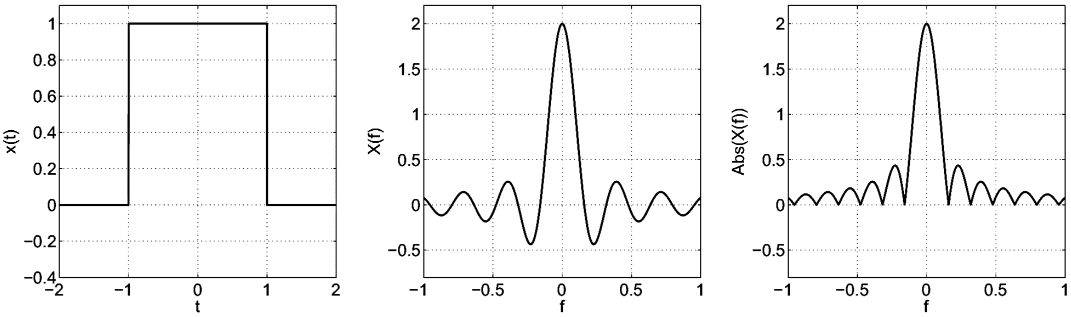


Figure 1.5: A box filter in time domain and corresponding sinc function in frequency domain which corresponds to a linear PSF. The zero response frequencies of the sinc function prevents its invertibility.

that the PSF of the motion blur is available whereas blind methods try to estimate the PSF from the motion blurred image. Then, by using the PSF, the blurred image is deconvolved to restore the original sharp image. The motion blurring process behaves like a box filter function as shown in Fig. 1.5. Because of the finite exposure time, the convolution with a box function in time domain corresponds to multiplication by a sinc function in frequency domain. The frequency transform of the PSF contains zeros (nulls). Therefore the corresponding spatial frequencies in the captured photo are lost. This makes the deblurring process an ill-posed problem.

Well known examples to deconvolution based methods are Richardson-Lucy [42, 32] and Wiener deconvolution [57]. Richardson-Lucy is an iterative restoration algorithm that uses a Poisson based statistical image pixel distribution model. Starting from an initial estimate of the image to restore , the method iteratively updates using the update expression given in (1.1), where t is the iteration index, $\tilde{f}_i$ is the i^{th} pixel of the original raster scanned image to be reconstructed, $h_{j,k}$ is the k^{th} pixel of the PSF of the j^{th} pixel and g_k is the k^{th} pixel of the degraded image.

$$\tilde{f}_i^{t+1} = \tilde{f}_i^t \sum_k \frac{h_{i,k}\, g_k}{\sum_j h_{j,k}\, \tilde{f}_j^t} \tag{1.1}$$

Wiener deconvolution is an application of Wiener filter to reduce the effect of noise amplification in deconvolution. It works in frequency domain and try to maximize the signal to noise ratio according to the image and noise power spectrum. The Wiener deconvolution is defined in frequency domain as given in (1.2), where $\tilde{F}$ and H are the Fourier transforms of restored image and the PSF, S is the mean power spectral density of the original signal f and N represents the power spectral density of the noise n. Other recent studies in this group are explained below in more detail.

$$\tilde{F} = \frac{H^* \, S}{|H|^2 \, S + N} \tag{1.2}$$

As a relatively new example than the above, [12] propose a method for uniform motion blurred images due to camera shake where in-plane rotation is negligible. They estimate the blur kernel in a coarse-to-fine fashion from a user specified patch which is rich in edge structure within the blurred image and the initial maximum size of the kernel. Then they perform a standard Richardson-Lucy deconvolution method for deblurring.

Another study assumes that the camera shake blur is mostly due to the 3D rotational motion of camera which is presented in [56]. Different from most of the deblurring methods in literature, their method handles non-uniform motion blur. They propose a new parametrized geometric model for blurring in terms of the camera rotations. They use two methods, first use only single blurred image and secondly use both a blurred and a sharp but noisy image.

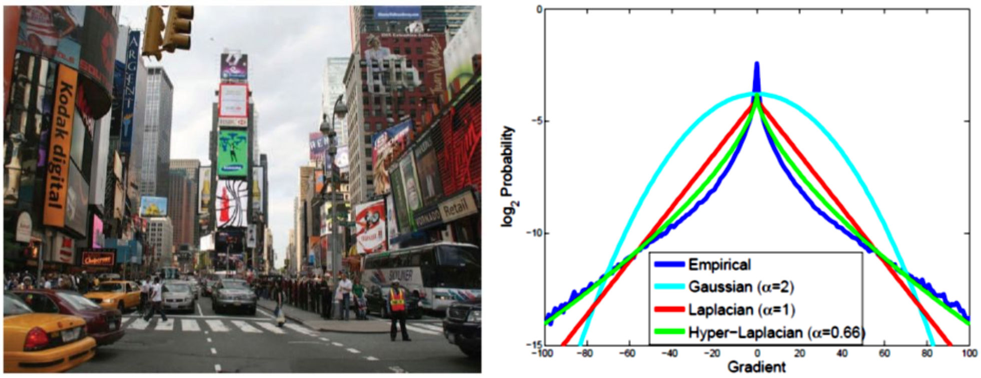


Figure 1.6: (Left) A typical real-world scene. (Right) The empirical distribution of gradients in the scene (blue), along with a Gaussian fit (cyan), a Laplacian fit (red) and a Hyper-Laplacian with $\alpha = 2/3$ (green) [25].

[25] presents a new deconvolution approach using hyper-Laplacian priors shown in Fig. 1.6. Their solution adopt either a look-up-table or analytical formulae. Either way, they can achieve comparable quality to iteratively reweighted least squares based methods while still being several order of magnitude faster in computation.

In [14], the authors present a new blur kernel estimation method based on the statistical irregularities of motion blurred images power spectrum. They use a power-law based model for describing natural images. Their method can achieve better running times compared to maximum a posteriori estimation based methods.

[48] focuses on reducing visual artifacts in deblurring such as noise and ringing. They

develop a probabilistic model for the spatial randomness of noise and a local smoothness prior that reduces ringing artifacts in smooth regions.

Other interesting examples include for example [22] where they use inertial measurement sensors to estimate blur function and use it for nonuniform deconvolution. Another sensor assisted motion deblurring study is presented by [20] for smartphones. They especially focus on the quality of human face deblurring in images.

A group of hardware related techniques try to manipulate the image capturing process by configuring or controlling the camera hardware. This manipulation generates some favorable properties for the motion blurred images that help the deblurring process. These methods are generally focused on the motion blur caused not by the camera ego-motion but by the scene motion.

Modulated Capture
PSF Estimation
Image Deconvolution

Figure 1.7: The steps of motion deblurring method proposed in [40]. They only manipulate the capturing part to get an invertible motion blur PSF.

Fluttered Shutter Approach: To prevent the zeros in PSF [40] propose a different exposure method. Their method includes the three familiar steps as given in Fig. 1.7. The only modification is done in the capturing part. Rather than keeping the shutter open throughout the exposure duration, they propose a fluttered shutter approach. They place a ferro-electric liquid crystal shutter in front of the camera lens as seen in Fig. 1.8 to modulate the exposure with a binary code they proposed. They call the resultant blur a “coded blur”. They toggle the shutter on and off in such a way that the motion is integrated in different durations that corresponds different filtering effects to be involved. The flutter changes the box filter to a broad-band filter that preserves high-frequency spatial details in the blurred image and the corresponding deconvolution becomes a well-posed problem. The corresponding PSF obtained by this method can have the maximum coverage in the frequency domain so that a null-free PSF can be obtained.

They use a binary code sequence with 52 chops. They suggest the use of two different code types. One of the codes are called Modified Uniformly Redundant Arrays (MURA) codes which are used in coded-aperture astronomical imaging. Another code is proposed by them which is the sequence of 1010000111000001010000110011110111010111001001100111. The frequency domain magnitudes of the proposed codes are given in Fig. 1.9 together with the conventional flat exposure. As seen from the figure, the coded exposure prevents the nulls in the PSF which provides invertibility of motion blur.

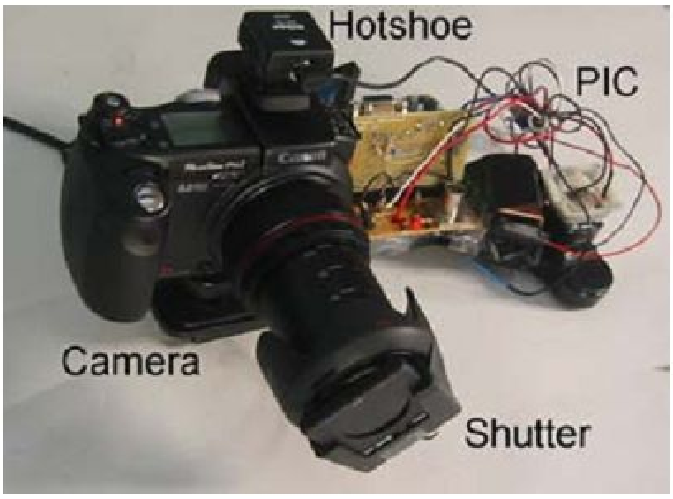
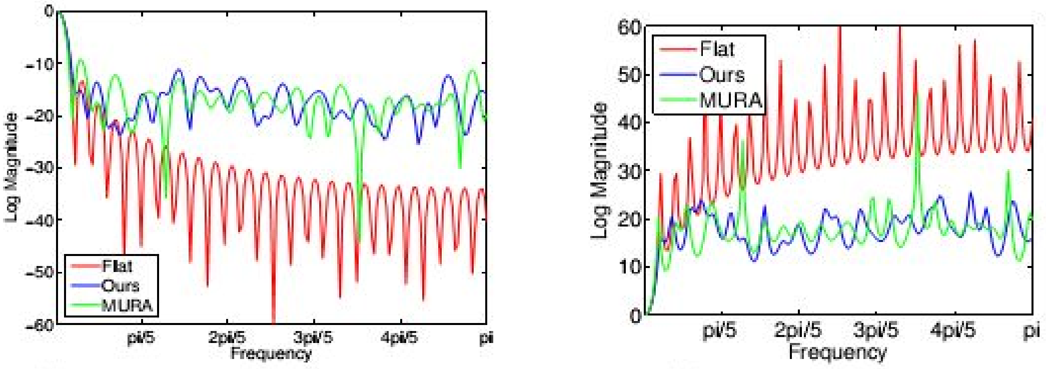


Figure 1.8: The prototype camera setup with fluttered shutter mechanism and its control circuit [40].

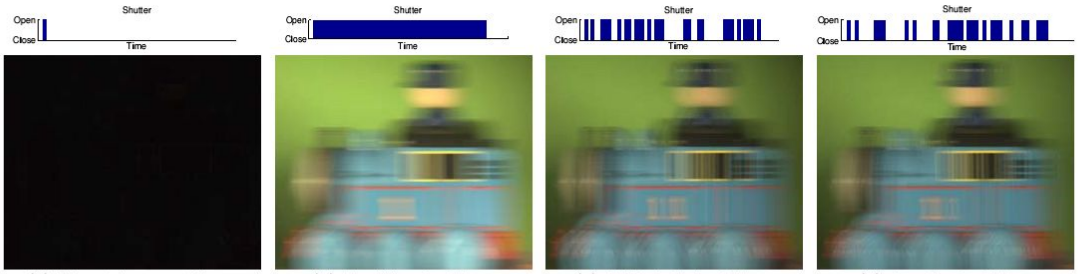


(a) Linear Convolution Filter DFT

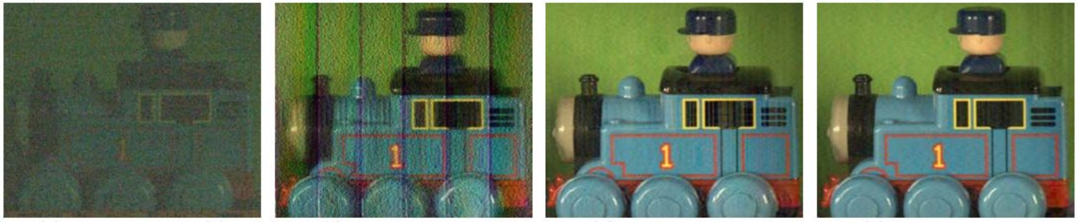
(b) Decoding Filter DFT

Figure 1.9: The DFT of the box filter (flat blur) is a sinc function and frequencies with sharp dips are lost. The MURA code filter is better than box filter but the variance of the frequency magnitudes is high compared to their proposed code. (b) The DFT of decoding filters show that box and MURA filters require significant amplification of certain high frequencies. [40]

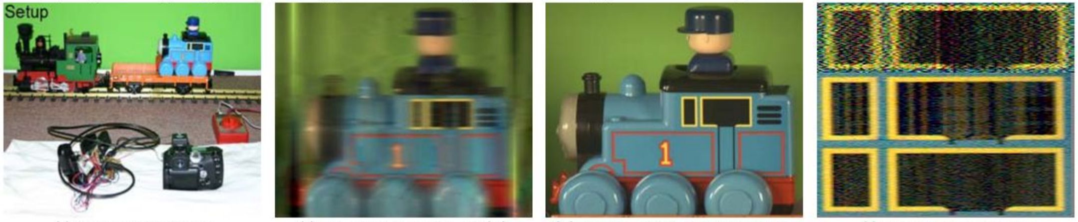
The next step of the process is PSF estimation as usual. They do not perform PSF estimation, but instead rely on a user assisted interactions to estimate the blur path. The last step is the deconvolution of the image by using the given PSF. Instead of using a circular deconvolution method, they use a linear algebra approach. The comparison of the obtained results are given in Fig. 1.10. It is seen from the results that fluttered shutter approach is very successful in preserving the high frequency components and also good at the obtained signal to noise ratio (SNR).



(a) Short Exposure Photo (b) Traditional, 200ms (c) MURA Code, 200ms (d) Our Code, 200ms



(e) Log intensity of (a) (f) Deblurred Image (g) Deblurred Image (h) Deblurred Image



(i) Toy Train Setup (j) RL deblur result of (b) (k) Photo of Static Toy Train (l) Deblurring details

Figure 1.10: The comparisons of obtained results. a) A short exposure photo is shown in a low light environment and its log intensity scaled form is given at (e). b) An image obtained with traditional continuous exposure with 200ms duration is shown. c) Fluttered shutter exposure using MURA code. d) Fluttered shutter exposure using proposed code. f,g,h) Deblurred results [40].

There are also some weaknesses of their study. Firstly, they require a camera with a custom fluttering mechanism otherwise it is not possible to apply their method using the conventional cameras. Secondly, the PSF should be known. Their method does not include an automated PSF estimation and therefore an accurate PSF should be maintained to their method in order to get good results. They assume the absence of large occlusions and dis-occlusions which breaks the spatially-invariant PSF assumption. This assumption also brings another constraint on the motion of the objects. They assume linear constant velocity motions that lead to spatially-invariant blur in photos. The direction of the motion should also be provided by the user and the motions having different directions are out of their scope. They also cannot handle transparent or translucent objects.

The inverse filtering methods can be classified according to their assumption on the availability of the PSF. Those methods which assume that the PSF is unknown are called blind deblurring methods and those that assume a known PSF are called non-blind methods. Even when the PSF is known, deconvolution amplifies noise and high frequencies are usually lost in the deblurred results.

Two camera hybrid method for PSF estimation: [6] proposes a hybrid method including two cameras for motion deblurring. Their main contribution is their proposed method to estimate an accurate PSF. They use a hybrid system including two cameras as seen in Fig. 12. One of the cameras capture low resolution, high frame rate images to get motion blur free images and the other camera captures high resolution, low frame rate images with high motion blur. The low resolution camera is used to obtain the motion path to estimate the PSF. High resolution camera is the main camera. The images captured by the high resolution camera will be deblurred by using the PSF obtained from the secondary low resolution camera. They make an optical flow based motion estimation for each frame obtained from the secondary camera. The discrete motion vectors are interpolated among the frames to obtain an accurate PSF. Finally, they utilize existing image deconvolution algorithms to obtain the deblurred image. They select Richardson-Lucy iterative deconvolution algorithm.

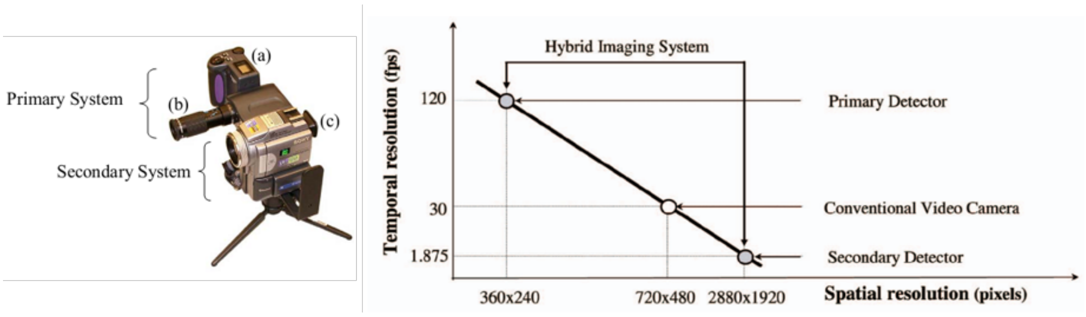


Figure 1.11: Hybrid camera setup proposed by [6]. The primary camera has a high resolution and low frame rate whereas the secondary camera has a low resolution and high frame rate.

Multi image deblurring using varying exposure: [2] proposes differently exposed multiple images to achieve invertible motion blur. Similar to the work of [40], the aim in this study is to obtain blurred images having invertible PSF. To achieve this, they use multiple images having different exposures each having different PSFs. When these images are combined, the nulls in the frequency components of one frame can be filled by other frames. Therefore, the combined frequency transform becomes null-free and makes the deblurring a well-posed problem. The advantage of their method over [40] is the use of a standard camera without the need of a flutter mechanism. The individual PSFs with zeros and the combined operator P are shown in Fig. 1.12 and compared with fluttered shutter method.

The deblurring process using this method includes additional steps which does not exist in inverse filtering methods. Similar to other methods, after capturing the images, the first step is the PSF estimation. Motion vectors are computed by matching the frames captured using the same exposure. Averaging the motion vectors for dif-

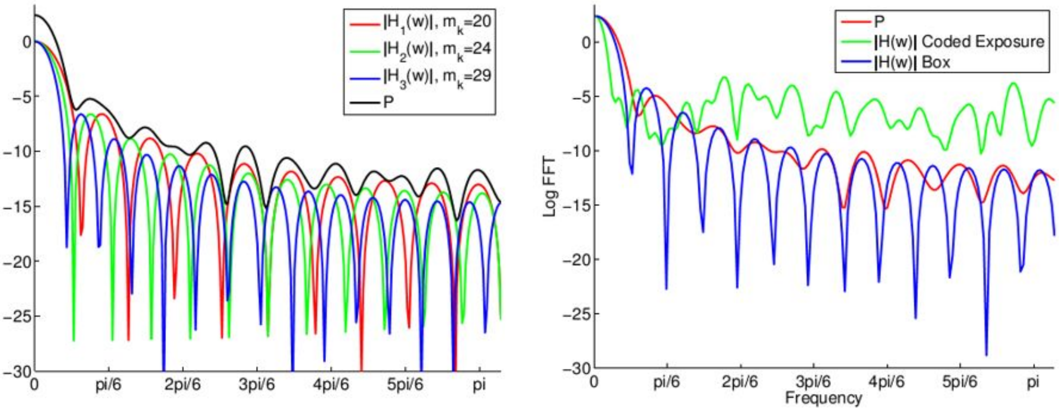


Figure 1.12: Noninvertible PSFs Hi(w) each having zeros and the combined operator P which does not have zeros shown on left. Comparison with coded exposure shown on right [2].

ferent exposures gives the estimate of object velocity v. For spatially-invariant linear blur, PSF is found by multiplying the space object velocity v (pixel/ms) with the exposure time (ms). The rest of the process moves iteratively. Each iteration begins with segmenting the moving object by using a background subtraction method. Then the blurred foregrounds need to be aligned before deconvolution. The input photos are scaled to the same intensity level using the ratios between the corresponding exposure times. After aligning the blurred foregrounds, deblurring is performed by solving the linear system equations. The next step is the alignment refinement. The errors caused by the potential misalignments occurred in the previous alignment step may yield errors in deconvolution. This refinement process is applied to improve the results. The iteration finishes with segmentation refinement step which is similar to the alignment step. The quality of segmentation and deblurring improves significantly after the first iteration. The algorithm iterates 1 or 2 times for the final result. The obtained results are shown and compared to other methods in Fig. 1.13. The presented method has similar limitations to the fluttered shutter approach. Moreover, it requires additional processing steps such as segmentation and alignment. However, this method does not require additional hardware or camera motion which is the fundamental advantage over other method.

Motion invariant deblurring for 1D: [26] presents a method to deblurring a scene with objects moving at different speeds along 1D direction, without requiring any segmentation and without the knowledge of the object speeds. The method depends on using a camera that moves during the exposure at a constant acceleration(parabolic integration). The key point is that, while the camera accelerates along the motions of objects, for any object velocity within a range, there is always one moment during exposure when the camera is perfectly tracking the object. In other words, there exists an instant when the relative speeds of the object and the camera are zero. The obtained

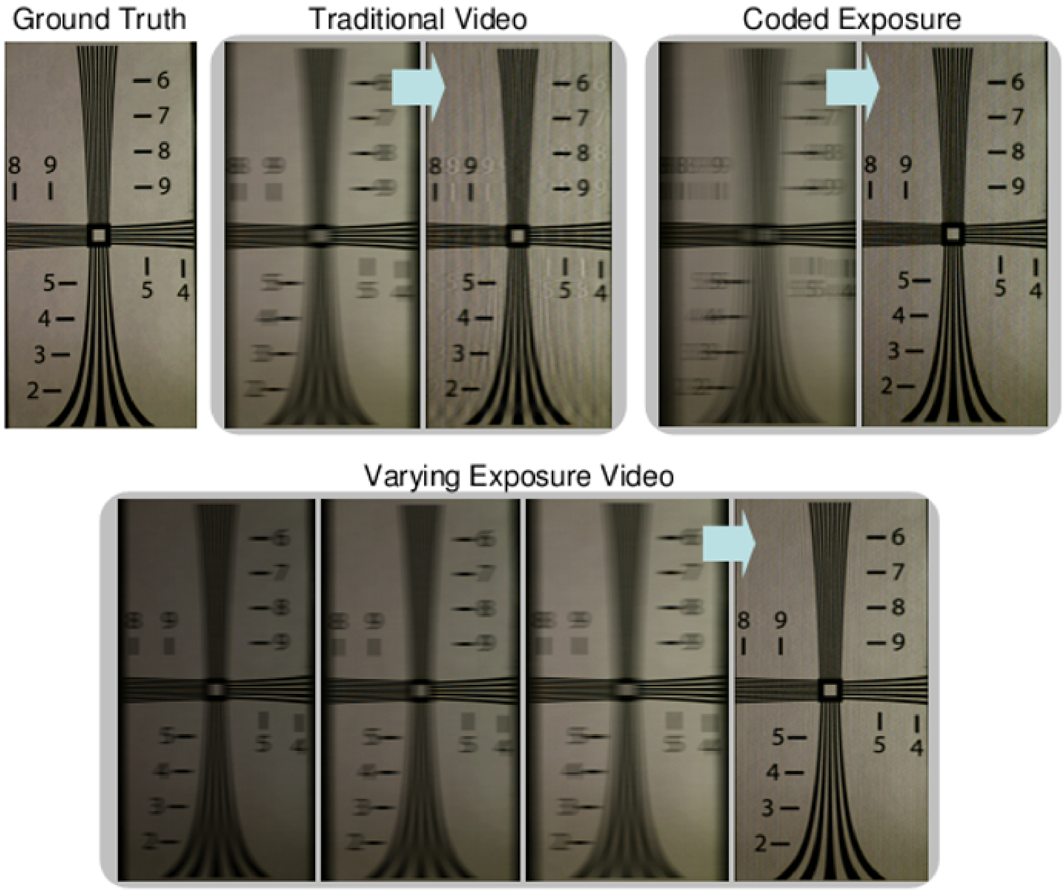
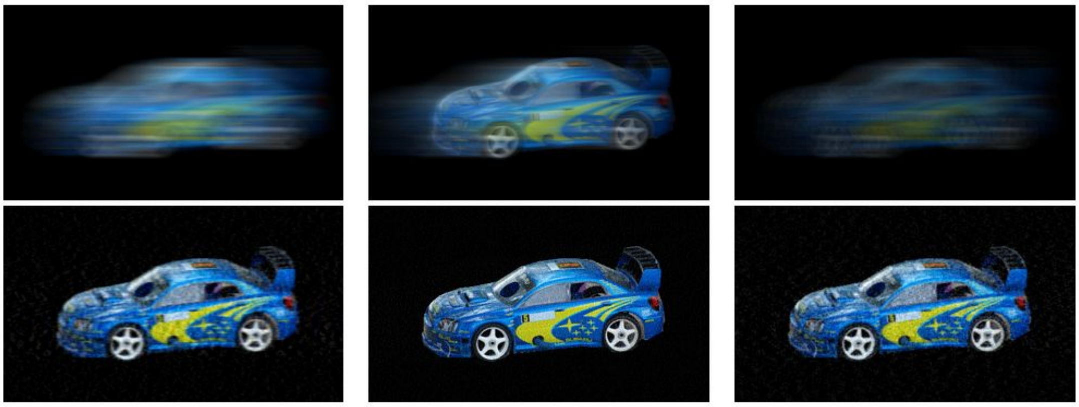


Figure 1.13: Comparison of varying exposure, traditional video and coded exposure methods. Varying exposure method achieves similar performance to the coded exposure method and a better performance than the traditional video [2].

results are shown in Fig. 1.14 together with a comparison with other methods. The advantage of the method is that it requires no motion estimation or object segmentation and provides nearly optimal reconstruction for the worst-case speed within a given range. However, its drawback is the requirement of a speed controlled platform and its use is limited only for 1D motions.

Motion invariant deblurring for 2D: [3] extends the 1D study of [26] for 2D motions by translating a camera sensor circularly about the optical axis during exposure. This enables the preservation of high frequencies for a wide range of in-plane linear object motion in any direction within some predetermined speed range. Fig. 1.15 shows the motion of the sensor for 1D and circular 2D cases. Although no object may be photographed sharply at capture time due to the continous sensor motion, objects having different motions can be captured in a single image and can be deconvolved with similar quality. In addition, circular sensor motion has good effects on blur estimation because of the distinct frequency zero patterns of the resulting motion blur



(a) Static (b) Parabola (c) Flutter Shutter

Figure 1.14: The motion blurred frames captured with a static camera, parabolic motion camera and fluttered shutter camera. Their deblurring outputs are given on the bottom images [26].

PSFs. The disadvantage of the method is that it does not favor static objects. Because of the sensor motion, static objects are also blurred. They suggest stopping the sensor motion for a while during the exposure for future work.

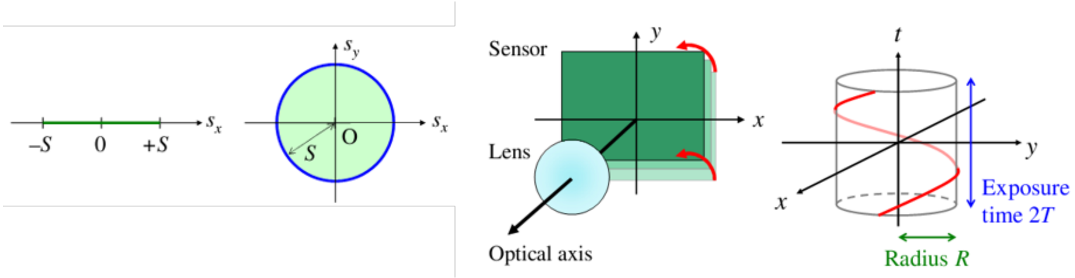


Figure 1.15: First figure shows the range of motion for 1D case proposed by [26] and the second figure shows the range of motion for the proposed 2D case by [3]. Third figure shows the circular motion of the sensor. The sensor is translated circularly about the optical axis. Last figure shows the sensor motion trajectory (red curve) in the space-time volume.

There are also studies that are benchmarking the available motion deblurring methods based on a quantitative metric. One of them is presented in [47]. They obtain linear motion blurred images using a 1D linear camera translation setup similar to our work. The methods are tested both on synthetic and real test images and compared based on PSNR metric. They test and compare Wiener deconvolution, R-L, Regularized filter, MLE and Sondhi, which are quite old methods. They also provide the computation times of the tested methods. Another study [24] is about a benchmarking study for single image blind deconvolution methods. They present a dataset by recording the camera shake motion and play back on a 6D mechanism as shown in Fig. 1.16. They test and compare deblurring methods by Fergus, Shan, Whyte, Cho, Xu and Krishnan using PSNR and similarity based quantitative metrics for comparison.

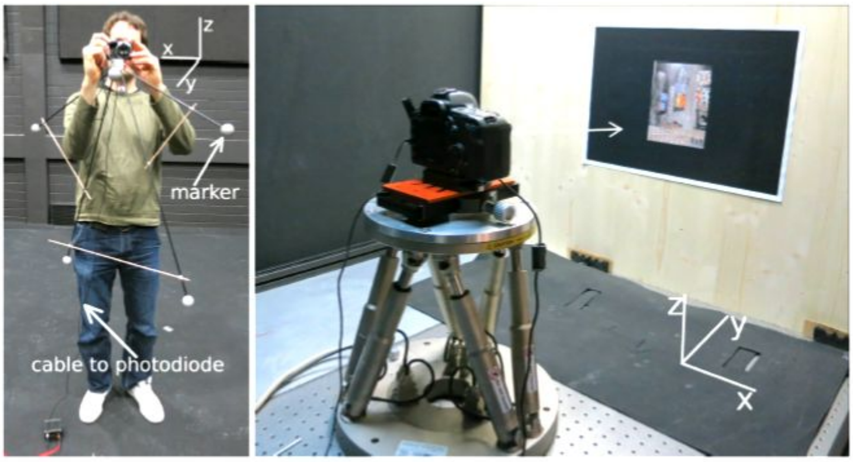


Figure 1.16: Experimental setup used in [24]. (Left) Light-weight structure with reflective spherical markers for recording camera shake with a Vicon system. (Right) Camera attached to a high-precision Stewart platform to take pictures with played back camera shakes in a controlled setting.

Most of the deblurring methods mentioned above just presents their results for visual inspection without any quantitative measure for comparison. Only a few of the studies compare their results quantitatively using PSNR(Peak Signal to Noise Ratio)[24, 4] and MSE(Mean Square Error) as a metric. The suitability of those quantitative metrics are also questionable for a fair comparison. There are also a number of other metrics that are used for image quality assessment purposes such as SSIM(Structural Similarity Index), UQI(Universal Quality Measure), PQR(Picture Quality Rating), MOS(Mean Opinion Score) etc. These metrics are generally used for image compression/decompression quality purposes and are not generally suitable for performance comparison of other different computer vision applications.

In this study, we test and compare the methods proposed by [56], [25],[48] and [14] on our motion blur dataset including uniform and nonuniform motion blurred images with different blur magnitudes.

1.2 Contributions

In the present thesis study, our focus is on motion blur problem and its compensation to increase the performance of feature detection methods in legged robotics computer vision applications. Our first contribution is the finding that available image quality assessment metrics do not indicate the quality of an image in terms of feature detection performance. This finding led us to propose a novel metric based on feature detector performance that we demonstrate more accurately reflects the performance of various single-frame image deblurring methods for computer vision applications, in particular vision for dexterous legged robotic systems. Then, we use this metric for a careful benchmarking of these methods demonstrating that conventional image quality metrics including those used in deblurring studies do not accurately predict deblurring contribution to feature detector performance.

Secondly, we analyze the properties of motion blur in legged robots and find out the particular properties of motion blur in legged robotic platforms. Being inspired from the fact that successive image frames captured from our legged robot have different magnitudes of motion blur, we propose a multi image motion deblurring method to combine the existing information in multiple frames for a better motion deblurring.

Finally, there was a need of a dataset for motion blur studies in legged mobile robotic platforms. However, we noticed that there was no available motion blur data set for this application area. Therefore, we build a comprehensive motion blur data set, composed of multiple on-board inertial sensors supplementing the camera sensor as well as synchronized data from external sensors for ground truth information. It is our hope that this data set is not only suitable for the objective comparison of current and future algorithms on motion deblurring for legged mobile robotic but also for other vision research for cameras mounted on legged platform.

1.3 Outline of the Thesis

The organization of the thesis is as follows: After this introduction chapter, in Chapter 2, we briefly introduce the mathematical background on motion blur formation, restoration of motion blurred images, problems in single frame inverse filtering and then propose a multi-image motion deblurring method in frequency domain that can be used for deblurring in legged mobile robotics applications.

In Chapter 3, we introduce our feature detection performance based metric in and its detailed justification using our relevant results. This is followed in Chapter 4 by the description of the multi-modal motion-blur data collected for vision research for legged platforms, which is also the basis of our experiments presented in Chapter 5. In Chapter 5, we present our detailed comparative results using our novel metric. The results and findings of our studies are discussed and summarized together with possible future studies in Chapter 6.

CHAPTER 2

PROPOSAL OF A MULTI-IMAGE MOTION DEBLURRING METHOD FOR APPLICATION TO A LEGGED MOBILE ROBOTIC PLATFORM

2.1 Theory of Motion Blur Formation and the Inverse Problem

2.1.1 Motion Blur Model as a Convolution

To be able to understand the image blurring problem, the image formation process should be modeled. The following equation gives a mathematical model [7] of how a camera observed image g(x,y) is formed from an ideal image f(x,y):

$$g(x,y) = n\left(\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} h(y,x;r,c)\cdot f(r,c)drdc; y, x\right) \tag{2.1}$$

Ideally every point in the scene should correspond to only one point on the captured image. However, the relative motion between the scene and the camera during the exposure time of the camera violates this assumption. Points in the scene may fall onto different pixels and the information of a scene point is spreaded into more than one pixels. The mathematical relation of this spreading is represented by a Point Spread Function (PSF). In the above equation, the point spread function for each pixel in the image is defined by h(y,x). Due to its nature, the PSF represents a real-valued intensity distribution and takes non-negative values only. Another effect in the image formation process is the unwanted changes in the measured pixel value which is called noise. Noise is mainly caused by the measuring device in the cameras and modeled in the above equation by n.

This model imposes different PSF for every pixel in the image which is computationally not feasible. It is also not possible to estimate a different PSF value for each pixel location. Therefore, it is assumed that every point has the same PSF, i.e. PSF is spatially invariant over the image. This assumption holds for a camera-scene setup where the scene is composed of points having the same distance to the camera and the same lateral motion relative to the camera. Single PSF for every pixel assumption yields:

$$\forall y, x, r, c : h(y, x; r, c) = h(y - r, x - c) \tag{2.2}$$

The noise term can also be included as an additive term with zero-mean. This assumption holds in most of the cases and also simplifies the image formation model as:

$$g(y, x) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} h(y - r, x - c) \cdot f(r, c) drdc + n(y, x) \tag{2.3}$$

The above model is written for continuous time signals. For digital images, this model can easily be converted to a discrete model as:

$$\begin{aligned} g(y, x) &= \sum_{-\infty}^{+\infty} \sum_{-\infty}^{+\infty} h(y - r, x - c) \cdot f(r, c) + n(y, x) \\ &= h(y, x) \otimes f(y, x) + n(y, x) \end{aligned} \tag{2.4}$$

The above operation is referred as 2D convolution of functions h and f. h is called convolution kernel and the dimensions of the h matrix are often called a convolution window.

The process of motion deblurring is composed of two sub-problems: The estimation of the function that caused the motion blur, and applying a restoration algorithm. PSF is briefly explained in the next section.

2.1.2 Point Spread Function

The PSF can be represented by a convolution kernel which satisfies some certain conditions. Every convolution kernel cannot be a valid PSF. A PSF is a function defined as:

$$h = \Re \times \Re \longrightarrow \Re_+ \tag{2.5}$$

Therefore, a PSF convolution kernel is composed only of non-negative values. The kernel h must also satisfy the following energy conservation constraint to be a valid blurring function:

$$\iint h(r,c)\,dr\,dc = 1 \tag{2.6}$$

in case of a discrete kernel:

$$\sum_{r}\sum_{c} h(r,c) = 1 \tag{2.7}$$

As given above, if the sum of all elements of a convolution kernel is equal to 1, then the kernel is called a normalized kernel. The average intensity value of an image does not change after applying a normalized convolution kernel which states that the energy is neither lost nor gained by the blurring operation.

2.1.3 Inverse Filtering

Once the motion blurred image and the PSF that is involved in the blurring process are known, the original image is tried to be obtained by finding and applying the inverse of the convolution operation. The process of reconstruction of the original image from the blurred one with the same blurring function is called deconvolution.

The simplest approach to deconvolution is the inverse filtering. The key idea of this method is that convolutions in spatial domain become pixel-by-pixel multiplications in frequency domain:

$$g = h \otimes f \quad \Leftrightarrow \quad G = H \cdot F \tag{2.8}$$

where G and F are Fourier transforms of the blurred and original images that are represented by g and f, respectively; H is the Fourier transform of the convolution kernel h. Knowing the convolution kernel h and the degraded image g, we can find the restored image $\tilde{f}$ by the following operation:

$$\tilde{f} = \mathcal{F}^{-1}\left(\frac{G}{H}\right) \tag{2.9}$$

where $\mathcal{F}$ is the discrete Fourier transform; $\mathcal{F}^{-1}$ is the inverse discrete Fourier transform and denotes the (complex) pixel-by-pixel division. This method seems to be very simple and effective however, the inverse filtering produces unsatisfactory results on (additive) noised images. Let us suppose that some additive noise n is present:

$$g = h \otimes f + n \quad \Leftrightarrow \quad G = H \cdot F + N \tag{2.10}$$

where the capitals denote Fourier transforms. The inverse filtering delivers the following approximation of :

$$\begin{aligned} \tilde{f} &= \mathcal{F}^{-1}\left(\frac{H \cdot F + N}{H}\right) \\ &= \mathcal{F}^{-1}\left(F + \frac{N}{H}\right) \end{aligned} \tag{2.11}$$

If H is a matrix with relatively small absolute values (which is almost always the case if H is a valid PSF) then the noise N will be amplified dramatically by the division. This makes the inverse filtering unusable, due to the fact that a small amount of noise is always present.

2.2 Multi-Image Motion Deblurring

The frequency domain plots of 1-D linear PSFs with different blur lengths are shown in Fig. 2.1. As can easily be noticed, the zeros of different PSFs appear at different frequencies. If we have more than one blurred frames of the same scene with different PSFs, the above inverse filtering operation can be written for each blurred image as,

$$\tilde{F}(\omega) = \frac{G_i(\omega)}{H_i(\omega)}, \qquad i = 1, 2, ..., n \tag{2.12}$$

where i denotes the frame index. Then the frequencies corresponding to the zeros of one PSF can be recovered from another PSF if the zeros do not overlap. Now, instead of the ideal case, if we add noise n to the above blurring operation, then it becomes,

$$G_i(\omega) = F(\omega)H_i(\omega) + N_i(\omega) \tag{2.13}$$

Then the inverse filtering becomes,

$$\tilde{F}_i(\omega) = F(\omega) + \frac{N_i(\omega)}{H_i(\omega)} \tag{2.14}$$

At frequencies where the $H_i(\omega)$ becomes very small, the noise is amplified severely. Therefore, to reduce the noise amplification in inverse filtering, it is better to utilize

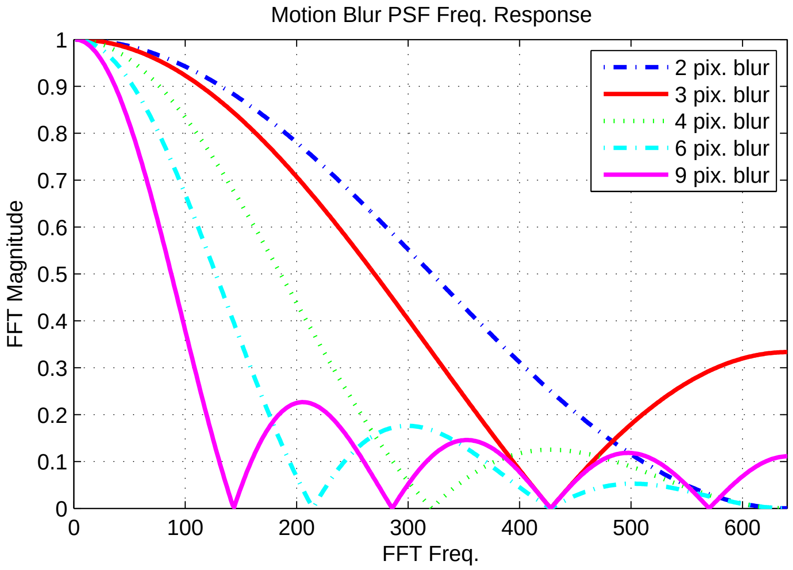


Figure 2.1: Frequency responses of 1-D linear PSFs with different blur lengths.

the maximum $H_i(\omega)$ for every frequency ω to reconstruct $F(\omega)$ with minimum possible noise amplification. We will apply the inverse filtering using non-overlapping frequency windows wi, where the sum of all windows are equal to identity.

$$\sum_{i=1}^{n} W_i = I \tag{2.15}$$

Then the inverse filtering of each blurred image is subject to its frequency window.

$$\tilde{F} \cdot W_i = \frac{G_i(\omega)}{H_i(\omega)} \cdot W_i \tag{2.16}$$

A frequency window w_i for i^{th} image is defined as,

$$W_i(\omega_x, \omega_y) = \begin{cases} 1, & |H_i(\omega_x, \omega_y)| > |H_k(\omega_x, \omega_y)| \text{ for } k = 1, 2, ...n \ \wedge \ k \neq i \\ 0, & \text{else} \end{cases} \tag{2.17}$$

Then the deblurred image is obtained by the summation of all windowed inverse filtered images in frequency domain as shown below.

$$\begin{aligned}\sum_{i=1}^{n}(\tilde{F}\cdot W_i) &= \tilde{F}\sum_{i=1}^{n} W_i \\ &= \tilde{F}\cdot I \\ &= \tilde{F}\end{aligned} \tag{2.18}$$

The windowed filtering operation can be illustrated by taking the maximum of frequency responses of each PSF. Fig. 2.2 demonstrates this operation over the given 2 PSFs. It can be noticed that, the combined filter PSF has no zeros and moreover, each frequency can be recovered with less amplification which will increase the SNR (Signal to Noise Ratio).

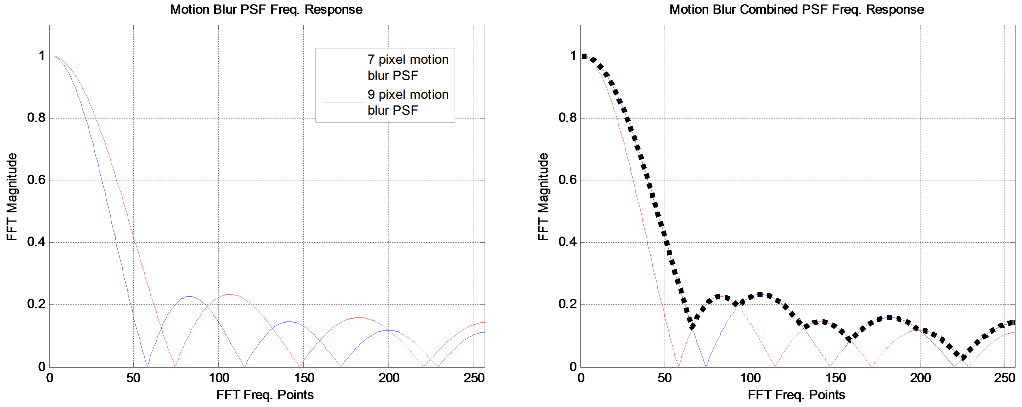


Figure 2.2: (Left) Frequency responses of two different PSFs. (Right) Illustration of the windowed filtering of PSFs based on maximum frequency responses. Dotted line shows the equivalent combined filter frequency response.

2.2.1 Zero Response Frequencies of a Linear Motion Blur PSF

Linear motion blur point spread function(PSF) can be modeled as a box filter which is given in Fig. 1.5. The width of the box filter corresponds to the magnitude of the motion blur. Its amplitude also changes according to the width so that the underlying area is unity. Since motion blur does not change the total energy in the image, this ensures the energy conservation.

The Fourier Transform of the box filter can be derived as follows. First, we define the box function that is symmetric with respect to y axis and whose width is T.

$$b_T(t) = \begin{cases} 1, & t \in \left[-\frac{T}{2}, \frac{T}{2}\right] \\ 0, & \text{else} \end{cases} \tag{2.19}$$

Then take the Fourier Transform of the defined box function.

$$\begin{aligned} \hat{b}_T(f) &= \int_{-T/2}^{+T/2} e^{-i2\pi f t} dt \\ &= \frac{e^{-i\pi f T} - e^{i\pi f T}}{-2\pi i f} \end{aligned} \tag{2.20}$$

The frequency domain transformation of a box filter is obtained as nothing but a sinc function as follows.

$$\begin{aligned} \hat{b}_T(f) &= T\,\frac{\sin(\pi f T)}{\pi f T} \\ &= T\,\mathrm{sinc}(\pi f T) \end{aligned} \tag{2.21}$$

The plot of the sinc function is given in Fig. 1.5. The magnitude of this sinc function is also shown on the last plot of the same figure.

Next, we will find the zero response frequencies of the given box filter by equating its Fourier Transform to zero as given below. f^z corresponds to the zero response frequencies.

$$\frac{\sin(\pi f^z T)}{\pi f^z T} = 0 \tag{2.22}$$

It is known that for $f = 0$, the response of the box filter is unity. Then for other frequency values of f, the denominator is nonzero. Therefore, we only have the numerator term equal to zero.

$$\sin(\pi f^z T) = 0 \tag{2.23}$$

For the above term to be zero, the sine function parameters should be integer multiple of π.

$$\Rightarrow \pi f^z T = n.\pi \ , \qquad n = 1, 2, ..., \infty \tag{2.24}$$

Then, the zero response frequencies are obtained simply as,

$$\Rightarrow f^z = \frac{n}{T} \leq \frac{1}{2} \, , \qquad n = 1, 2, ..., \infty \tag{2.25}$$

The set of zero response frequencies can be written as,

$$f^z = \left\{ \frac{1}{T}, \frac{2}{T}, ..., \frac{N}{T} \right\} \ , \qquad N = \left\lfloor \frac{T}{2} \right\rfloor \tag{2.26}$$

Let us assume two different linear motion blur functions corresponding to box filters with a width of $T1$ and $T2$. Then the zeros will appear at frequencies given below.

$$\begin{aligned} T = T_1 \quad &\Rightarrow \quad f^z = \left\{ \frac{1}{T_1}, \frac{2}{T_1}, ..., \frac{N_1}{T_1} \right\} \\ T = T_2 \quad &\Rightarrow \quad f^z = \left\{ \frac{1}{T_2}, \frac{2}{T_2}, ..., \frac{N_2}{T_2} \right\} \end{aligned} \tag{2.27}$$

If the widths of box filters are irrational, then there will be no common zeros at the same frequencies.

$$\text{if} \quad T_1, T_2 \notin \mathbb{Q} \quad \Rightarrow \quad f_1^z \cap f_2^z = \varnothing \tag{2.28}$$

However, if the widths of box filters are rational multiples of each other, then there exists a ratio $\frac{a}{b}$ such that,

$$\text{if} \quad T_1, T_2 \in \mathbb{Q} \quad \Rightarrow \quad \frac{T_1}{T_2} = \frac{a}{b} \, , \quad a, b \in \mathbb{Z} \ \wedge \ a \perp b \tag{2.29}$$

If a and b satisfies the following conditions, then there exists common zero frequencies of PSFs.

$$\text{if} \quad a \leq \frac{T_1}{2} \wedge b \leq \frac{T_2}{2} \Rightarrow \quad f_1^z \cap f_2^z \neq \varnothing \tag{2.30}$$

Then, the common zero response frequencies form a set of frequencies including the terms given below.

$$f_{1,2}^{z} = \left\{ \frac{a}{T_1}, \frac{2a}{T_1}, ..., \frac{na}{T_1} \right\} = \left\{ \frac{b}{T_2}, \frac{2b}{T_2}, ..., \frac{nb}{T_2} \right\} \tag{2.31}$$

where n can be found using the following equation.

$$n = \left\lfloor \frac{T_1}{2a} \right\rfloor = \left\lfloor \frac{T_2}{2b} \right\rfloor \tag{2.32}$$

Next, we will demonstrate the common zeros for different box filter widths. The following examples calculate the zeros of box filters having widths of 8 and 16, 10 and 25, 9 and 11 pixels.

Example 1. This example calculates the common zeros of two blur filters whose magnitudes (8 and 16) are integer multiple of one another, in particular 2. Calculation of zero response frequencies for PSF pairs:

$$\text{Given } T_1 = 8 \text{ pixel}, \quad T_2 = 16 \text{ pixel}$$

For the 8 pixels wide box filter, the zero response frequencies are:

$$\begin{aligned} T_1 = 8 \quad &\Rightarrow \quad f_1^z = \left\{ \frac{1}{8}, \frac{2}{8}, ..., \frac{N_1}{8} \right\} \quad \text{where } N_1 = \left\lfloor \frac{T_1}{2} \right\rfloor = 4 \\ &\Rightarrow \quad f_1^z = \left\{ \frac{1}{8}, \frac{2}{8}, ..., \frac{4}{8} \right\} \end{aligned} \tag{2.33}$$

For the 16 pixels wide box filter, the zero response frequencies are:

$$\begin{aligned} T_2 = 16 \quad &\Rightarrow \quad f_2^z = \left\{ \frac{1}{16}, \frac{2}{16}, ..., \frac{N_2}{16} \right\} \quad \text{where } N_2 = \left\lfloor \frac{T_1}{2} \right\rfloor = 8 \\ &\Rightarrow \quad f_2^z = \left\{ \frac{1}{16}, \frac{2}{16}, ..., \frac{8}{16} \right\} \end{aligned} \tag{2.34}$$

The widths of the box filters are rational numbers therefore there exists a, b such that,

$$\begin{aligned} T_1, T_2 \in \mathbb{Q} \quad &\Rightarrow \quad \frac{T_1}{T_2} = \frac{8}{16} = \frac{1}{2} = \frac{a}{b}, \quad a, b \in \mathbb{Z} \ \wedge \ a \perp b \\ &\Rightarrow \quad a = 1, \quad b = 2 \end{aligned} \tag{2.35}$$

Common zero response frequencies are,

$$f^z_{1,2} = \left\{\frac{1}{8}, \frac{2}{8}, ..., \frac{4}{8}\right\} = \left\{\frac{2}{16}, \frac{4}{16}, ..., \frac{8}{16}\right\} \tag{2.36}$$

For a frequency spectrum of a 1280 pixels horizontal resolution image (fmax< 640 Hz), the zero response frequencies are calculated as follows: For 8 pixels wide box filter:

$$f^z_1 = \left\{\frac{1}{8}, \frac{2}{8}, ..., \frac{4}{8}\right\}.1280 = \{160, 320, 480, 640\} \tag{2.37}$$

For 16 pixels wide box filter:

$$f^z_2 = \left\{\frac{1}{16}, \frac{2}{16}, ..., \frac{8}{16}\right\}.1280 = \{80, 160, 240, 320, 400, 480, 560, 640\} \tag{2.38}$$

Common zero response frequencies:

$$\begin{aligned} a \leq \frac{T_1}{2} \wedge b \leq \frac{T_2}{2} \Rightarrow \quad & f^z_1 \cap f^z_2 \neq \varnothing \\ \Rightarrow \quad f^z_{1,2} = \left\{\frac{1}{8}, \frac{2}{8}, ..., \frac{4}{8}\right\}&.1280 = \{160, 320, 480, 640\} \end{aligned} \tag{2.39}$$

Example 2. This example calculates the common zeros of two blur filters whose magnitudes (10 and 25) are not integer multiple of each other but have at least one common divisor, in particular 5. Calculation of zero response frequencies for PSF pairs:

$$\text{Given } T_1 = 10 \text{ pixel}, \quad T_2 = 25 \text{ pixel}$$

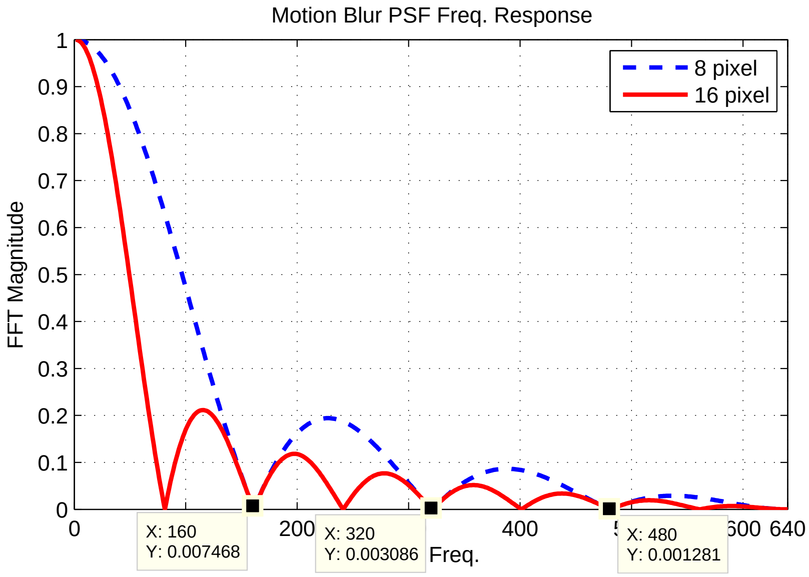


Figure 2.3: Frequency response graph of two PSFs whose magnitudes (8 and 16) are integer multiple of one another. Graph shows the common and individual zero response frequencies as calculated above.

For a frequency spectrum of a 1280 pixels horizontal resolution image (fmax< 640 Hz), the zero response frequencies are calculated as follows: For 10 pixels wide box filter:

$$f_1^z = \left\{ \frac{1}{10}, \frac{2}{10}, ..., \frac{5}{10} \right\} .1280 = \{128, 256, 384, 512\} \tag{2.40}$$

For 25 pixels wide box filter:

$$\begin{aligned} f_2^z &= \left\{ \frac{1}{25}, \frac{2}{25}, ..., \frac{12}{25} \right\} .1280 \\ &= \{51.2, 102.4, 153.6, 204.8, 256, 307.2, 358.4, 409.6, 460.8, 512\} \end{aligned} \tag{2.41}$$

Common zero response frequencies:

$$\begin{aligned} \frac{T_1}{T_2} = \frac{a}{b} = \frac{10}{25} \quad &\Rightarrow \quad a = 2, \;\; b = 5 \\ f_{1,2}^z = \left\{ \frac{2}{10}, \frac{4}{10} \right\} &.1280 = \{256, 512\} \end{aligned} \tag{2.42}$$

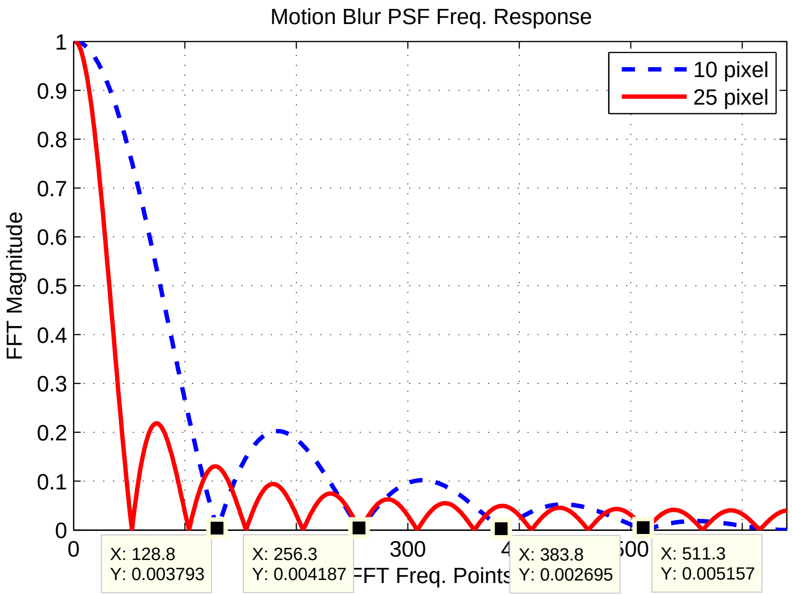


Figure 2.4: Frequency response graph of two PSFs whose magnitudes (10 and 25) are not integer multiple of each other but have at least one common divisor. Graph shows the common and individual zero response frequencies as calculated above.

Example 3. This example calculates the common zeros of two blur filters whose magnitudes (9 and 11) are relatively prime with each other. Calculation of zero response frequencies for PSF pairs:

$$\text{Given } T_1 = 9 \text{ pixel}, \quad T_2 = 11 \text{ pixel}$$

For a frequency spectrum of a 1280 pixels horizontal resolution image (fmax< 640 Hz), the zero response frequencies are calculated as follows: For 10 pixels wide box filter:

$$f_1^z = \left\{ \frac{1}{9}, \frac{2}{9}, ..., \frac{4}{9} \right\} .1280 = \{142.2, 284.4, 426.7, 568.9\} \tag{2.43}$$

For the 11 pixels wide box filter, the zero response frequencies are:

$$f_2^z = \left\{ \frac{1}{11}, \frac{2}{11}, ..., \frac{5}{11} \right\} .1280 = \{116.4, 232.7, 349.1, 465.5, 581.8\} \tag{2.44}$$

Common zero response frequencies:

$$\begin{aligned} \frac{T_1}{T_2} = \frac{a}{b} = \frac{9}{11} \quad &\Rightarrow \quad a = 9,\ \ b = 11 \\ a \nleq \frac{T_1}{2} \ \wedge \ b \nleq \frac{T_2}{2} \quad &\Rightarrow \quad f^z_{1,2} = \varnothing \end{aligned} \tag{2.45}$$

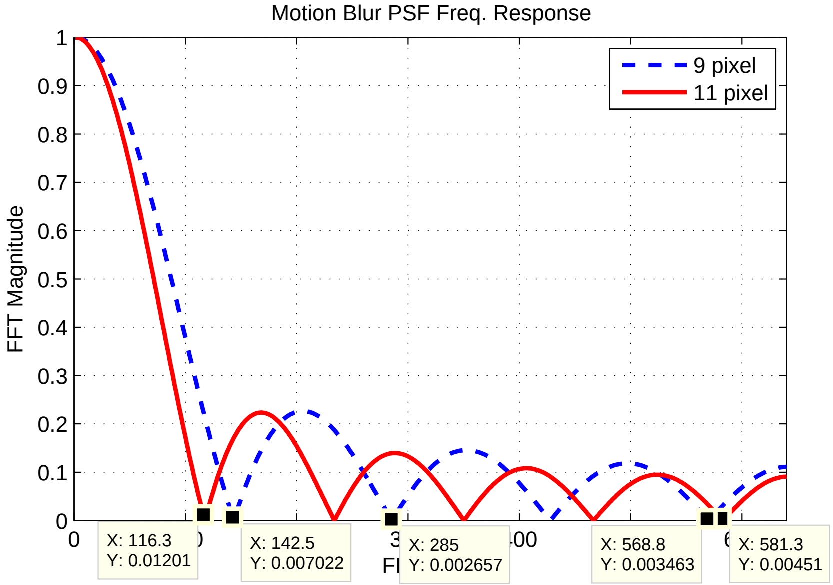


Figure 2.5: Frequency response graph of two PSFs whose magnitudes (9 and 11) are relatively prime with each other. Graph shows the common and individual zero response frequencies as calculated above.

2.3 Motion Blur in Legged Mobile Robotic Platforms

Our efforts on motion deblurring are ultimately aimed to reduce the motion blur problem in the SensoRHex robot platform. In parallel to our studies on multi-frame deblurring, we try to understand and simulate the motion blur formation on the SensoRHex robot which is the application area for motion deblurring. The motion blurring of images captured from a legged mobile robot platform shows some particular properties. The blur is caused by the interaction between the leg and the surface while the robot is walking. This surface-leg interaction induces some mechanical vibrations on the robot body and so on the camera. However these vibrations are not continuous and fade out with time. Therefore, the magnitude of motion blur changes while the robot walks. The analysis of image frames shows that there are captured both very sharp, very blurred frames and different level of blurring in-between. In this section, we explain projection of camera motion onto image plane based on IMU (Inertial Measurement Unit) data. Then we demonstrate the simulated motion blur for a given 6D camera motion.

2.3.1 IMU Based 6-D Spatially Varying Motion Blur PSF Computation and Simulation

The vibrations on the robot body and so as the camera may occur in different forms depending upon the direction and speed of the movement. Camera motions involving both translational and rotational motions yield a rich content of motion blur. The camera can do 6D motion. It can rotate around three axes and translate along three axes. This motion can be conducted one at a time as well as all combined together. As a result of this motion during the capture, there appears a motion blur effect in the image [3,4,6]. The aim of this section is to model the motion blur based on the camera motion and simulate the blurring process.

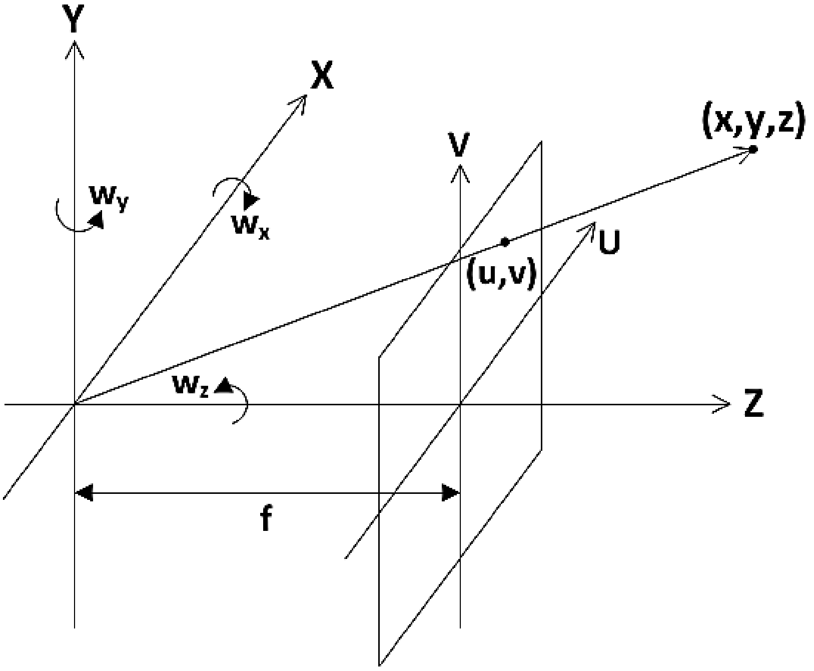


Figure 2.6: Coordinate axes definition for camera motion. Image plane is placed on a plane parallel to the x-y plane and with a perpendicular distance f from the origin. Optical axis is along the z axis.

The motion of the camera can be measured by an IMU device. According to this motion, the scene points will project onto different points on the image plane causing motion blur. We will utilize the pinhole camera model and follow the derivation in [34] to calculate the velocities of points on the image plane. Coordinate axes definition for camera motion is shown in Fig. 2.6. Image plane is placed on a plane parallel to the x-y plane and with a distance f from the origin. Optical axis is along the z axis. The roll, pitch and yaw rotations are defined around the z, x and y axes respectively. The translational motions are also defined with respect to this coordinate frame. The pinhole camera model utilize the following equation to transfer world coordinates (X, Y, Z) into the corresponding image plane coordinates (x, y),

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} f\dfrac{X}{Z} \\ f\dfrac{Y}{Z} \end{bmatrix} \tag{2.46}$$

where f denotes the focal length of the camera. Taking the derivative of the equation with respect to time yields the following equation which gives the velocities of image points.

$$\begin{bmatrix} u \\ v \end{bmatrix} = \frac{d}{dt}\begin{bmatrix} x \\ y \end{bmatrix} = f\begin{bmatrix} \dfrac{\dot{X}Z - X\dot{Z}}{Z^2} \\ \dfrac{\dot{Y}Z - Y\dot{Z}}{Z^2} \end{bmatrix} = \begin{bmatrix} \dfrac{f}{Z} & 0 & -\dfrac{fX}{Z^2} \\ 0 & \dfrac{f}{Z} & -\dfrac{fY}{Z^2} \end{bmatrix}\begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{Z} \end{bmatrix} \tag{2.47}$$

The world point velocities are actually zero. However the velocity is induced by the camera motion. Therefore, instead of moving the camera, we can move the world point so that the same effect appears on image plane. The velocity of a point in space caused by the roll, pitch and yaw axis rotations can be written as below.

$$\begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{Z} \end{bmatrix} = \begin{bmatrix} 0 & -\omega_Z & \omega_Y \\ \omega_Z & 0 & -\omega_X \\ -\omega_Y & \omega_X & 0 \end{bmatrix}\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \tag{2.48}$$

Substituting this into the image velocity equation yields,

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \omega_Z Y - \omega_Y f - \dfrac{\omega_Y}{f}X^2 + \dfrac{\omega_X}{f}XY \\ \omega_Z X - \omega_X f - \dfrac{\omega_Y}{f}XY + \dfrac{\omega_X}{f}Y^2 \end{bmatrix} \tag{2.49}$$

The above equation considers the velocity caused by rotation only. If the translational velocities are included in the above equation, then it becomes as follows.

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \omega_Z Y - \omega_Y f - \dfrac{\omega_Y}{f} X^2 + \dfrac{\omega_X}{f} XY - \dfrac{f}{Z} v_X + \dfrac{u}{Z} v_Z \\ \omega_Z X - \omega_X f - \dfrac{\omega_Y}{f} XY + \dfrac{\omega_X}{f} Y^2 - \dfrac{f}{Z} v_Y + \dfrac{v}{Z} v_Z \end{bmatrix} \tag{2.50}$$

Using the above equation, we compute the image plane velocities when 6D motion is known. Then we obtain the corresponding PSF for the image plane velocities. Fig. 2.7 shows the resultant velocity field when primary one axis translations are applied. The PSFs corresponding to these motions are obtained and used to simulate the motion blur. The resultant simulated images are given next to the velocity graphs. Note that, the latter figure has spatially varying motion blur PSF.

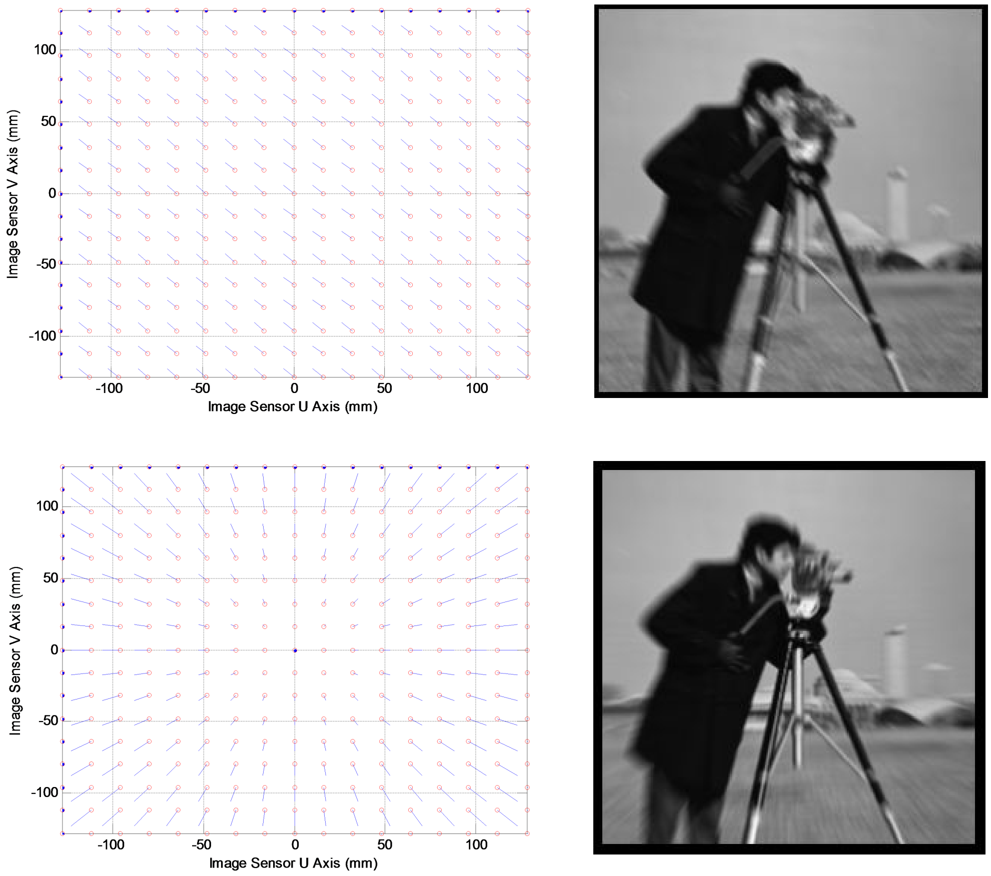


Figure 2.7: The resultant velocity field when translation along x and y axis are applied is shown on top graph and translation along z axis is shown on bottom graph. The PSFs corresponding to these motions are obtained and used to simulate the motion blur. The resultant simulated images are given next to the velocity graphs. Note that, the latter figure has spatially varying motion blur PSF.

Fig. 2.8 shows resultant velocity field when rotation around either of pitch, yaw and roll axes are applied. The PSFs corresponding to these motions are obtained and used to simulate the motion blur. The resultant simulated images are given next to the velocity graphs. Note that, all the figures have spatially varying motion blur PSFs, however the effect is seen dramatically on the bottom figure.

2.4 Estimation of 2-D Nonlinear Point Spread Function

The motion blur analysis presented above, models the motion blur as a linear PSF which can be represented by a single motion blur magnitude in pixels and the direction of the motion blur using a single angle value. This assumption limits the cases where the model is valid. When the motion of a camera on a legged mobile robot is considered, it is obvious that the camera does not have to move in a single direction. For example, it is very possible for the camera to move on a circular arc which can not be modeled accurately using the 1-D linear model. To be able to handle these cases, we extend the 1-D linear motion blur PSF assumption to be able to handle 2-D nonlinear motion blur cases which is frequently encountered in legged robotics vision applications.

The linear PSF is determined by calculating the beginning and ending position/orientation of the camera using the external motion tracking data as explained in Section 2.3.1. Instead of using the initial and final camera poses, the samples of the motion tracking data are integrated to get the trajectory of the camera during the exposure period. An example set of motion data are projected onto image plane using (2.50) and the resultant 2-D nonlinear pixel motion trajectory is shown in Fig. 2.9 on top. The corresponding 2-D nonlinear PSF is computed and shown at the bottom of the same figure.

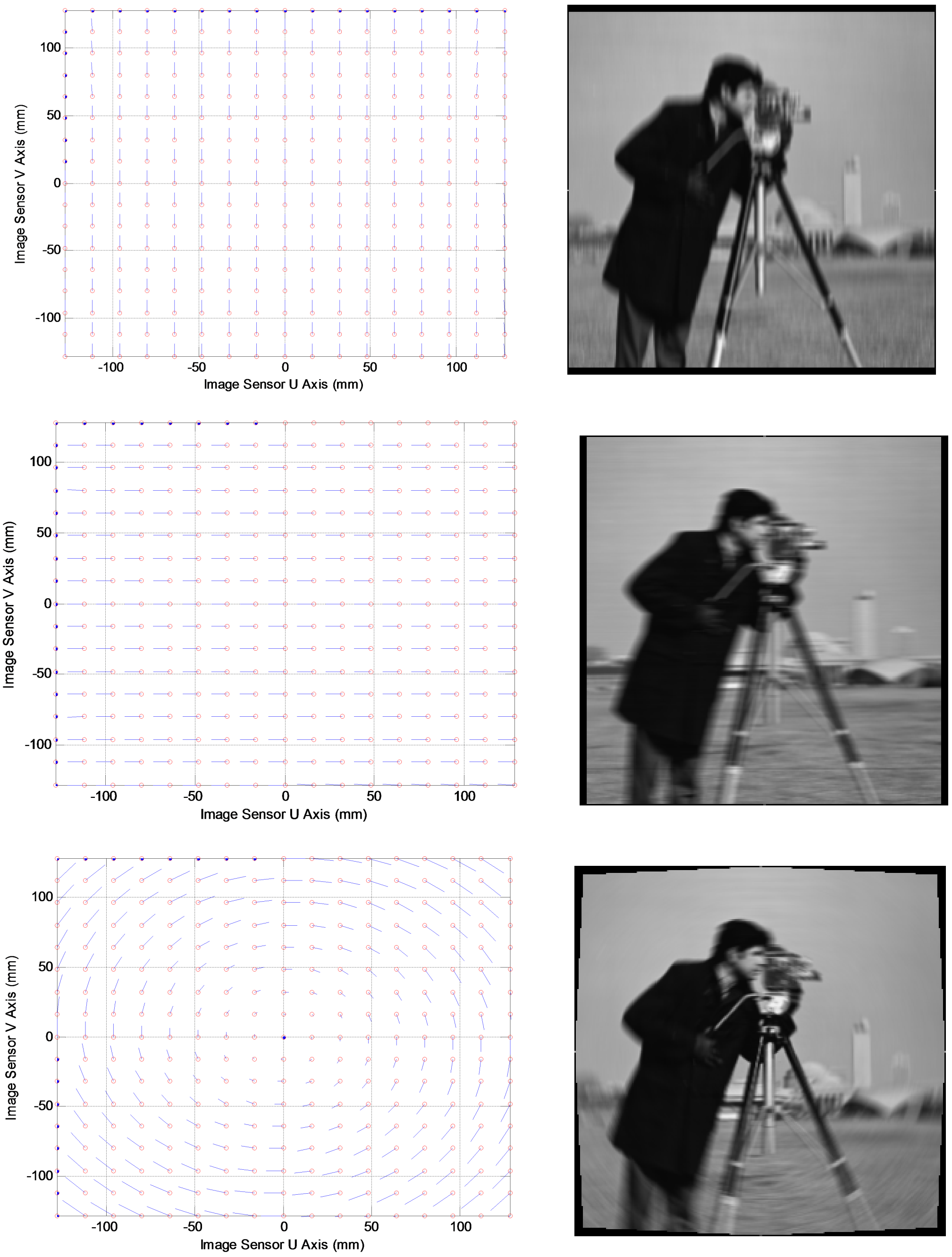
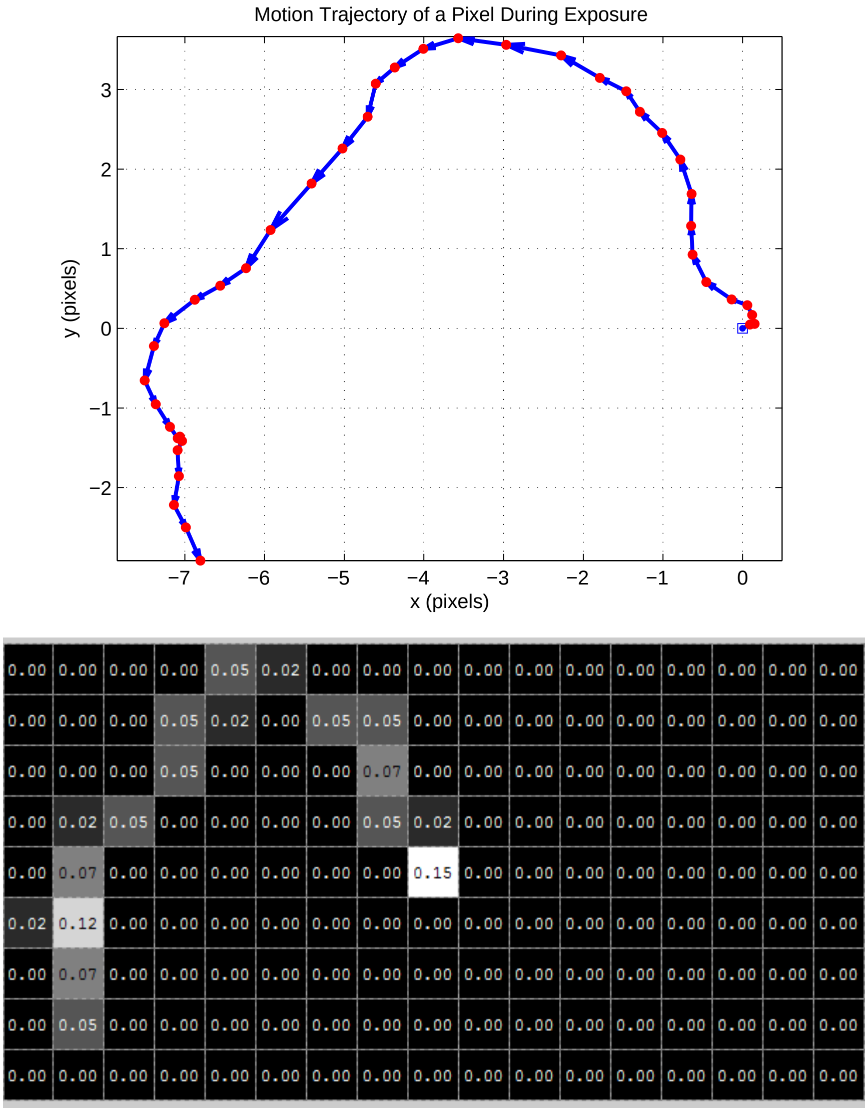


Figure 2.8: The resultant velocity field when rotation around either of pitch, yaw and roll axes are applied is shown top, center and bottom figures respectively. The PSFs corresponding to these motions are obtained and used to simulate the motion blur. The resultant simulated images are given next to the velocity graphs. Note that, all the figures have spatially varying motion blur PSFs, however the effect is seen dramatically on the bottom figure.



0.00	0.00	0.00	0.00	0.05	0.02	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.05	0.02	0.00	0.05	0.05	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.05	0.00	0.00	0.00	0.07	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.02	0.05	0.00	0.00	0.00	0.00	0.05	0.02	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.07	0.00	0.00	0.00	0.00	0.00	0.00	0.15	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.02	0.12	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.07	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.05	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Figure 2.9: (Top) 2-D nonlinear pixel motion trajectory using a set of motion data which are projected onto image plane using (2.50). (Bottom) The corresponding 2-D nonlinear PSF.

CHAPTER 3

MOTION BLUR ASSESSMENT AND FEATURE BASED IMAGE QUALITY METRIC FOR COMPUTER VISION APPLICATIONS

3.1 Available Metrics for Image Quality

Visual quality assessment is an important field for performance comparison of image processing algorithms. They can depend on either human opinion based subjective assessment or a mathematical processing of image which is referred as objective metrics, to compute a quality score. Objective metrics can be grouped into two as referenced or no reference metrics. Some of the image quality assessment metrics which are available in literature are as follows. The ones we consider in this study are especially explained in more detail.

i. **MSE (Mean Squared Error)** is a pixelwise error metric computed using the differences of reference image pixels $x_{j,k}$ and the degraded image pixels $x_{j,k}$ as given in (3.1).

$$MSE = \frac{1}{MN} \sum_{j=1}^{M} \sum_{k=1}^{N} \left(x_{j,k} - x'_{j,k} \right)^2 \ , \tag{3.1}$$

where the image consists of $M \times N$ pixels. Images having lower MSE scores are interpreted to have better quality.

ii. **PSNR (Peak Signal to Noise Ratio)** [19] is a metric for the ratio between the maximum possible power value of a signal and the power of distorting noise that affects the quality of the signal. It is computed based on the MSE expression as given in (3.2).

$$PSNR = 10 \log \frac{(2^n - 1)^2}{MSE} \tag{3.2}$$

The PSNR is expressed in logarithmic scale since most of the signals have a large dynamic range. Therefore, the unit of PSNR is expressed in decibells(dB). Images having higher PSNR scores are interpreted to have better quality.

iii. **NCC (Normalized Cross Correlation)** [54] is an image correlation based metric as given in (3.3).

$$NCC = \frac{\sum_{j=1}^{M}\sum_{k=1}^{N}\left(x_{j,k}\,.\,x'_{j,k}\right)}{\sum_{j=1}^{M}\sum_{k=1}^{N}\left(x_{j,k}\,.\,x_{j,k}\right)} \tag{3.3}$$

iv. **SSIM (Structural Similarity Index)** [54] is a reference based metric that measures the similarity of an image to a reference image. It is computed on windows defined by x and y of common size $N \times N$. The computation is based on three terms which are luminance (l), contrast(c) and structure (s) as shown in equations given below where, μ_x, μ_y, σ_x, σ_y and σ_{xy} are the local means, standard deviations, and cross-covariance for x and y.

$$l(x,y) = \frac{2\mu_x\mu_y + c_1}{\mu_x^2 + \mu_y^2 + c_1} \tag{3.4}$$

$$c(x,y) = \frac{2\sigma_x\sigma_y + c_2}{\sigma_x^2 + \sigma_y^2 + c_2} \tag{3.5}$$

$$s(x,y) = \frac{\sigma_{xy} + c_3}{\sigma_x\sigma_y + c_3} \tag{3.6}$$

The SSIM metric is a weighted combination of the terms given in (3.4), (3.5) and (3.6).

$$SSIM(x,y) = \left[l(x,y)^{\alpha} \cdot c(x,y)^{\beta} \cdot s(x,y)^{\gamma}\right] \tag{3.7}$$

If the exponents α, β and γ are set to 1 and $c_3 = c_2/2$, then the metric simplifies as given in (3.8). The SSIM score can get value between -1 an 1 where 1 corresponds to the ideal case(identical images).

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_x\sigma_y + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \tag{3.8}$$

v. **NRMSE (Normalized Root Mean Squared Error)** is a normalized error measure computed using the differences of reference image pixels $x_{j,k}$ and the degraded image pixels $x_{j,k}$ as given in (3.9). The second expression is for normalization where x_{max} and x_{min} are the maximum and minimum values of image pixels respectively.

$$RMSE = \sqrt{\frac{\sum_{j=1}^{M} \sum_{k=1}^{N} \left(x_{j,k} - x'_{j,k}\right)^2}{M.N}},$$
$$NRMSE = \frac{RMSE}{x_{max} - x_{min}} \tag{3.9}$$

vi. **MAE (Mean Absolute Error)** is another pixelwise error metric computed using the absolute differences of reference image pixels $x_{j,k}$ and the degraded image pixels $x_{j,k}$ as given in (3.12).

$$MAE = \frac{1}{MN} \sum_{j=1}^{M} \sum_{k=1}^{N} \left| x_{j,k} - x^{'}_{j,k} \right| \tag{3.10}$$

vii. **UQI (Universal Quality Index)** is a structural similarity based index which is similar to SSIM. The details of the measure can be found in [53].

viii. **VSNR (Visual Signal to Noise Ratio)** is a wavelet-based visual signal to noise ratio for natural images which is explained in [9] in detail.

ix. **MOS / DMOS (Difference Mean Opinion Score)** are subjective metrics that are based on the human users view of the quality.

x. **PQR (Picture Quality Rating)**

xi. **VIF (Visual Information Fidelity)** assumes that Human Visual System (HVS) perceives natural images and therefore instead of modeling HVS, this metric is defined using a natural image statistics based model. The details of the measure can be found in [49].

The common property of the above assessment methods is that, they all try to measure the visual quality of an image. As we have discussed earlier, in robotics or other machine vision applications, the end user of the captured image frames are not a human but the machine code that is used for pattern recognition, tracking, etc. Most of these machine vision algorithms depend on the performances of feature detection algorithms. Therefore, from the view of machine vision, the image quality is (or should be) measured as the ability to successfully detect the features. One may argue that some machine vision algorithms have improved performances even on low quality images. However, this does not mean that those methods do not yield better performances on images where a more accurate feature extraction can be achieved. Therefore in general, we can propose that more accurate feature detection will improve the performances of higher level processing algorithms.

Most of the deblurring methods proposed in literature are demonstrated on a limited number of images. The performance evaluation is generally left to reader by means of a visual inspection of the deblurred images. This is a very subjective and error prone approach even if results are shared on-line and not only in print. Only a few of the studies report their results using an objective quality metric such as PSNR or MSE. These metrics are computed using pixel by pixel difference between a reference image and a corrupted or recovered image as given in (3.1) and (3.2) where $x_{j,k}$ represents the pixel value at location (j, k) of the reference image while $x'_{j,k}$ is the one of the test image, M and N are the horizontal and vertical dimensions of the images and n is the number of bits used to represent the image pixel values. These metrics are frequently used (and suitable) for measuring the error in image coding/decoding, compression/decompression applications. We believe that they are not for our application focus of computer vision and image understanding.

3.2 Available Metrics for Image Sharpness

Detection of motion blur magnitude, in other words how much an image is blurred, is generally a common pre-process in most of the applications related with motion blur. There are many methods proposed in literature to determine the motion blur magnitude. However, each method has different pre-assumptions and has its own disadvantages when compared to each other. Therefore, the method to use should be selected based on the application.

According to the obtained video frames from the robot camera, it is observed that the frames are exposed to motion blur with various amounts. Some of the frames are highly exposed to motion blur whereas some of the others are very sharp i.e. without any motion blur. Therefore, the video frames should be classified according to the amount of motion blur before deblurring process. According to this classification result, the frames that are not exposed to considerable motion blur will be ignored and the other motion blurred frames will be considered for deblurring process.

There are a number of available motion blur measures proposed in the literature. Most of these measures are based on the idea to emphasize high frequencies of the image and measure their quantity. It corresponds with our intuitive expectation that the blurring suppresses high frequencies. One of the simplest blurring measures is calculated using the integral of variance of image gray levels over the whole image frame which can be formulated as

$$M_1 = \int\int \left(g_i\left(x,y\right) - m_i\right)^2 dxdy \tag{3.11}$$

where m_i denotes the mean gray level value of $g_i(x, y)$ which represents the 2D image signal of the i^{th} frame. This measure is monotonic and, it is equivalent to the image energy in Fourier domain for zero-mean images.

$$\int\int \left|G_i\left(u,v\right)\right|^2 dudv \tag{3.12}$$

In the graphic given in Fig. 2, the blurring of an example image sequence using image variance as a measure is shown. Peaks indicate sharper frames and valleys indicate blurred frames.

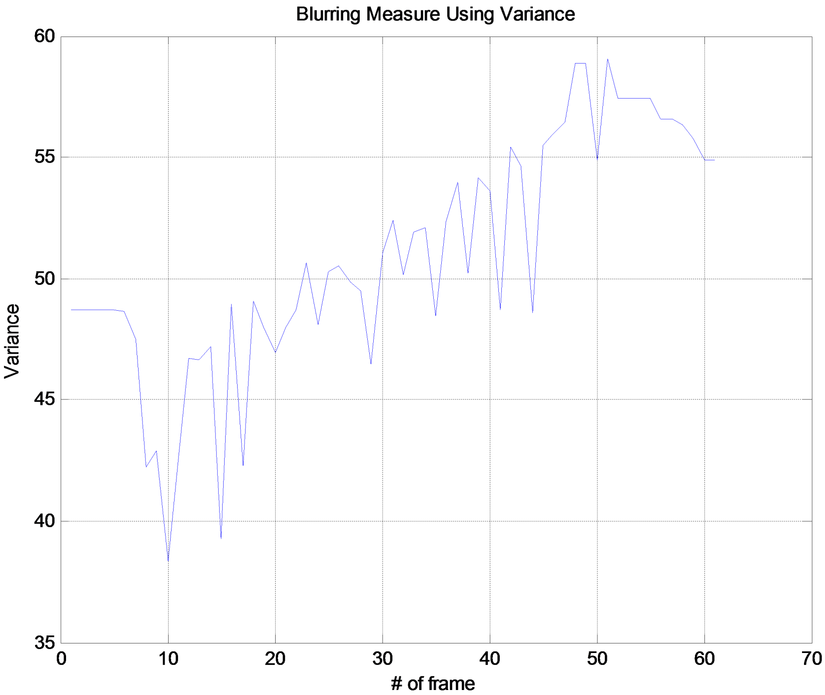


Figure 3.1: Blurring quantity of an image sequence captured from the camera on the SensoRHex robot using the variance measure.

Several other blur measures are defined by means of image derivatives. We know that gradient gives us the rate of change of the signal. For sharp images, the rapid transitions of brightness on the edges yield high amplitude of gradient. Using the sums of L1 norm of image gradient, the following expression gives a scalar value which is correlated with the sharpness of the image.

$$M_2 = \int\int \left|\frac{\partial g_i(x,y)}{\partial x}\right| + \left|\frac{\partial g_i(x,y)}{\partial y}\right| dxdy \tag{3.13}$$

where M2 is higher for a sharper image and decreases proportional to the increasing motion blur effect.

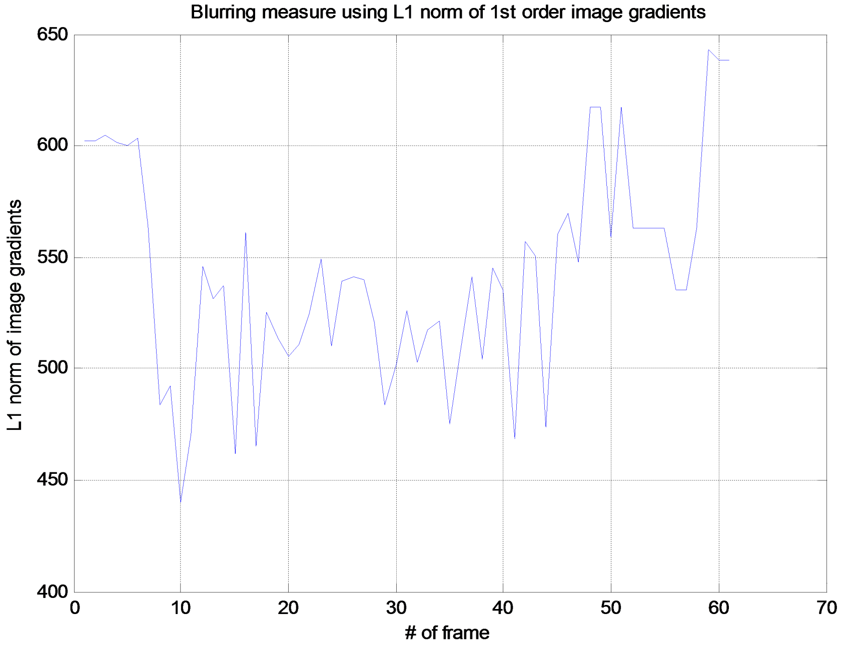


Figure 3.2: Blurring quantity of an image sequence captured from the camera on the SensoRHex robot using the L1 norm of the 1st order image derivatives as a measure.

L2 norm of image gradient (sometimes called gradient energy) is another derivative based measure of image blurring given by the following expression.

$$M_3 = \int\int \left(\frac{\partial g_i(x,y)}{\partial x}\right)^2 + \left(\frac{\partial g_i(x,y)}{\partial y}\right)^2 dxdy \tag{3.14}$$

Instead of first derivatives, another alternative is to use the second order derivatives. The following blur measure computes L1 norm of second derivatives.

$$M_4 = \int\int \left|\frac{\partial^2 g_i(x,y)}{\partial x^2}\right| + \left|\frac{\partial^2 g_i(x,y)}{\partial y^2}\right| dxdy \tag{3.15}$$

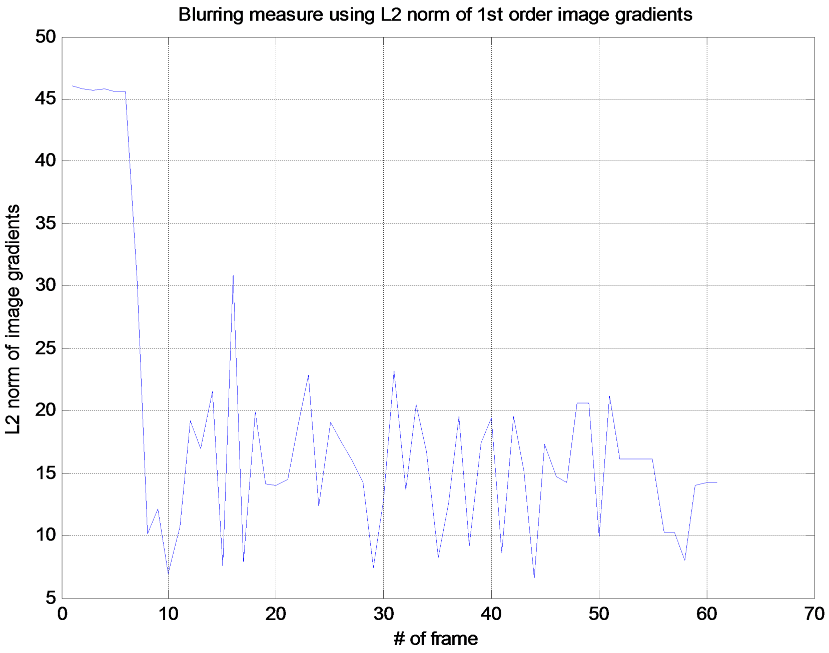


Figure 3.3: Blurring quantity of an image sequence captured from the camera on the SensoRHex robot using the L2 norm of the 1st order image derivatives as a measure.

Energy of image Laplacian can also be used as a blur measure. Using the following expression, M5 gives an approximate measure for the image blurring.

$$M_5 = \int\int\left(\frac{\partial^2 g_i\left(x,y\right)}{\partial x^2} + \frac{\partial^2 g_i\left(x,y\right)}{\partial y^2}\right) dxdy \tag{3.16}$$

3.3 Proposed Standard Deviation Based Motion Blur Measurement of an Image Within an Image Sequence

The motion blur detection methods in literature can be analyzed mainly in two groups according to the reference. The first group of methods assumes that there is only one motion blurred frame of the scene. These methods try to extract the motion blur using only the single blurred image itself. These methods are referred as “no-reference” methods since there is no reference frame available which represents the sharp snapshot of the considered scene. A popular method in this group is based on the just noticeable blur approach proposed in [13]. This metric is generally used to compare visual quality and it is applicable for relatively small blur amounts. An-

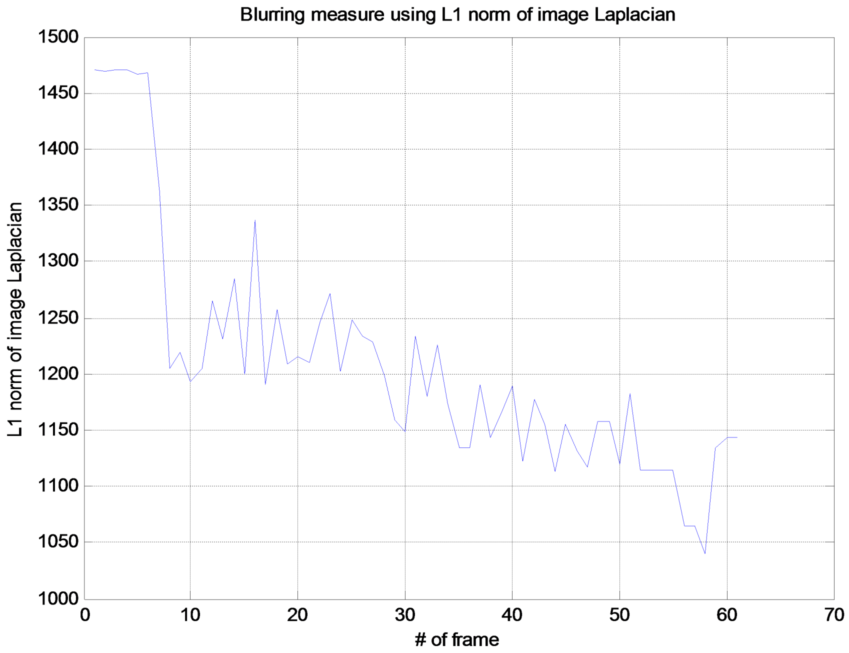


Figure 3.4: Blurring quantity of an image sequence captured from the camera on the SensoRHex robot using the L1 norm of the image Laplacian as a measure.

other no-reference metric is presented in [35]. Their metric is based on a cumulative probability of blur by taking into account the Human Visual System. In [33] a blur metric based on the analysis of the spread of the edges in an image is given. Wavelet transform coefficients are utilized in [55] claiming that blurring disrupts the phase coherence. The study in [58] utilizes contrast based blur invariant features together with local standard deviations to determine spatially varying blur in images. In [29], the authors utilize the gradient profile sharpness of image edge and the histogram of sharpness distribution. In [51], they consider both the defocus and motion blur artifact. They use local edge blur to estimate the overall image blur. Although this group of methods has more application area because of the lack of reference need, their performances depend much on the scene type since there is no information available about the sharp representation of the scene.

The second group of methods depends on the knowledge of a reference frame which is not exposed to any motion blur and therefore a sharp representation of the scene. Because of the use of a reference frame, this group of methods is called "reference based" methods. These methods try to determine the amount of motion blur that causes the reference scene to appear as the motion blurred frame captured of that scene. Generally these methods perform better than the no-reference based methods since there is more information available about the scene. However, there are fewer studies in this group available in literature since the cases where the reference frame is available are rare. An important study in this group is presented in [36]. They assume the existence of global motion blur and a sharp reference frame. They use linear least squares fit approach in a wavelet-based Haar filters and claim the real-time operation of the algorithm. A study on the motion blur removal for humanoid robots is presented in [28]. They first classify the images as less blurred and severely blurred by using Just Noticeable Blur Metric [13] as a quantitative criterion. For less blurred images, they propose a maximum a posteriori framework by taking advantage of the previous sharp image as reference. Our proposed metric can be classified in the second group as a reference based method.

For the analysis of the image texture in the point of blur amount, several methods can give information on the high frequency content available in the image data. A motion blurred image is poor in high frequency components since they are smoothed out. Therefore analyzing the statistical properties of the texture of the images captured from the same scene can give information on which one is sharper. In development of this metric, a synthetic checkerboard pattern is first used as an experimental input image since it has both horizontal and vertical sharp edges and it is easier to find the ground truth blur amount manually.

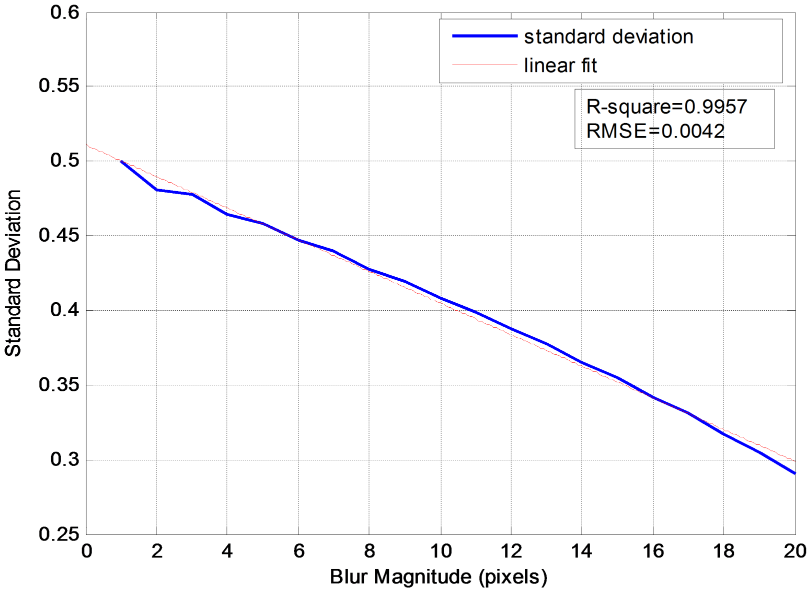


Figure 3.5: Variation on blur magnitude (pixels) versus standard deviation of the synthetically generated checker board pattern. .

R-square value and the root-mean square error value (RMSE) are considered for measuring the linearity of the proposed metric with respect to the blur magnitude criteria by fitting a linear line. It can be seen that standard deviation ensures an R-square value of 0.9957. Therefore, being founded of the assumptions, it can be concluded that standard deviation function has a good linearity relation with respect to the motion blur magnitude. The equation that is used in the calculation of standard deviation is given by the following expression.

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2} \tag{3.17}$$

The calculation of any motion blur, mainly the magnitude of the blur in both directions, is performed with the assumption that the standard deviation of the motion blurred image changes linearly as a new, synthetic filter is applied on, as previously mentioned. The algorithm used in this study predicts the blur amount based on a comparison of each frame to a reference frame, i.e. the sharpest frame, in a window of frames containing a predefined number of frames taken in time.

Before we continue with the estimation of the blur the camera is exposed to, let us mention how the sharpest frame is selected among the others in a time window of n frames. The image that the program chooses as the reference image in a time slot is simply the image which has the highest standard deviation in the window that is examined. During the testing of the operation of the function, we used a fixed time window of 10. Therefore, inside of the window, the reference image chosen for each image is unique and the reference input changes as the window slides in time. One drawback of this kind of operation is that during the real-time operation it needs several image frames to be buffered to find the reference frame. This means that the algorithm calculates the blurring of a frame only after n previous frames are available in the buffer. This of course, is related only to the choice of the time window, so the type of application and can be adjusted accordingly.

After determining the sharpest image taken in a fixed time window, the function utilizes the valid assumption of linearity of standard deviation against motion blur amount. At this stage, the algorithm fits a linear function to the graph of standard deviation versus motion blur magnitude. At the next step, it determines the inverse function of the line obtained, i.e. the function from the set of standard deviations to the set of motion blur amount, called the motion blur function which is linear by definition.

At the last step, the function estimates the blur amount of any non-reference image coupled to that reference image as the output of the motion blur function. The input variable of the metric is the calculated standard deviation of the considered blurred image. valid assumption that the images, prior or posterior to the coupled references, are alike in terms of content and therefore can be used as a starting point for the calculations.

As a result, our blur metric separates the operation time into frame window of a given length and determines the sharpest frames for the frames in that window to use them

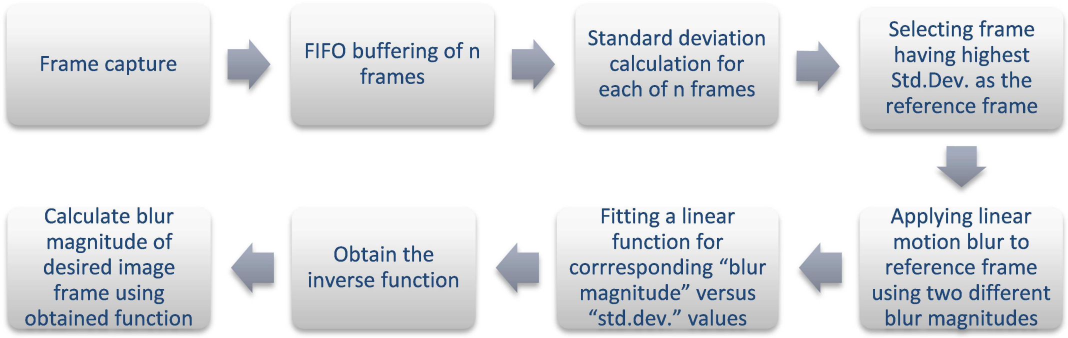

Frame capture
FIFO buffering of n frames
Standard deviation calculation for each of n frames
Selecting frame having highest Std.Dev. as the reference frame
Applying linear motion blur to reference frame using two different blur magnitudes
Fitting a linear function for corrresponding “blur magnitude” versus “std.dev.” values
Obtain the inverse function
Calculate blur magnitude of desired image frame using obtained function

Figure 3.6: Flow chart of the proposed motion blur detection method.

as reference. Then, blurs each reference image to determine a linear relation between the blur amount and change in standard deviation and refers to this relation estimating the blur amount of any image. The steps of the algorithm are given by the flow chart in Fig. 3.6.

In order to gain an insight into the motion of the legged robot, we need to observe the amount of the motion blur by calculating the standard deviation of the blurred images. Figure 3 demonstrates the variation of standard deviation for 50 consecutive images captured at 25 fps from the camera mounted on the legged robot platform. Coarsely, it can be said that frames having higher standard deviation are relatively sharper than the frames having lower standard deviation values. From the graph, it can be realized that there are both sharp and blurred images in the video sequence obtained while the robot is walking in a periodic manner. The sharp frames can be used as a reference frame representing the non-blurred representation of the scene. Actually, the reference of each motion blurred frame is different. However, the consecutive image frames are not changing too much with respect to its neighboring frames. Therefore, it can be assumed that a reference frame is valid among its neighboring frames collected in a window. The window size should be selected properly so as to both there exists a sharp frame to be selected as the reference frame and the reference frame still remains valid for all the other frames in the window. In our case, we selected a window of 10 consecutive image frames. Therefore, this observation will shape our approach during the calculation steps.

3.3.1 Justification of Standard Deviation Based Motion Blur Measurement on a Legged Robot Image Sequence

After constitution of theoretical structure and enhancement of the operation of the function, it is tested on the consecutive image frame samples taken from a camera mounted on a legged robot. Comparison between theoretical and real values of blur

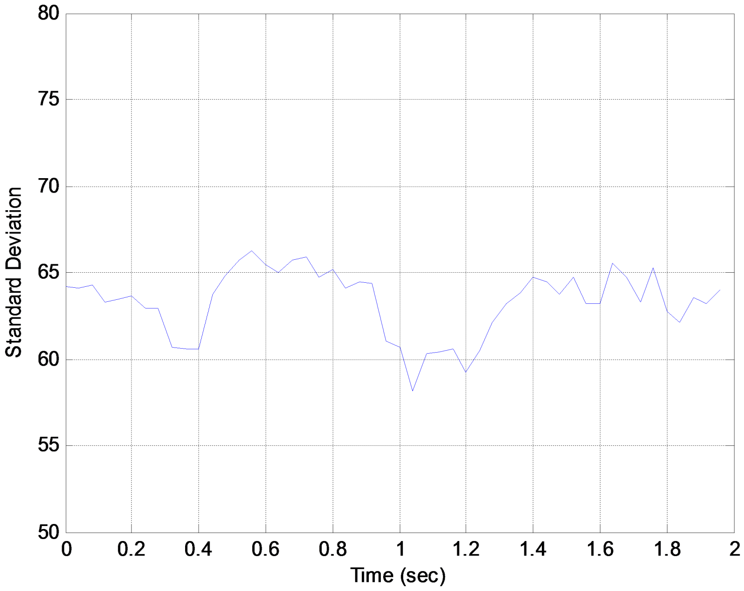
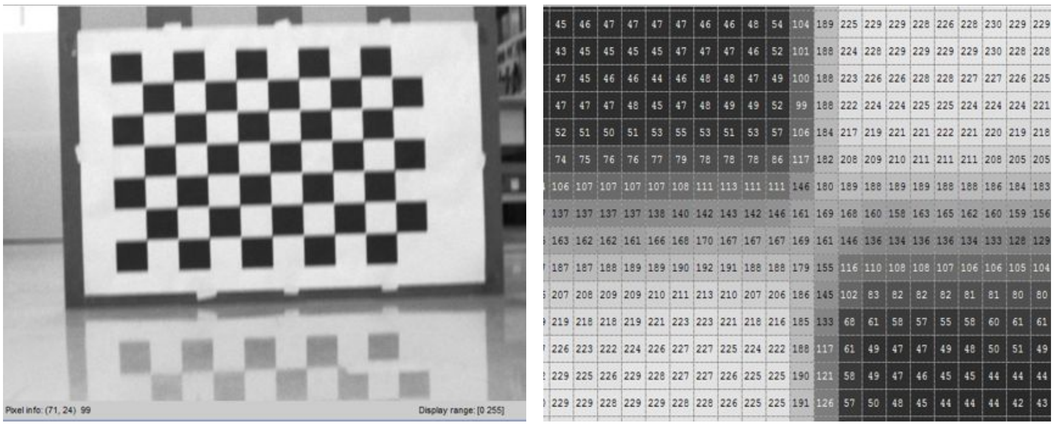


Figure 3.7: Standard deviation vs. time of the 50 consecutive images captured from a camera mounted on a legged mobile robot platform.

amount for two different images which are selected randomly is as follows.

The experimental blur magnitude of the first randomly chosen image is calculated as 7.9 pixels while the corresponding ground truth value is 8.9 pixels. The image frame is visualized in Figure 5 together with a magnified image patch which shows the blurring of pixels around the edge.



45	46	47	47	47	47	46	46	48	54	104	189	225	229	229	228	226	228	230	229	229
43	45	45	45	45	47	47	47	46	52	101	188	224	228	229	229	229	229	230	228	228
47	45	46	46	44	46	48	48	47	49	100	188	223	226	226	228	228	227	227	226	225
47	47	47	48	45	47	48	49	49	52	99	188	222	224	224	225	225	224	224	224	221
52	51	50	51	53	55	53	51	53	57	106	184	217	219	221	221	222	221	220	219	218
74	75	76	76	77	79	78	78	78	86	117	182	208	209	210	211	211	211	208	205	205
106	107	107	107	107	108	111	113	111	111	146	180	189	188	189	189	188	188	186	184	183
137	137	137	137	138	140	142	143	142	146	161	169	168	160	158	163	165	162	160	159	156
163	162	162	161	166	168	170	167	167	167	169	161	146	136	134	136	136	134	133	128	129
187	187	188	189	189	190	192	191	188	188	179	155	116	110	108	108	107	106	106	105	104
207	208	209	209	210	211	213	210	207	206	186	145	102	83	82	82	82	81	81	80	80
219	218	218	219	221	223	223	221	218	216	185	133	68	61	58	57	55	58	60	61	61
226	223	222	224	226	227	227	225	224	222	188	117	61	49	47	47	49	48	50	51	49
229	225	226	229	228	227	227	226	225	225	190	121	58	49	47	46	45	45	44	44	44
229	229	228	229	229	228	228	226	225	225	191	126	57	50	48	45	44	44	44	42	43

Figure 3.8: First randomly chosen image taken from camera mounted on a legged robot platform and its blown-up version. The experimental blur magnitude is calculated as 7.9 pixels while the ground truth value is 8.9 pixels.

The experimental blur magnitude of the second chosen image is calculated as 1.69

pixels while the corresponding original value is 2.23 pixels. The image frame is visualized in Figure 6 together with a magnified image patch which shows the blurring of pixels around the edge.

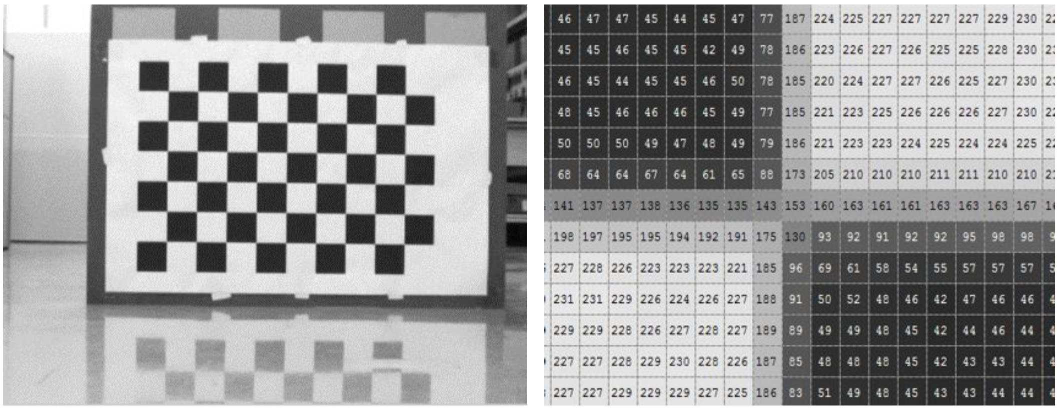


Figure 3.9: Second randomly chosen image taken from camera mounted on a legged robot platform and its blown-up version. The experimental blur amount value is 1.69 pixels while the original value is 2.23 pixels.

The mean error and the standard deviation of error of the metric are provided as criteria of accuracy and preciseness. The mean error is calculated as 1.2598 while the standard deviation of error is found as 2.9042.

The theoretical and real values of blur amount taken from a random frame window consisting of 20 consecutive frames are provided in Fig. 3.10.

After the implementation of metric to the consecutive image frame samples taken from a camera (Point Grey Flea2 640x480 resolution) mounted on a legged robot, it is observed that the metric definitely give a deep understanding about the comparison of the blur amounts of frames between each other. However, some errors occur in the computation of blur amount globally as can be seen in Fig 7. These errors can be minimized by proper selection of the size of frame window. The window size must be chosen such that it should be as large as possible to include a sharp frame for the sake of accuracy of the calculated motion blur magnitudes. On the other hand, it must also be not so large since the comparison of standard deviations of two images which are too much far from each other is meaningless. In this study, the window size is kept at a selected constant value. However, using an adaptive window size can improve the accuracy of the results. Another approach may be to utilize more than one reference frame to reduce the dependence on the proper selection of a single reference.

Regarding the computation time of the algorithm, the tests on a 1.73GHz computer having 1 GB RAM have shown that for a file containing 10 test images, the time interval the first results are computed, i.e. the first estimations appear, is from 0.8s to 1.1s. After characterizing the time window by its sharpest image and motion blur

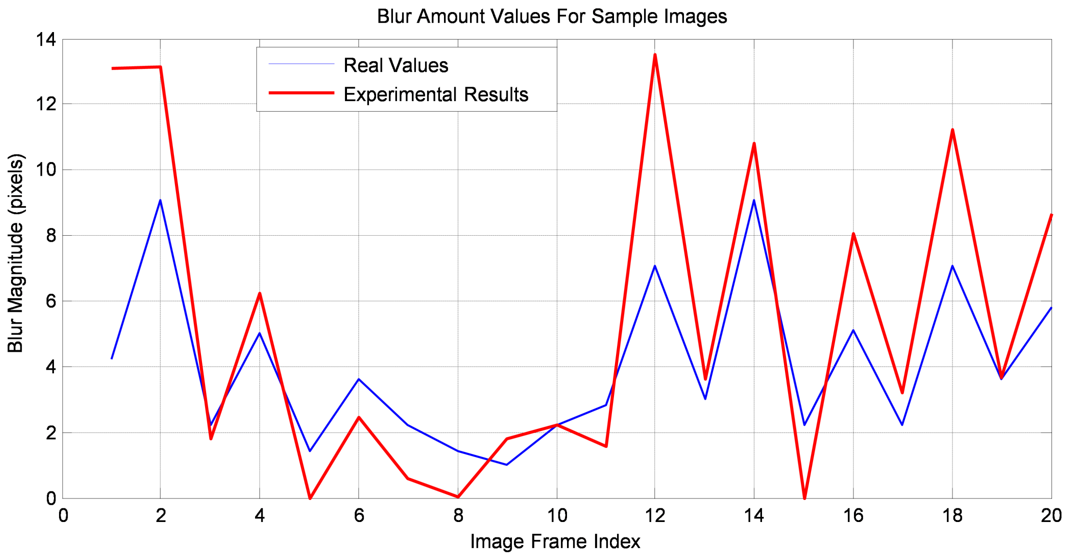


Figure 3.10: Comparison of experimental and real values of blur magnitude for a randomly selected image frame index which are taken from images captured from a camera mounted on a legged mobile robot platform.

function it takes about 30ms to compute each frame's amount of motion blur estimation separately. This shows that the metric can be computed in real-time even on a computer having low computational resources.

3.4 Proposed Feature Detection Based Image Quality Metric for Computer Vision Applications

In any study where image distortion is involved, one needs to define an image quality metric to assess the loss/recovery of image information. Similarly, the assessment of motion blurred and deblurred images need a quality metric to assess the outcome of any effort to recover distorted images. The definition of such a metric highly depends on the intended application. In the literature, widely used quality metrics are mostly based on subjective visual assessments or based on pixel-wise distortion measures. However, computer vision applications such as robot vision require a drastically different approach. Since vision algorithms rely on accurate detection of image features such as "corners", "edges" or other higher dimensional features, one needs an image quality metric that is specifically based on feature detection performance. The proposal of such a metric for the evaluation of the effects of motion blur and single-image deblurring algorithms is our focus in the present study.

It is known that the motion blur effect causes an information loss in the image content especially in the high frequency components. It also deteriorates the performances of feature detection methods such as corner features which generally depend on the spatial relation of neighboring image pixels. The corner features that are detected on an example sharp, blurred and deblurred images are given in Fig. 3.11. The performance loss of feature detection methods reflect negatively to the higher level computer vision algorithms that utilize them. Motion deblurring methods can be used as a workaround in order to recover blurred images and hence increase the performance of such algorithms. On the other hand, while motion deblurring helps recovering some of the image content, it may also introduce some specific distortions such as ringing which may result in some true image features to get lost and some false features to appear. Therefore, the relevant performance of a motion deblurring method in the field of computer vision and in particular for robotic applications should be measured in terms of the improvement of the feature detection performance. In the following discussion, we lay out a proposal for such a measure and use it for a comprehensive experimental study.

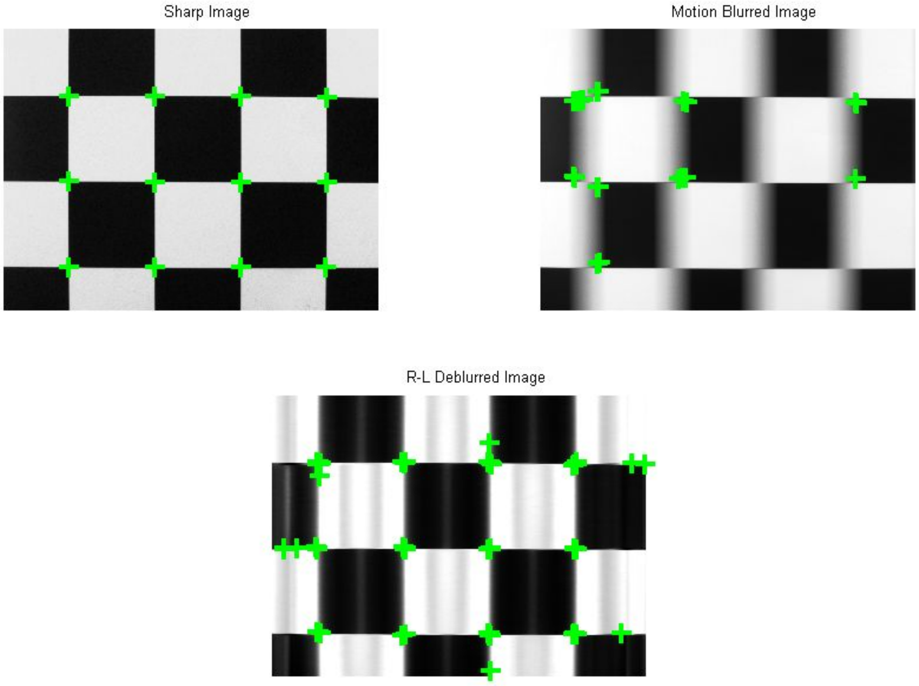


Figure 3.11: (a)Corners detected on sharp image. (b)Corners detected on blurred image. (c)Corners detected on deblurred image.

3.4.1 "Receiver Operating Characteristic" for Feature Detection Performance

Feature detection methods, as most other detection algorithms, generally have a sensitivity parameter such as a threshold value that effects the number of detected features on an image. There is often a trade-off: Changing this parameter may result in more true positives(TP) with accompanied more false positives(FP). Alternatively, the parameter can be adjusted to reduce FPs at the expense of missing some TPs, reducing detection performance. Hence, it is generally not sufficient to test the output of a feature detector for a single sensitivity parameter value. In order to discuss the benefit of a deblurring method on a feature detector, it should be tested with a set of parameters. The results of this variable parameter analysis are generally referred as ROC(Receiver Operating Characteristic) Curves and frequently utilized in RADAR and Pattern Recognition literature. We adopt this method to characterize motion deblurring methods based on the resulting performance of feature detectors.

A ROC analysis is mainly the determination of samples of a probability relationship curve, namely samples of the True Positive Rate(TPR) as a function of the False Positive Rate (FPR) parametrized over a set of sensitivity parameter values. Originally, the TPR and FPR are defined as given in (3.18) and (3.19) respectively. TPR and FPR can take values over the interval $[0\ 1]$.

$$TPR = \frac{TP}{TP + FN} \tag{3.18}$$

$$FPR = \frac{FP}{FP + TN}, \tag{3.19}$$

In our terminology; TP can be defined as the number of correctly detected features, False Negatives(FN) is the number of features that can not be detected although there is, FP is the number of false alarms and True Negatives(TN) is the number of features that are not detected as expected. However, the use of TN is not very appropriate for this application. A medium sized image includes millions of pixels and the number of features are generally not more than hundreds. Therefore the rest of the pixels should be referred as TN which is not very meaningful. Therefore, instead of using the above definition of FPR, we stick to the FPR definition given in [21] which is more suitable to this application. Current FPR definition is given in (3.20).

$$FPR = \frac{FP}{TP + FP} \tag{3.20}$$

The ground truth corners are considered as the corner points of checkerboard pattern on sharp images. Due to the motion blur, a corner point may sometimes appear as two splitted corner points. We call this phenomena a "Diplopia" effect. A detected corner point is considered as a TP if the location of it is in the "vicinity" of the ground truth corner location. The "vicinity" here is defined as a circular region whose center is on the ground truth corner location and its diameter is equal to the maximum motion blur magnitude on that image.

Using the TPR definition in (3.18) and FPR definition in (3.20), we propose a feature detection ROC curve based performance analysis of motion deblurring methods. ROC curve analysis indicate a comparative performance evaluation of detection methods for a set of possible operating points defined with variable thresholds or other algorithmic parameters. Instead of comparing the methods at a single operating point such as for a given Constant FPR (this is called as the CFAR technique [41]), ROC analysis provides an overall performance evaluation over a range of operating points. In ROC analysis, the curve that travels closer to the upper-left corner of the graph range (an ideal point of unity TPR and zero FPR) is considered to perform better overall. This provides a higher TPR with a lower FPR as cost. Therefore, the area under the ROC curve is frequently used as the performance indicator for detection or classification methods. In this study, we use the area under the feature detection ROC curve and name our performance metric as “FD-AROC”. Using this performance metric, we conduct a thorough comparative analysis of how different deblurring methods, when combined with a subsequent feature detection method, perform in extracting selected visual features in the image. It is important to emphasize that our purpose is not to evaluate feature detectors but the deblurring algorithms although the proposed metric can also easily be used for the evaluation of the former.

An example ROC graph of three arbitrary detectors are shown in Fig. 3.12. When the detection performances of these three methods are compared using the FD-AROC metric, it can be concluded that the method represented with blue (*) markers has the highest overall performance since it has the highest area under the ROC curve. Second best is represented with green (+) markers and the red (o) markers has the worst performance. FD-AROC metric changes from 0 to 1. Therefore, a metric value 1 represents the perfect detector while 0 represents the worst case in which none of the true features could be detected.

For the comparison of ROC curves, we propose one more method besides area under ROC based FD-AROC metric. Instead of the area under the ROC curve, this measure scores how close a ROC curve is to the upper left corner of the graph (TPR=1 and FPR=0) which is the best score a method can get. It is measured using the Euclidean distance between the upper left corner ([0,1]) and the closest point of the ROC curve. We call this measure as MinDist. It is easier to calculate than the AROC measure. We show that both give close results in comparisons and therefore they can be used interchangeably for the comparison of the ROC curves.

3.4.2 Justification of Feature Based Metric

Some of the studies in literature try to compare the performances of motion deblurring methods in terms of some image quality metrics. Generally these metrics try to measure the visual quality of a processed image with respect to a pixel-wise error

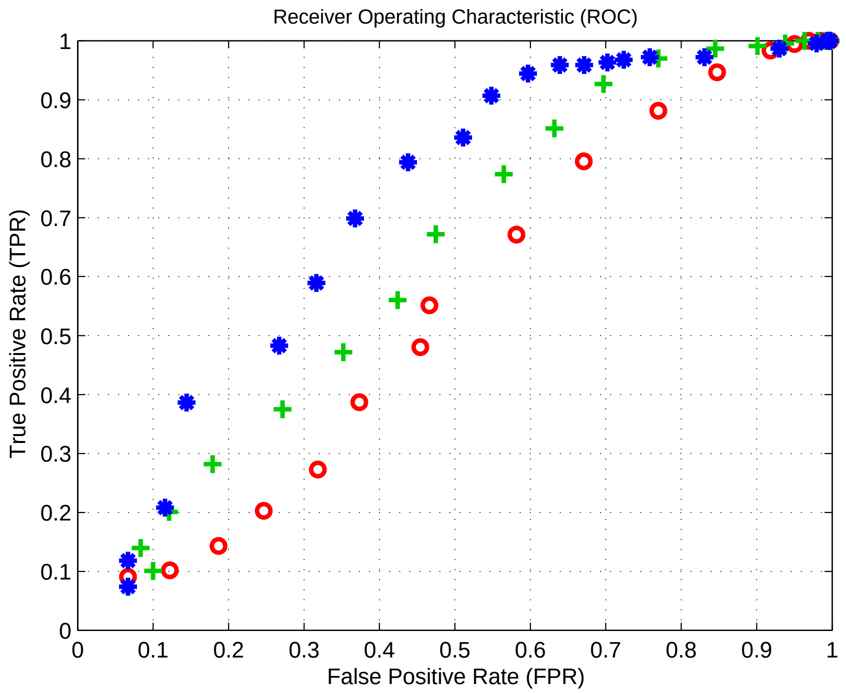


Figure 3.12: An example ROC graph of three arbitrary detectors. The one represented with blue star markers has the highest performance, second one is the green plusses and the red circles are the worst.

as compared to a reference image. However, for computer vision applications, the “quality of an image” should be judged based on the performance of computer vision algorithms, in particular the low level feature detectors. One can argue that existing image quality metrics for an image should also highly correlate with the feature detector performance on that image. Our claim, which we attempt to prove in this section through experimental results, is that the existing metrics may lead to misleading results for the computer vision application and hence it is more reliable to use a feature detection performance based metric instead.

In this section we compare three of the frequently used and well known image quality metrics from the literature, namely PSNR [19], NCC [27] and SSIM [54], together with the proposed metric on a set of examples. All of the metrics are computed on a controlled set of motion blurred images having different blur magnitudes. The blurred images also have the corresponding sharp images as the reference. The other details of the dataset is presented in Chapter 4.

The scores obtained using PSNR, NCC and SSIM metrics are demonstrated in Table 3.1 together with the proposed FD-AROC feature metric. The results for two different feature detectors(Harris & FAST) tested at two different motion blur magnitudes are given as an example. "Sp02" data includes 20 pixel and "Sp09" data includes 90 pixel 1D linear, uniform motion blurred images which are applied the given motion deblurring methods.

Table 3.1: Comparison of three frequently used image quality metrics and FD-AROC.

DATA	**Metric**	**Metric Scores (Ordered from best to worst)**						
Sp02	**PSNR**	Wiener	No-DB	Shan	R-L	Krish.	Goldst.	Whyte
		21.19	20.89	20.74	20.30	19.95	18.94	18.75
	NCC	Krish.	Shan	No-DB	Goldst.	R-L	Whyte	Wiener
		1.086	1.082	1.082	1.076	1.075	1.060	1.057
	SSIM	No-DB	Shan	Krish.	R-L	Whyte	Wiener	Goldst.
		0.858	0.837	0.813	0.799	0.776	0.764	0.689
	FD-AROC	Krish.	Shan	Whyte	No-DB	Goldst.	R-L	Wiener
	(Harris)	0.9972	0.9704	0.9462	0.9444	0.9370	0.8889	0.8655
	FD-AROC	Goldst.	Krish.	No-DB	Whyte	Shan	R-L	Wiener
	(FAST)	0.7877	0.6980	0.6036	0.5693	0.4836	0.3142	0.2658
Sp09	**PSNR**	R-L	Wiener	Whyte	No-DB	Shan	Goldst.	Krish.
		19.07	16.84	16.77	16.37	15.36	15.02	14.31
	NCC	Shan	No-DB	R-L	Goldst.	Whyte	Wiener	Krish.
		1.064	1.062	1.059	1.057	1.055	1.054	1.052
	SSIM	R-L	No-DB	Shan	Whyte	Wiener	Goldst.	Krish.
		0.784	0.777	0.745	0.734	0.600	0.577	0.514
	FD-AROC	Krish.	Goldst.	R-L	No-DB	Whyte	Shan	Wiener
	(Harris)	0.8526	0.6444	0.6419	0.6196	0.6038	0.5903	0.3808
	FD-AROC	Goldst.	Krish.	No-DB	Whyte	R-L	Shan	Wiener
	(FAST)	0.6318	0.6003	0.4599	0.3066	0.2445	0.2191	0.1585

The given table is a representative selection from our results in Chapter 5 and illustrates mainly one important result: None of the metrics can reliably predict which deblurring method will lead to an overall best feature detection performance when combined with a standard feature detector. A striking example is on the first row of the table: For small blur case (Sp02), while the Wiener deblurring method seems to maximize the performance in terms of PSNR, in fact it leads to the worst final feature detection performance! Another example is on the 3rd row of the table: SSIM measure ranks the Goldstein method as the worst performing deblurring method. In fact, it is the best overall method in terms of the feature detection performance, hence the FD-AROC measure.

CHAPTER 4

A COMPREHENSIVE MOTION BLUR DATASET FOR LEGGED ROBOTICS

4.1 Motivation and Scope of a New Motion Blur Dataset

The verification and performance evaluation of a motion deblurring method requires synthetic and/or real test data with reliable ground truth information. There are many studies in literature on the motion blur and deblurring. Some of the studies (blind methods) require only motion blurred image(s) whereas there are also methods(informed) that require both motion blurred image(s) and the corresponding motion or the resultant blurring function(PSF). Proposed methods are generally tested with a custom generated limited test set which emphasize the strong sides of that method. However, performance of the methods can be very dependent on the testing data. To be able to make a fair comparison of the methods, they should be tested on the same data set which covers enough scenarios of the application area. For example, if the application area of a deblurring method is the motion blur on legged platforms, then a test data set collected from a camera on a moving wheeled vehicle may not reflect its true performance on the application area. It is obvious that, the legged walking platforms have some particular motion characteristics and therefore that reflects to the properties of motion blur in captured frames.

Although there are many studies and data sets available on motion blur, there is no data set that can be used to evaluate the performances of the blind and non-blind motion deblurring algorithms on legged robotics application area. One of the aims of this thesis study is to build a comprehensive motion blur data set which will be used for the current and future studies on legged mobile robotic based computer vision tasks evaluations and comparisons. Although there are many different types of motion blur, a set of carefully selected scenarios can represent similar characteristics that can be applicable to a wide range of applications. Our database consists mainly of two scenarios: The first includes a very precisely controlled 1-D linear camera motion setup that is used to obtain both spatially invariant(uniform) and spatially varying(nonuniform) motion blurred images. The second is an unconstrained realistic motion blur data taken from a camera on a legged mobile platform while the robot is running at different speeds. The methods and specifications of our dataset are discussed in detail in the following sections.

4.2 1-D, Linear, Spatially Invariant Motion Blur Dataset

Before testing the methods on a fully realistic data set, it is generally advantageous to test on a more controlled data set where the pre-assumptions of the method are totally met. The first setup, as seen in Fig. 4.1, is established using the 1D motion platform developed by Quanser [39]. This platform is composed of a track and a cart moving on it. The cart is driven by a DC motor and the cart position is obtained through a high resolution optical quadrature encoder. The motors are driven by a motor driver. The encoder and the motor is controlled by a PC running Simulink via a data acquisition card. A Pointgrey Flea2 camera is placed on top of the cart using a mounting plate. There is a projector led light to provide proper light conditions for the camera. This setup can generate a speed and position controlled linear camera motion. By adjusting the speed, exposure and heading direction of the camera, it is possible to generate both spatially varying and invariant linear motion blur with precise ground truth information.

Coordinate axes definition for camera motion is shown in Fig. 2.6. Image plane is placed on a plane parallel to the x-y plane and with a distance f from the origin. Optical axis is along the z axis. Camera is mounted such that it translates along the sides, i.e. x axis. The roll, pitch and yaw rotations are defined around the axes z, x and y axes respectively. The translational motions are also defined with respect to this coordinate frame.

A glass plate with a checkerboard pattern is placed in front of the camera objective. It is oriented such that the plate is parallel to the image plane. The alignment of the plate to the image plane is done precisely by the help of a laser leveler device and a caliper. The camera is equipped with an 8mm lens having adjustable aperture and focus. The aperture is kept very narrow to have a large depth of field. This also helps the pinhole camera model fit better to the obtained images.

The motion of the setup is controlled by a ready to use speed controller designed in Simulink by Quanser. The controller is modified to capture images at predefined cart positions while controlling the cart speed to track a given speed profile. The cart speed can be adjusted accurately from as low as 0.1cm/sec and up to a maximum speed of 25cm/sec. To capture images at desired locations the controller generates an external camera trigger signal. Camera exposure is set before the beginning of the camera motion experiment and kept constant until the end of the experiment.

The accuracy of the setup is tested with an example speed profile which applies a

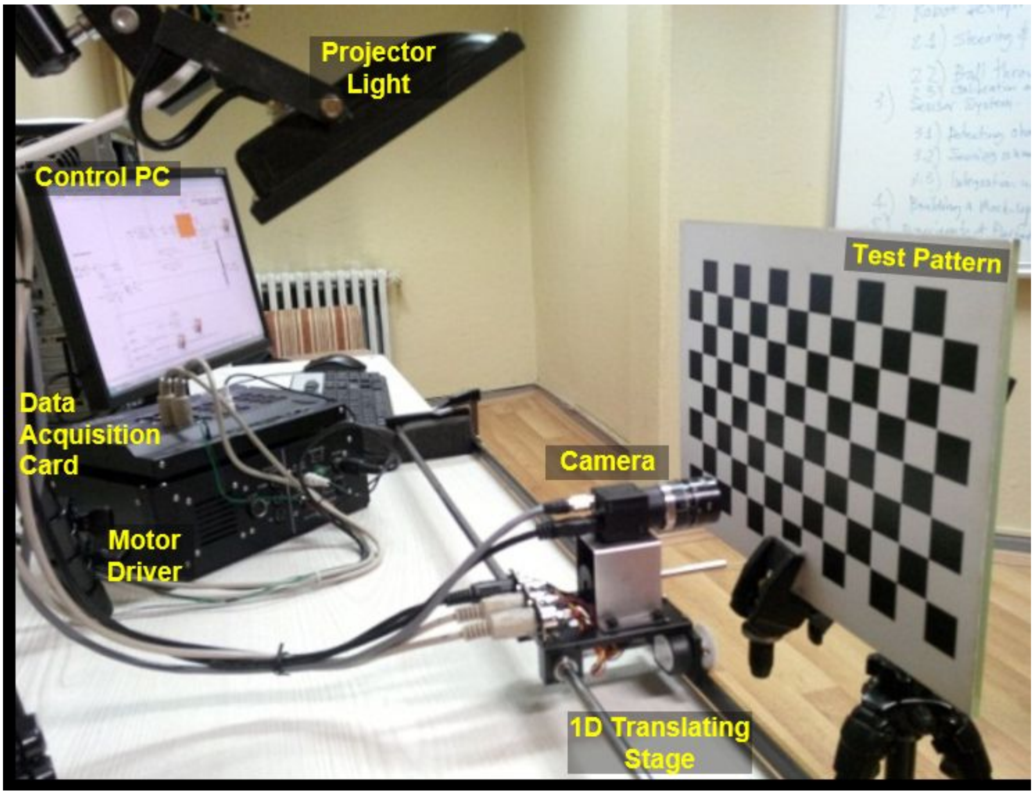


Figure 4.1: Accurate speed and position controlled one dimensional translating stage with a Pointgrey Flea2 camera mounted on top. This stage is used to demonstrate controllable camera speed resulting in controllable motion blur (i.e. PSF).

10cm/sec step input. Fig. 4.2 shows the step response of the system. The rise time is measured as 40ms. The mean speed for the 10cm/sec input is measured as 9.83cm/sec and there is a steady state error of 0.17cm/sec on the average with a 0.14 standard deviation.

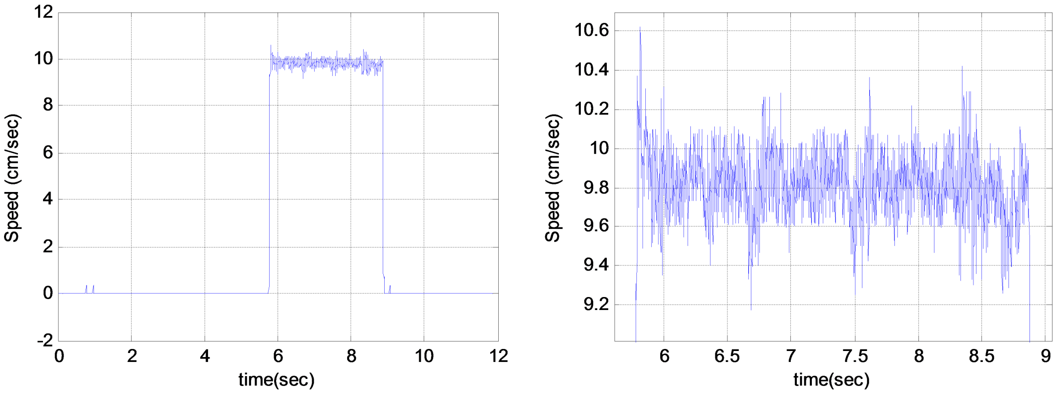
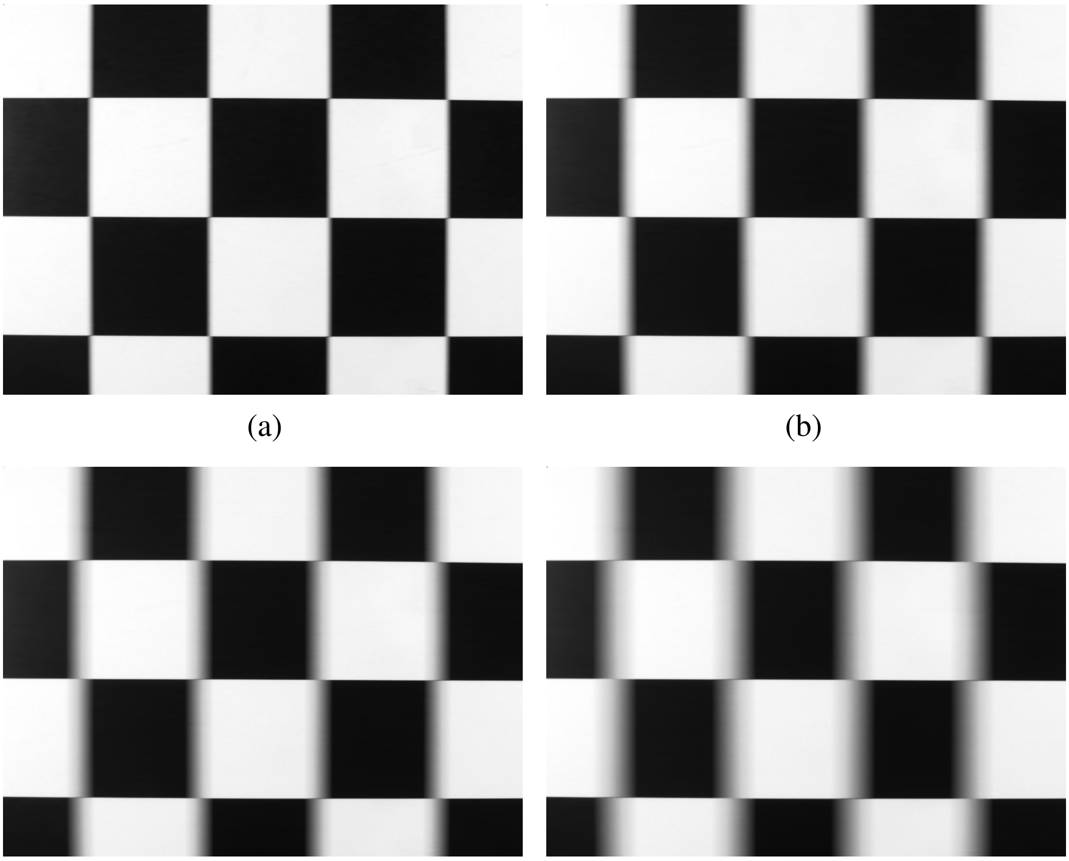


Figure 4.2: The velocity profile of the speed controlled setup when a 10cm/sec step input speed is applied. The graph on the right shows the magnification of ripples.

Fig. 4.3 shows the image frames captured while the camera is translating at different speeds that are 2cm/sec, 5cm/sec, 7cm/sec and 11cm/sec. It can be seen that each frame has different amount of motion blur. Although the camera is translating at different speeds, it is trigged to start exposure at the same track position to capture the same pattern. This helps us to overcome the alignment problem of images. Alignment differs only a few pixels based on the translation speed because of the trigger delay. It can easily be corrected by subtracting an offset distance that is proportional to the camera speed.



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

Figure 4.3: Images obtained by the controlled camera motion setup. The exposure is kept constant at 100ms for all images. The four images are all captured at the same cart position while the cart is moving at different speeds. The cart speeds are 2cm/sec, 5cm/sec, 7cm/sec and 11cm/sec for images numbered from (a) to (d) respectively.

The camera is moved at different constant speeds from 2cm/s up to 13cm/s. The platform motion starts exactly from the same location each time and the camera is externally triggered at 2cm intervals by the controller during its constant speed motion. There are a total of 120 motion blurred images with 1280x960 pixels resolution in the data set obtained at 8 different camera speeds and using 100ms exposure time.

There are also the corresponding sharp reference images captured at exactly the same positions as the blurred ones but without motion blur. The properties of the 1-D linear uniform motion blur dataset is summarized in Table 4.1.

Table 4.1: Properties of the camera speed controlled 1D linear, uniform motion blur dataset experiments.

Camera Speed	**Motion Blur Magnitude**	**Exposure Time**	**Number of Frames**	**Test Pattern**
2cm/sec	20 pixels	100 ms 3 ms	15 frames	Checkerboard Pebble Stone Prismatic Objects
3cm/sec	30 pixels			
5cm/sec	50 pixels			
6cm/sec	60 pixels			
7cm/sec	70 pixels			
9cm/sec	90 pixels			
11cm/sec	110 pixels			
13cm/sec	130 pixels			

The test pattern used in the setup can be changed easily for obtaining motion blur data from different scenes. Other than the checkerboard pattern, we took a high resolution photograph of a natural office scene, pebble stone and a setup of prismatic objects placed on top of a table with black background. The scenes of office, pebble stone and prismatic objects are shown in Fig. 4.4. The motion blur dataset experiments are repeated for the pebble stone and prismatic objects scene. We have photographed a number of other scenes however, did not used as a test pattern in the current dataset. These scenes are shown in Fig. 4.5. They are particularly selected from outdoor natural scenes, since legged robots can move in outdoors at different kinds of terrains and these are the scenes that are frequently exposed to the camera of the robot. It is possible to add more scenes to the dataset according to the necessities of future studies.

4.3 1-D Spatially Varying Motion Blur Dataset

The 1-D camera motion setup for obtaining spatially invariant linear motion blur is modified to obtain zooming blur which is a spatially varying motion blur type. The camera is rotated 90 degrees clockwise around its yaw axis and attached to the moving cart. The cart is moved from far to near the checkerboard pattern at different speeds to simulate zooming based motion blurring effect.

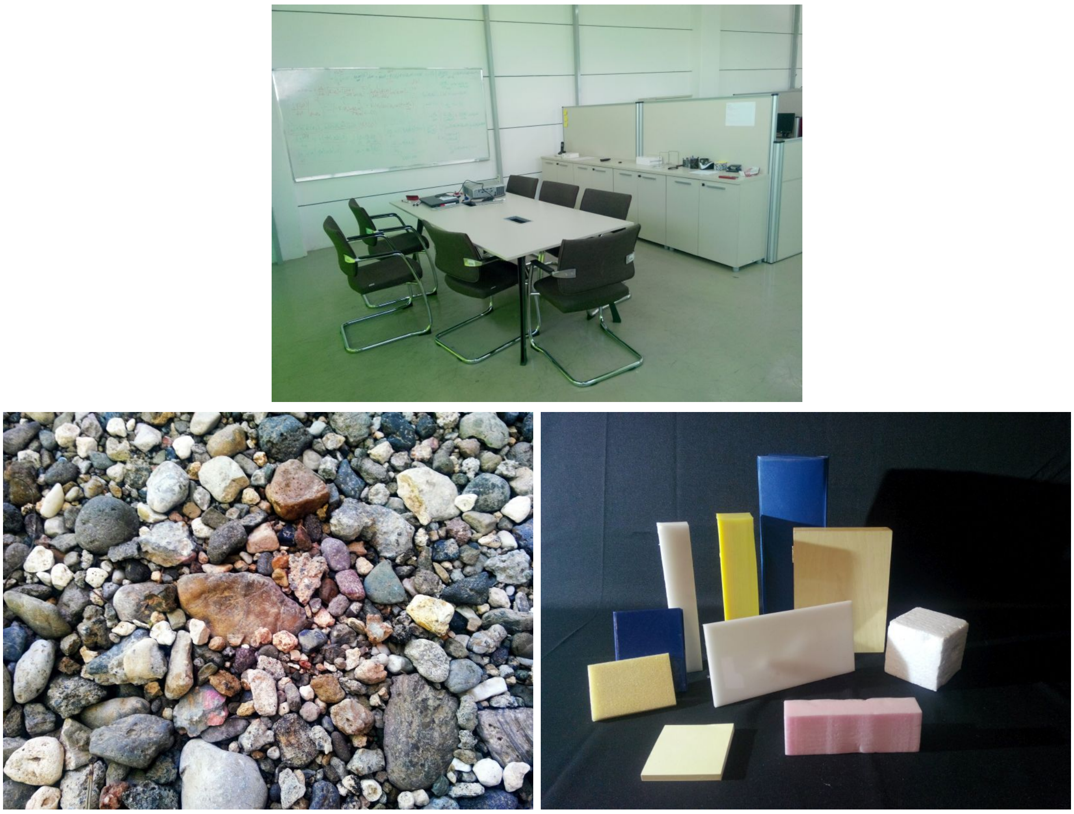


Figure 4.4: The scenes of office(top), pebble stone(bottom left) and prismatic objects(bottom right).

4.4 Multi-Sensor Motion Blur Dataset for Legged Mobile Robotic Vision Applications

The data set collection setup includes 4 sensors shown in Fig. 4.9: Camera, Fiberoptic Gyro (FOG), MEMS IMU and external motion tracking system. The camera is by Pointgrey Flea2 FL2-08s2c with 1024x768 pixels resolution streaming 15 fps. The communication interface is IEEE1394 Firewire and GPIO. The Fiberoptic Gyro is manufactured by Fizoptika, model VG941-3MD, streams 600 Hz Euler Angles (Roll,Pitch,Yaw) via RS232 serial communication at 115200 baud rate. IMU is from MicroStrain, model 3DM-GX1 which has 3 axis MEMS gyro and 3 axis MEMS accelerometer with compass. It has an internal filtering for accurate state estimation. However, we turned the filtering off because it is decreasing the highest achievable sampling rate. Therefore it can output raw data at 333 Hz. OptiTrack external motion tracking system (model s250e) can provide 6D state of a defined rigid body which has at least 3 reflective markers rigidly placed on it. The data is transmitted at up to 250 Hz via UDP protocol through Ethernet or wireless network.

The general overview of our experiment area is shown in Fig. 4.8. Six OptiTrack

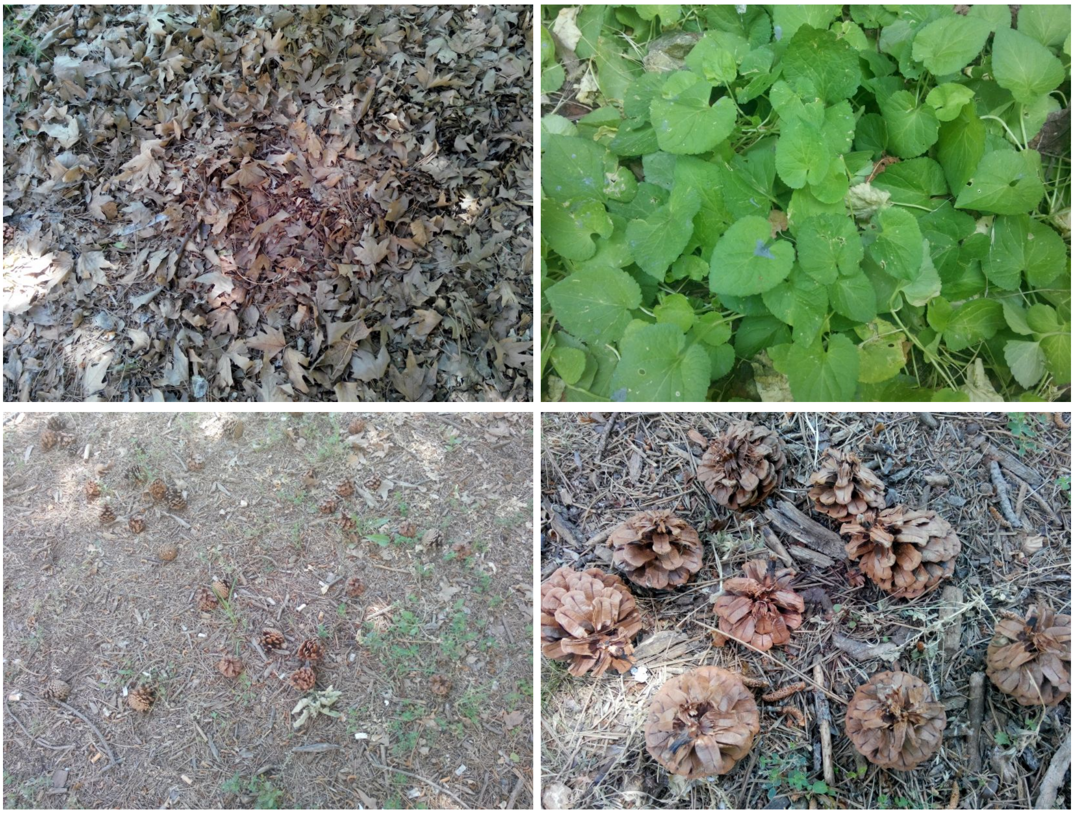


Figure 4.5: The scenes of "dry leaves"(upper left), "green leaves"(upper right), "Forest Ground"(lower left) and "pine cones"(lower right).

cameras are placed to the area (ceiling) to form a two triangular region such that at least 3 cameras can see the robot at any place during the experiments. The system can only calculate the 6D position and orientation if and only if at least two cameras can detect at least 3 of the reflective markers. Increasing the number of cameras that can see the robot at the same time increases the accuracy of the data. The camera system covers an area of about 5m in length and 3m in width. They are calibrated and the right handed coordinate axes are assigned such that the moving direction of the robot is –z axis, and the vertical direction from the floor plane is +y axis.

Table 4.2: The sensors used in dataset and their properties.

Sensor	**Model**	**Sampling Rate**	**Properties**
Camera	Pointgrey FL2-14S3C	15 fps	1280×960 RGB
FOG	Fizoptika VG941-3MD	600 Hz	±500 deg/sec
IMU	MicroStrain 3DM-GX1	333 Hz	±300 deg/sec, ±5 g
Motion Capture	OptiTrack S250e	120 Hz	832×832, 56°FOV

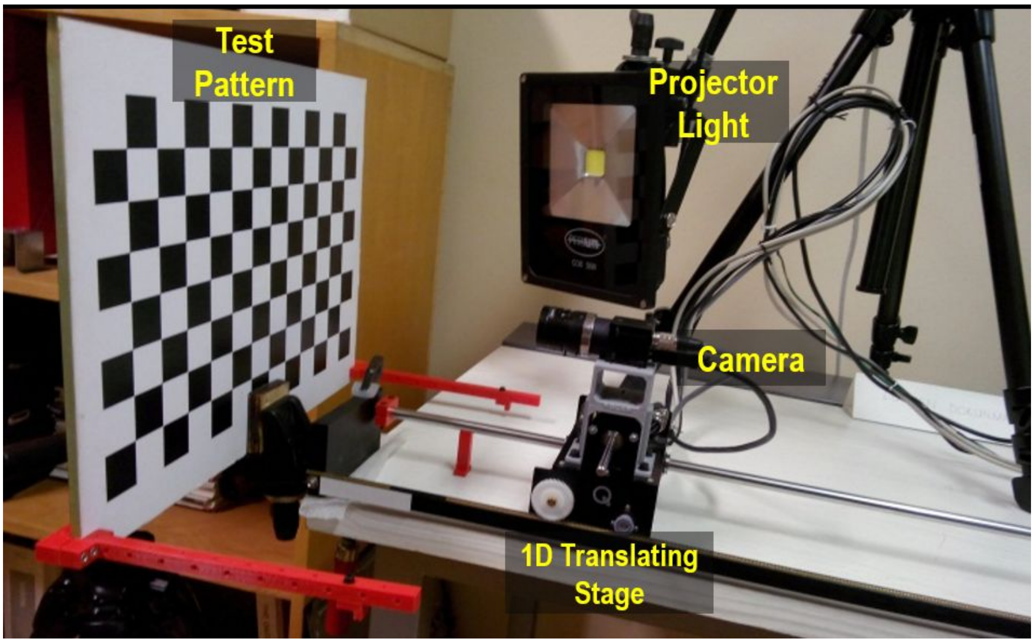


Figure 4.6: The setup for obtaining spatially varying motion blur caused by the zooming effect. The camera is rotated 90 degrees clockwise around its yaw axis and attached to the moving cart. The cart is moved from far to near the checkerboard pattern at different speeds to simulate zooming based motion blurring effect.

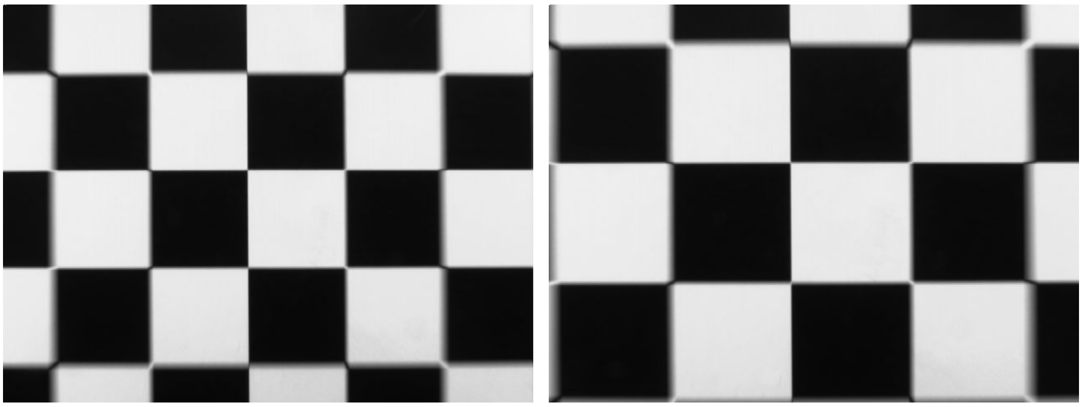


Figure 4.7: Spatially varying motion blurred images obtained by the above setup. The exposure is kept constant at 100ms for all images. The images are captured at same cart speed of 13cm/sec with 2cm distance intervals each during the approaching motion.

The FOG and IMU are used to measure the rotational velocities and linear acceleration of the camera. Therefore they should be rigidly connected to get accurate and reliable data. We designed a mounting part shown in Fig. 4.10 to mount FOG, IMU and the camera together. The designed part is manufactured in PLA plastic via a 3D printer. The manufactured mounting parts and assembled camera, FOG and

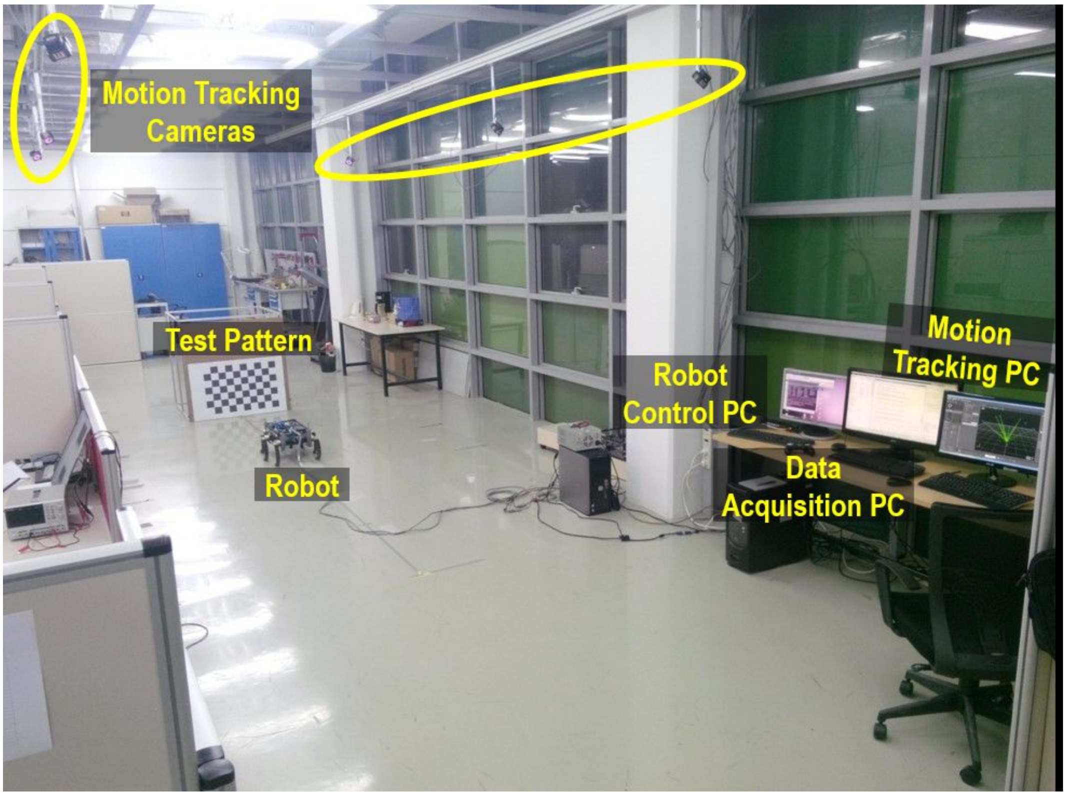


Figure 4.8: The general overview of the experiment area. There are 6 OptiTrack cameras mounted on the ceiling. The checkerboard pattern is placed vertical to the ground plane and the optical axis of the camera. Three PCs are used; one is for controlling of the robot; one of them is used for real-time data capturing from the sensors; and the other one is optitrack server which computes 6D position & orientation and sends packets via UDP protocol through network to the data capturing PC.

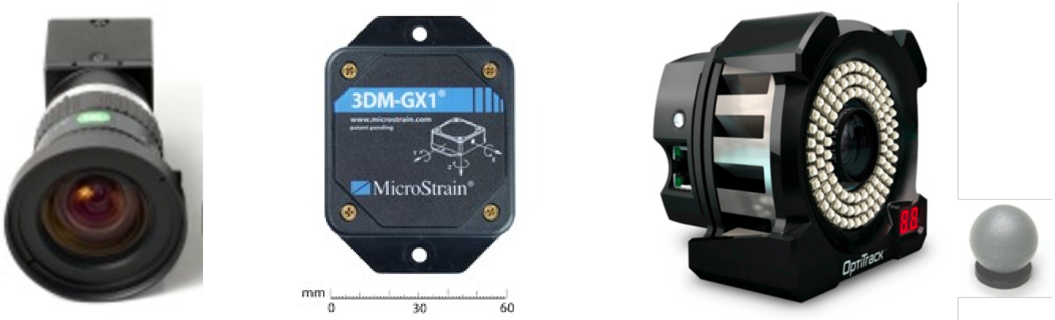


Figure 4.9: The PointGrey Flea2 camera (models 14s3c and 08s2c), MicroStrain IMU, OptiTrack s250e motion tracking camera and its reflective marker are shown respectively

IMU structure is shown in Fig. 4.11. The whole structure is designed such that it can also be assembled to the body frame of SensoRHex with a couple of auxiliary parts that are also designed and 3D printed. The FOG and IMU are assembled such that their coordinate axis directions are parallel to each other. The possible misalignments originated from the manufactured parts are neglected. The optical axis of camera corresponds to the -x axis of IMU & FOG and -z axis of OptiTrack. The vertical axis to the ground plane in the upper direction corresponds to -y axis of IMU & FOG and y axis of OptiTrack.

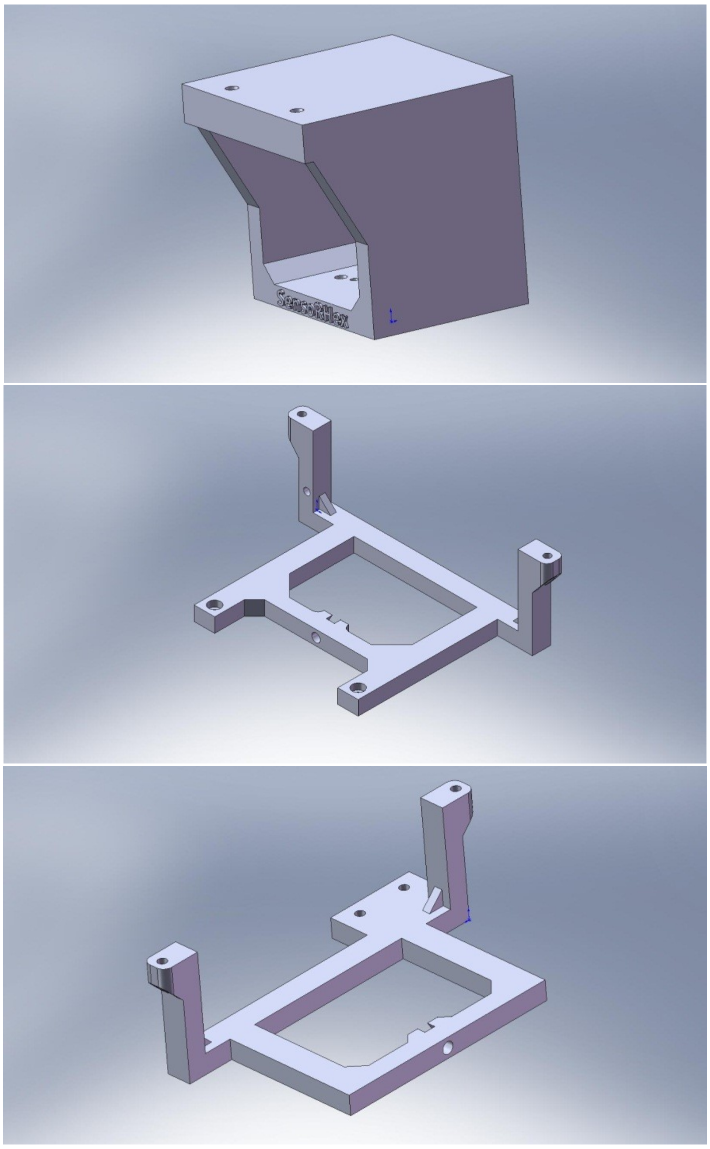


Figure 4.10: (Top) Camera mounting part for mounting both to the FOG, IMU and the body frame of SensoRHex. (Bottom) Mounting structure for assembling FOG and IMU together. The whole structure is also used for mounting to the camera.

The camera, FOG & IMU structure assembly onto the body frame of SensoRHex is shown in Fig. 4.12. The camera is accurately placed on the center of the front

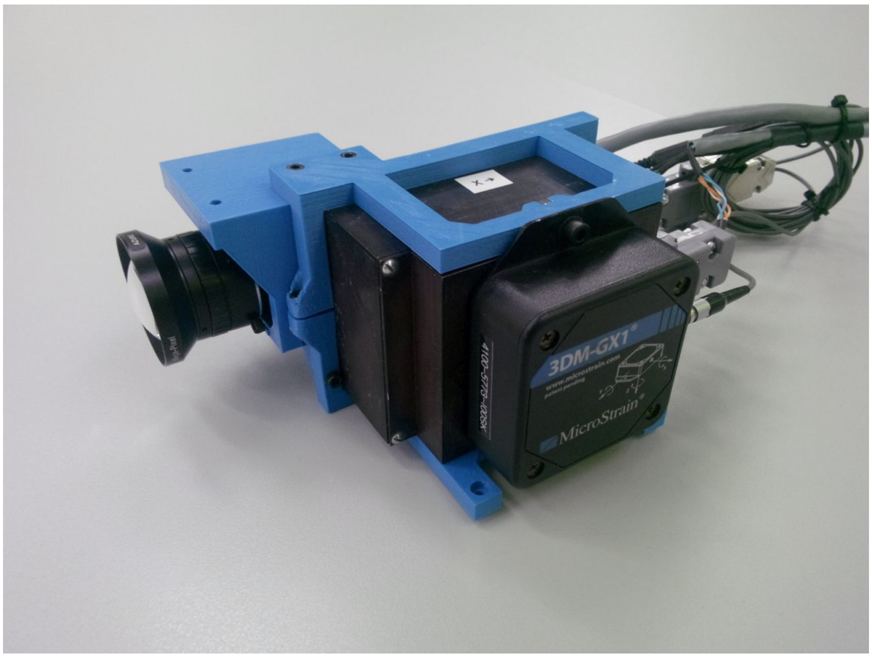


Figure 4.11: The manufactured mounting parts and assembly of the camera, FOG and IMU. The whole structure is designed such that it can also be assembled to the body frame of SensoRHex with a couple of auxiliary parts that are also designed and 3D printed.

frame. 11 reflective markers of external motion tracking system are rigidly mounted asymmetrically to the various places of the body frame via 3D printed mounting parts.

The robotic platforms have the ability of locomotion on a wide range of environment and terrain. They can both walk on flat indoor surfaces as well as on outdoor rough terrains. The camera images are frequently used for recognizing objects, localization and mapping kind of applications. Therefore, the data set might include data captured in a number of different environments and scenes. However, there are some limitations for collecting data outdoor. For instance, the external motion tracking cameras are not suitable for outdoor data capturing. Therefore, our data set includes only indoor captured data for the time being. However, it is possible to change the indoor scene. The scene can be a natural office scene as well as there can be a more structured pattern placed in front of the robot at a known distance for a more controlled scene capture. As frequently discussed in our reports, visual feature points are important information that can be derived from the images. Therefore, we place a checkerboard pattern in front of the camera to capture feature rich images whose ground truth is possible to extract.

It is frequently needed to manually analyze the data in the collected data set. However it is not easy to manually analyze the data without an easy to use interface, especially, when there is more than one kind of data such as visual images and numeric motion

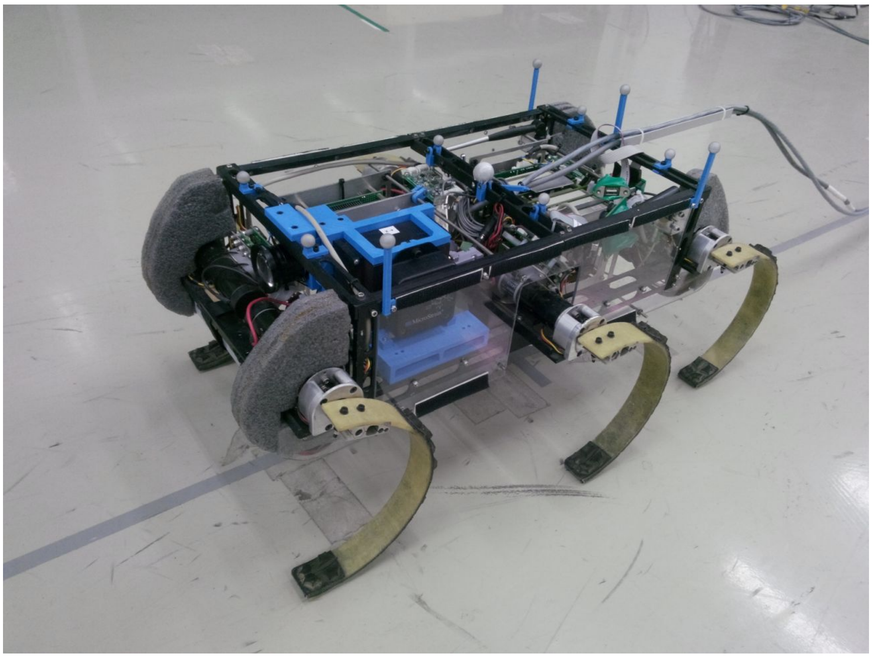


Figure 4.12: The camera, FOG & IMU structure assembled onto the body frame of SensoRHex. The camera is accurately placed on the center of the front frame. 11 reflective markers are rigidly mounted asymmetrically to the various places of the body frame via 3D printed mounting parts.

velocities that are captured from different sensors at different sampling rates. The best way to do this is to design a graphical user interface which has the capability to display the data that corresponds to a desired time instance and has the ability to easily navigate on the data.

The graphical user interface (GUI) designed for manual data analysis of multi-sensor motion blur data set is shown in Fig. 4.14. The complete rotational motion data are shown together with the image frame selected by clicking on the desired time instance. The motion data that correspond to the frame capturing period of selected frame is also plotted on the small figure windows seen on the left side. It is possible to move one frame forward or backwards using the mouse buttons. A preview of homogeneously selected frames from the whole sequence is displayed for a quick overview. The motion data and the image frames can be zoomed in and out using mouse and keyboard shortcuts. It is also possible to apply a corner feature detection algorithm to the selected frame. There is an option to enable or disable the corner feature detection algorithm since it takes time to find the features.

There have been a number of problems and difficulties in the real-time capturing of data from four different sensors at different sampling rates. All of the data capture process are controlled by programs written in Matlab running at windows operating system. However, there occurs some data drop at the OptiTrack data. OptiTrack cam-

Table 4.3: Properties of the legged robot motion blur dataset experiments.

Average Robot Speed	Experiment Duration
8 cm/sec	39 sec
12 cm/sec	38 sec
16 cm/sec	37 sec
22 cm/sec	33 sec
35 cm/sec	28 sec
48 cm/sec	17 sec

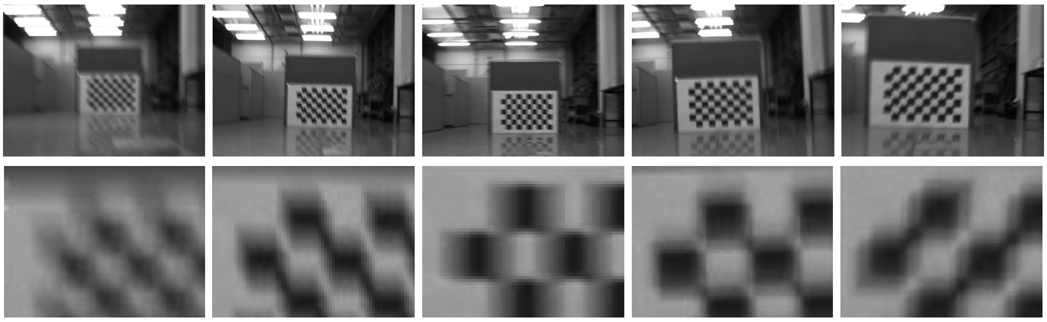


Figure 4.13: Sample image frames taken from the SensoRHex motion blur dataset together with their magnified views below them.

eras are connected to a server PC via Ethernet switches and the server broadcasts the computed motion data to other PCs connected to the same network via UDP protocol. Since UDP is not a lossless packet transmition protocol, some of the packets may be dropped because of the traffic congestions in network or the lack of processing power of the receiving computer to cope with the incoming packets in real-time. Since the capture computer is very busy with data capturing at high rates and video frames compression, the processing power is not sufficient to process the UDP packets in time. This causes frame drops in the received OptiTrack motion data. Although the OptiTrack is capable of sending data at up to 250 Hz, the data drop reaches percentages more than %60. Therefore, we set the sampling data rate to 120 Hz to decrease the frame drops below %10. However, it can be seen from the figures that some of the OptiTrack samples are contradicting with the gyro data which are caused by the frame drops.

4.5 Offline Synchronization of Motion Data & Image Data

All of the data from the four sensors are recorded at Matlab platform on the same computer to capture a synchronous data. However, the captured data are not perfectly synchronized due tp the operating system related issues. It is not possible to initial-

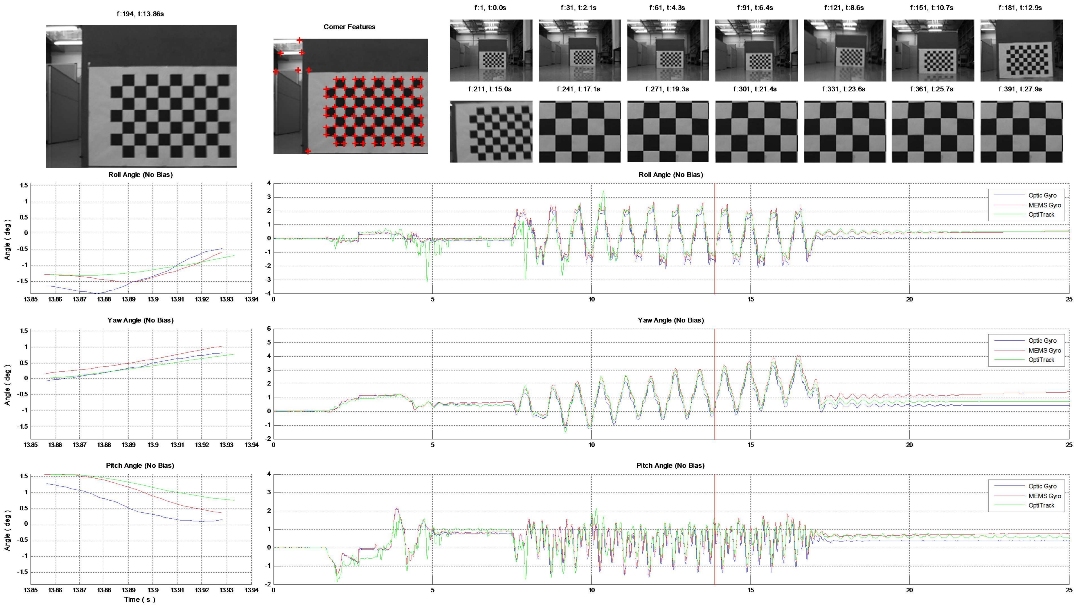


Figure 4.14: The graphical user interface (GUI) designed for manual data analysis of multi-sensor motion blur data set. The complete rotational motion data are shown together with the image frame selected by clicking on the desired time instance. The motion data that correspond to the frame capturing period of selected frame is also plotted on the small figure windows seen on the left side. A preview of homogenously selected frames from the whole sequence is displayed for a quick overview. It is also possible to zoom in and out to the motion data and the image frames.

ize the sensors to stream their data at the same desired time, since windows is not a real-time operating system. Therefore, a fine offline data synchronization is required. Since there are different kinds of data (visual and numerical) that are defined in different spaces, it is not easy to synchronize these data unless they are defined according to a common space. Therefore, we decided to utilize optical flow vectors space as the common space. The optical flow vectors are computed both from the motion data and the image sequence data. Then, optical flow vectors computed from the gyro data is matched with the optical flow computed from image frames for fine synchronization.

4.5.1 Optical Flow Computation From the Motion Data

What we call “optical flow vectors” in this part is somewhat different than the general computation and usage of optical flow. Optical flow is generally computed from the pixel movements between two images. However, we calculate and use optical flow vectors to represent the movements of pixels during the exposure time. This slight difference comes from the violation of the assumption which states that the move-

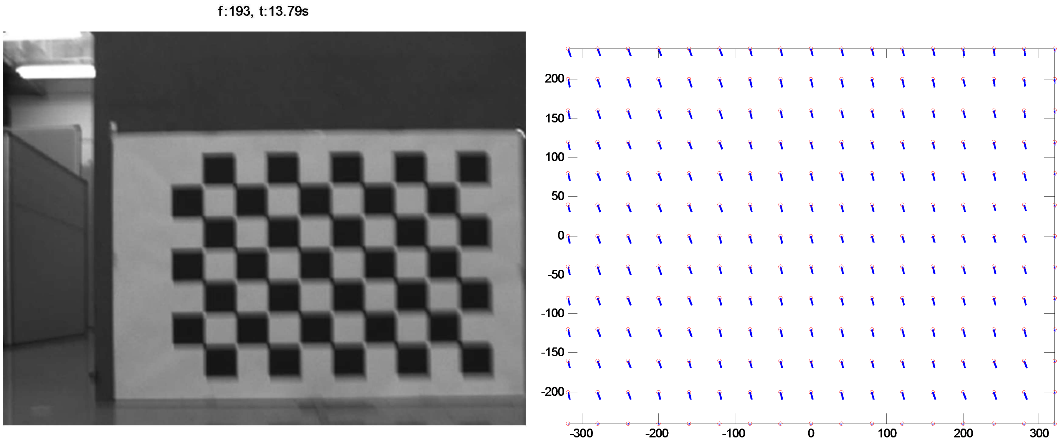


Figure 4.15: An example motion blurred frame (frame:193 at t=13.79s) taken from the collected dataset and the corresponding optical flow vectors computed from the gyro data.

ments of the pixels are negligibly small during the exposure. However, it is obvious that the movements of the pixels are relatively large within an exposure period in motion blurred images.

An example frame taken from the dataset and the corresponding optical flow vector field are shown in Fig. 4.15 The optical flow vectors are computed using the Euler angles that correspond to the frame capturing time which is shown in Fig. 10. Euler angles can be directly read from OptiTrack whereas the gyro rates taken from FOG and IMU needs to be integrated.

There have been a number of problems and difficulties in the real-time capturing of data from four different sensors at different sampling rates. All of the data capture process are controlled by programs written in Matlab running at windows operating system. However, there occurs some data drop at the OptiTrack data. OptiTrack cameras are connected to a server PC via Ethernet switches and the server broadcasts the computed motion data to other PCs connected to the same network via UDP protocol. Since UDP is not a lossless packet transmition protocol, some of the packets may be dropped because of the traffic congestions in network or the lack of processing power of the receiving computer to cope with the incoming packets in real-time. Since the capture computer is very busy with data capturing at high rates and video frames compression, the processing power is not sufficient to process the UDP packets in time. This causes frame drops in the received OptiTrack motion data. Although the OptiTrack is capable of sending data at up to 250 Hz, the data drop reaches percentages more than 60%. Therefore, we set the sampling data rate to 120 Hz to decrease the frame drops below 10%. However, it can be seen from the figures that some of the OptiTrack samples are contradicting with the gyro data which are caused by the frame drops.

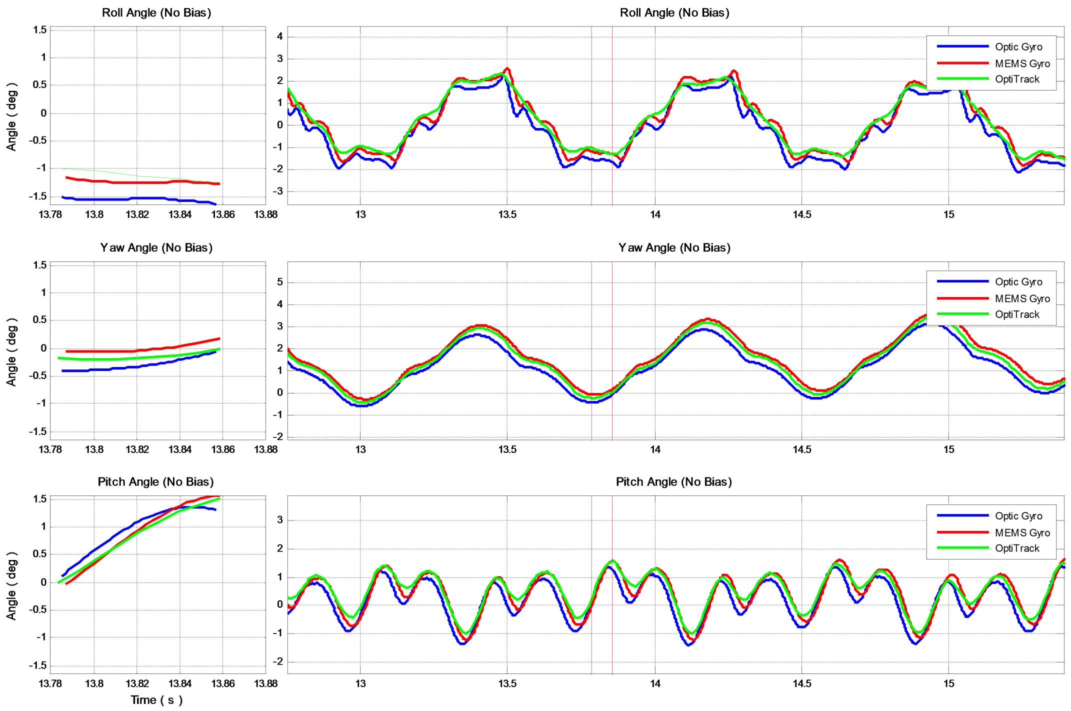


Figure 4.16: Euler angles from OptiTrack, FOG and IMU. Gyro based Euler angles are computed by the integration of gyro rates. The two red bars shows the frame capturing period (13.79s to 14.46s) of the example frame (f:193) shown in Fig. 9. Small figure on the left shows the gyro rates taken during the frame capturing period of the selected frame.

4.5.2 Optical Flow Computation From the Image Frames

Optical flow computation from the images requires two image frames. The main assumption of the optical flow algorithms is the brightness constancy which assumes that the brightness values of the pixels do not change for the consecutive frames. Actually in real images this assumption is impossible to strictly satisfy due to noise and motion blur. Since optical flow means that the pixels have a motion, then it is only possible to prevent motion blur by shortening the exposure time with respect to the frame period, so that the motion of pixels during the exposure are negligibly small when compared to the complete frame period. However, even if the motion blur may decrease the performance of the optical flow algorithm, it gives a comparable estimate on the motion.

For the computation of the optical flow, we used a well performing method in the literature proposed by Sun, Roth and Black [50] . For the computation we used two example consecutive blurred frames from the data set. One of the frames is more blurred than the other which enforces the limits of brightness constraint assumption. The selected frames and the computed optical flow vector field is shown in Fig. 4.18.

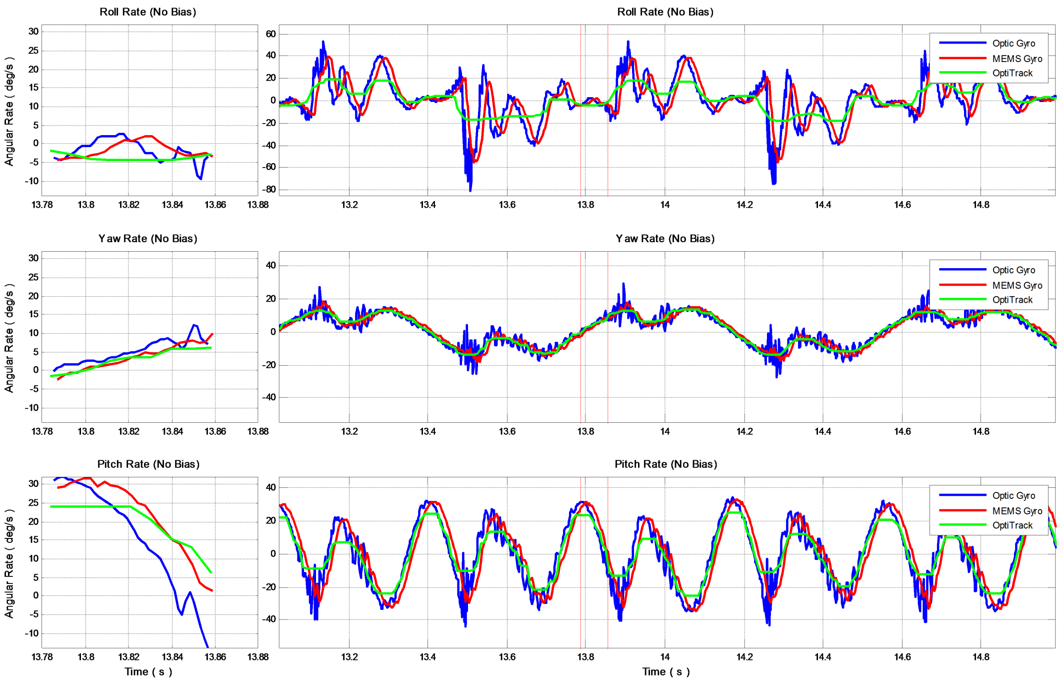


Figure 4.17: Euler rates captured from the FOG, IMU and OptiTrack. Small figure on the left shows the gyro rates taken during the frame capturing period of the example frame (f:193) shown in Fig. 9

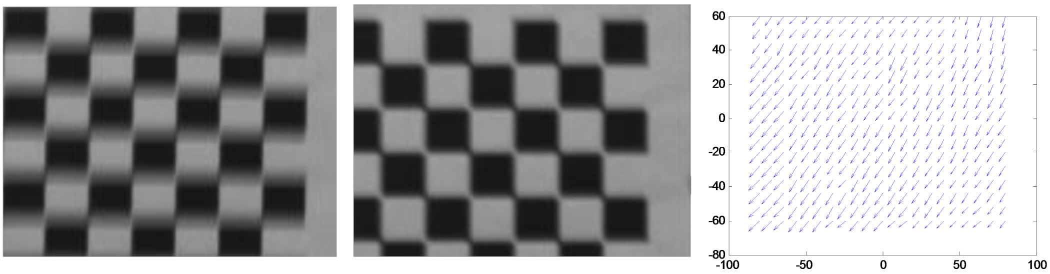


Figure 4.18: Selected consecutive motion blurred image frames for optical flow computation and the corresponding optical flow vector field computed by the method given in [32].

Although there is significant motion blur in one frame, the computed optical flow is good enough to estimate the motion of the frames.

4.5.3 Synchronization Using Optical Flow From Images & Camera Motion

Fig. 4.19 shows the frame averages of computed optical flow vectors. The red curve is computed from gyro data and the blue curve represents the optical flow frame averages computed from consecutive image frames. Although the waveforms have

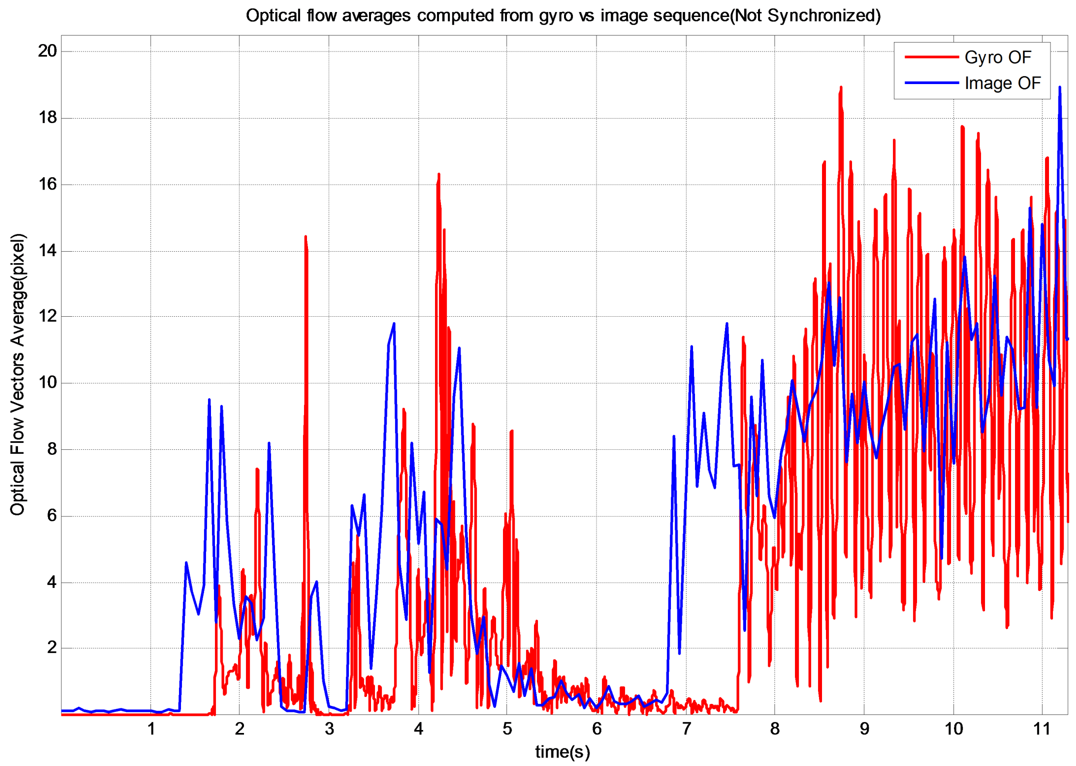


Figure 4.19: Optical flow averages computed from gyro versus image sequences. The starting of motion can easily be identified from both curves. It is observed that the gyro and camera data are not synchronized.

similar behaviors, it can easily be seen that there is a lack of synchronization. For example, the motion seems to have started earlier in image frames than the gyro samples. Therefore, we have utilized the starting and ending of optical motion for synchronization.

After the synchronization by the mentioned method, we have obtained the plots shown in Fig. 4.20. It can be seen that the general behavior of the waveforms matches well. The parameters obtained from this synchronization method is utilized to synchronize the motion and image data.

The synchronization parameters revealed that the timestamps and sampling rates of the actual data logged in computer differs slightly due to the non-real-time operating system related issues. Since the data is logged from different sensors via different communication interfaces, the occurrences of delays and sample losses are inevitable. Data fusion plays an important role here to be able to compensate the errors and imperfections in the collected data. More advanced data fusion and estimation methods, such as Kalman Filter, can be utilized if more accurate data is required.

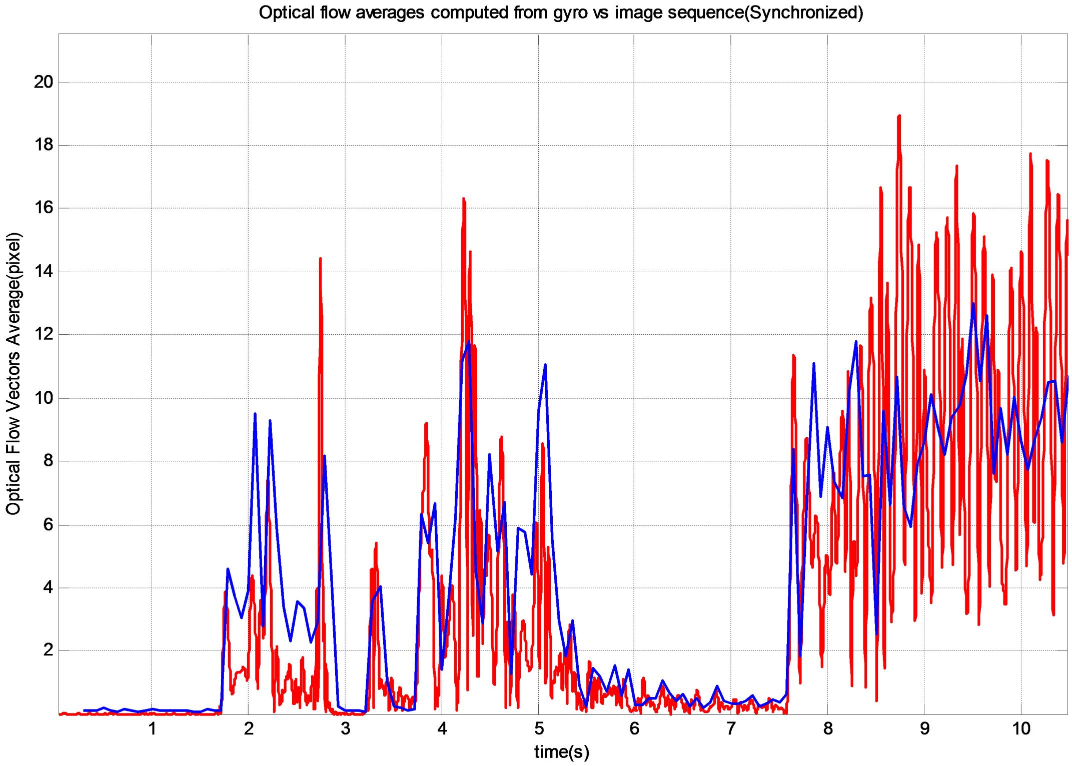


Figure 4.20: Optical flow averages computed from gyro versus image sequences. The gyro and camera data are synchronized by adding the required time shift and adjusting the actual frame rate of camera from 15 fps to 14 fps.

CHAPTER 5

EXPERIMENTS AND TEST RESULTS ON MOTION DEBLURRING STUDIES

In this chapter, we demonstrate experimentally the comparative performances of: 1)Single frame motion deblurring methods in terms of a) FD-AROC metric, b) a feature descriptor based metric and 2) the improvements possible a) using multi-frame approach and b) the extension to nonlinear blur kernel estimate.

The experiments substantiate two contributions of the thesis: a) A more reliable performance metric indicative of feature based performance of deblurring. b) The novel multi-frame deblurring method proposed based on variable non-overlapping motion blur spectral profiles. The chapter also covers a descriptor based characterization of these methods.

5.1 Testing and Benchmark Comparison of Single Frame Motion Deblurring Methods

In this section, the performances of selected single frame motion deblurring methods are compared based on the proposed feature detection metric. It is important to note that, the given performance comparisons of deblurring methods should not be evaluated alone but considered together with the applied feature detector. The ROC graphs and the FD-AROC metric indicate the relative performances of deblurring methods under the applied feature detector. Therefore, one can decide on which deblurring method to use together with the corresponding feature detector if the input images are exposed to motion blur.

The proposed metric is applied to a wide range of motion blur cases that are available in our motion blur dataset. The performances of methods vary according to the magnitude, uniformity and the linearity of the motion blur. Firstly, the methods are tested using the controlled 1D linear spatially invariant motion blur data with different camera speeds which yield different blur magnitudes. There are two different scenes used for experiments. One of them is the generic checkerboard pattern scene and the

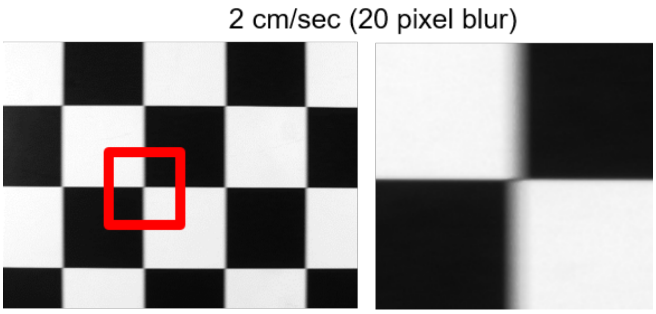


Figure 5.1: An example motion blurred frame from Sp02 dataset and its zoomed in view.

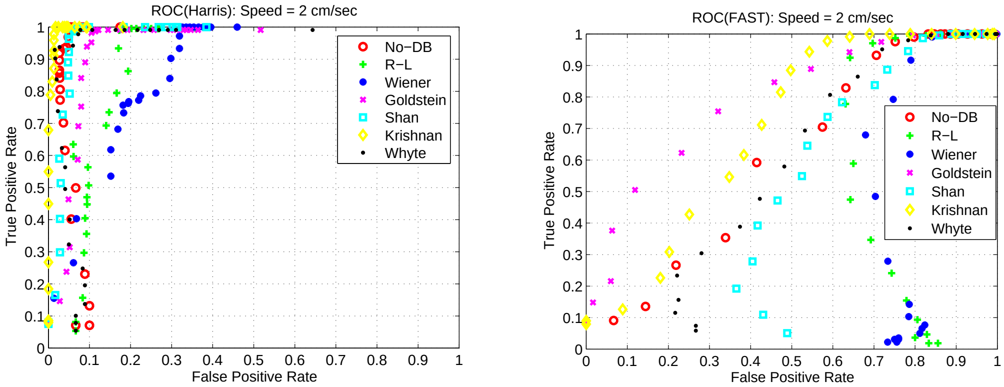


Figure 5.2: ROC analysis of Harris(left) and FAST(right) feature detectors under different motion deblurring methods using the linear uniform 20 pixel motion blurred images in Checkerboard dataset.

other one is the natural office scene. The ROC analysis of Harris and FAST feature detectors under different motion deblurring methods using the linear uniform 20 pixel motion blur checkerboard dataset is shown in Fig. 5.2. This corresponds to a camera speed of 2cm/sec which is the lowest motion blurred image data in the dataset. The top figure shows the results under Harris detector and the bottom one corresponds to the FAST detector. It can easily be observed that Harris has a good performance on low motion blurred images regardless of which deblurring method is used to reconstruct the image. The detection performance of FAST features is poor even at low motion blur, proving that motion blur has a significant impact on this important feature. However, the positive effect of motion deblurring methods is also significant in this case. One can observe from the results that Goldstein and Krishnan deblurring methods significantly increase the feature detection performance, although not matching the performance of Harris which appears to be much more robust against motion blur.

The ROC analysis for 20 pixels blurred checkerboard pattern dataset is repeated for 20 pixels blurred natural office scene. The ROC curves corresponding to the natural office scene dataset is shown in Fig. 5.3. The ROC curves in the left figure corre-

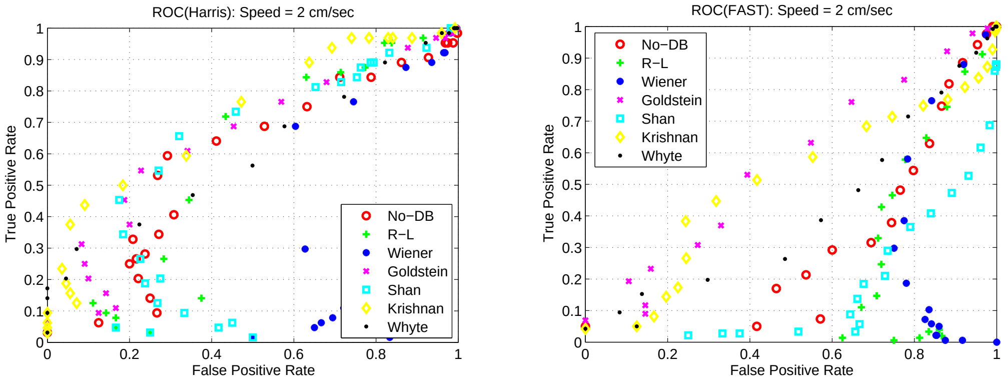


Figure 5.3: ROC analysis of Harris(left) and FAST(right) feature detectors under different motion deblurring methods using the linear uniform 20 pixel motion blurred images in Office Scene dataset.

sponds to Harris feature detector under different motion deblurring methods and no deblurring. Similarly, the right figure corresponds to results with FAST feature detector. As can be seen from the figures, the order of deblurring methods obtained with the natural office scene dataset are more or less the same as the order obtained from the checkerboard pattern dataset shown in Fig. 5.2. However, the feature detection performance in natural office scene is lower than the checkerboard pattern. This result is expected since the corner features in checkerboard pattern are significant where the corner features in natural office scene are not as strong as the checkerboard corner features. Some of the motion deblurring methods contribute to the performance of feature detectors as in checkerboard experiments where the others may degrade the performance of the detector even more than the no deblurring case. Harris corner detector performance is not effected much by motion deblurring methods either positively or negatively. Therefore, for low motion blur cases, there is no need to use a deblurring method with Harris. However, for FAST detector, Krishnan's and Goldstein's methods improve the detection performance dramatically.

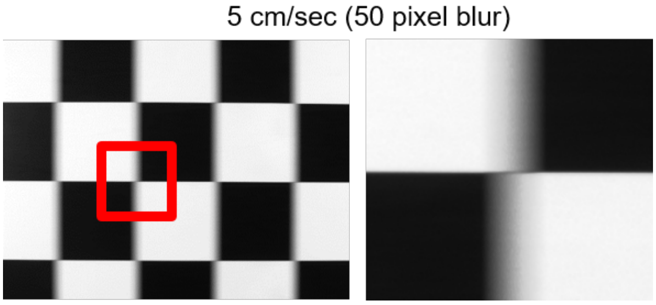


Figure 5.4: An example motion blurred frame from Sp05 dataset and its zoomed in view.

When the speed of the camera is increased, the ROC performance begin to get worse as expected since the amount of motion blur gets higher. Fig. 5.5 shows the ROC

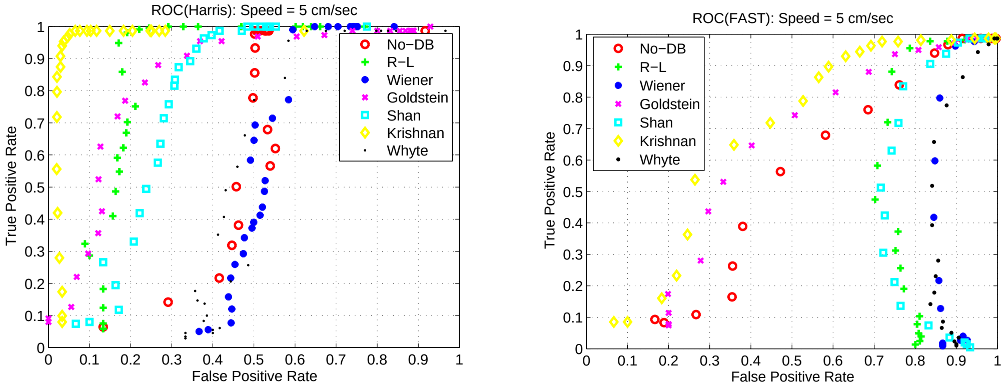


Figure 5.5: ROC analysis of Harris(left) and FAST(right) feature detectors under different motion deblurring methods using the linear uniform 50 pixel motion blurred images in Checkerboard dataset.

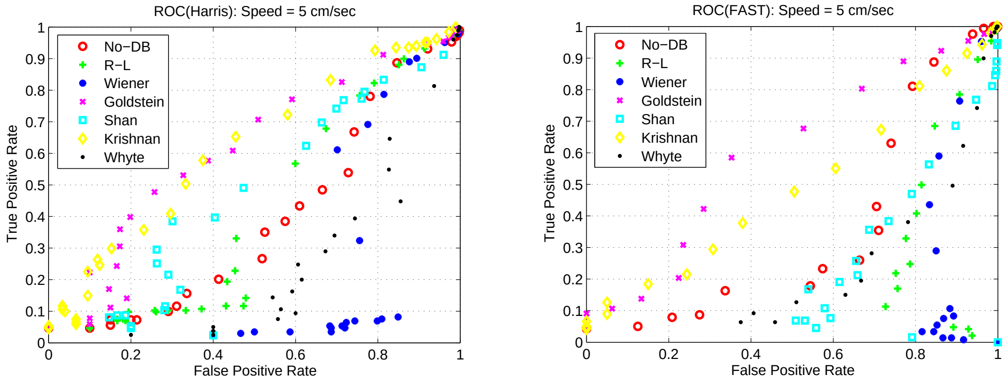


Figure 5.6: ROC analysis of Harris(left) and FAST(right) feature detectors under different motion deblurring methods using the linear uniform 50 pixel motion blurred images in Office Scene dataset.

graph of methods on 50 pixels motion blurred images. The decrease in detection performance is more considerable for Harris detector than the FAST. However, the utilization of Krishnan deblurring method increases the detection performance of Harris dramatically. Goldstein, R-L and Shan methods also have a positive effect in the detection performance of Harris. For FAST, the detection performance does not drop dramatically when the motion blur is increased. However, it is still possible to increase the detection performance when Krishnan and Goldstein deblurring methods are applied to the motion blurred images.

Fig. 5.6 shows the ROC curves obtained from natural office scene dataset with 50 pixels motion blur. As the motion blur increases, the effect of motion deblurring methods become more significant both for Harris and FAST. Krishnan, Goldstein and Shan improves the performance of Harris where only Krishnan and Goldstein contributes to the performance of FAST.

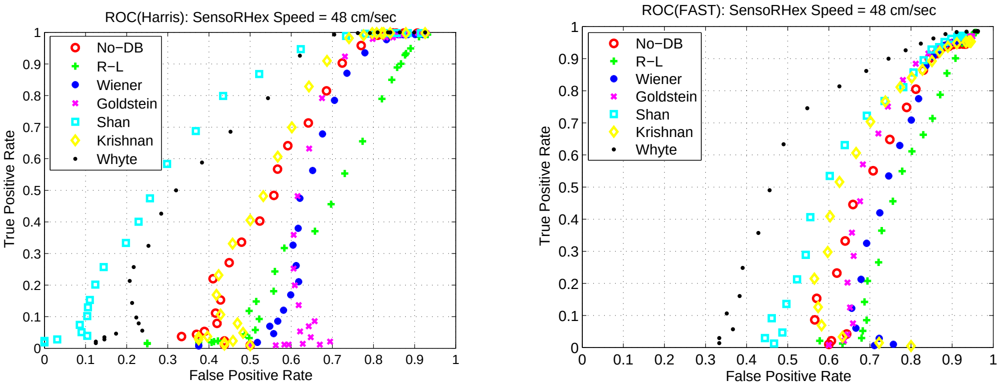


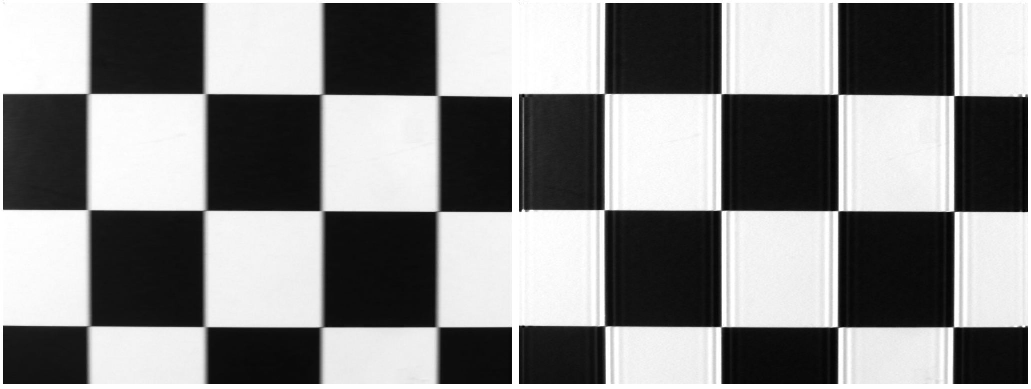
Figure 5.7: ROC analysis of Harris(left) and FAST(right) feature detectors under different motion deblurring methods using the SensoRHex motion blur dataset.

Up to now, the presented results are all obtained using a fully controlled 1D linear and uniform motion blur data set which guarantees that the assumptions of all the utilized deblurring methods are met. For some applications these results can be applicable only if the blur consists of translation along a single axis. The dataset also includes scenarios in development where these assumptions, such as blur uniformity or time invariance are selectively removed. Within the scope of the present paper, our focus is on demonstrating the two extreme of the scenario spectrum: In the other extreme, such as the legged robotics application, the motion of the camera is not controlled and therefore results in complicated motion blur PSFs. Most of the time, this motion is a combination of roll, pitch and yaw rotations together with translations along x,y and z axes. Often, for far field imaging, the translational motion effects can be negligible as compared with the rotational motion. In this case, there appears nonlinear and nonuniform motion blurred frames which cause some of the assumptions of motion deblurring methods to be violated. To be able to measure the applicability of motion deblurring methods in these scenarios, we tested the above methods on our motion blur dataset obtained from the research robot platform SensoRHex. The results are obtained using the images captured while SensoRHex is walking at highest locomotion speed: 48cm/sec. Although the robot has the ability to walk faster, the motion blur dataset is collected for speeds up to 48cm/sec which already results in significant magnitude of motion blur in the captured image frames. Some of the frames are given as an example from the dataset together with their magnified views below them in Fig. 4.13.

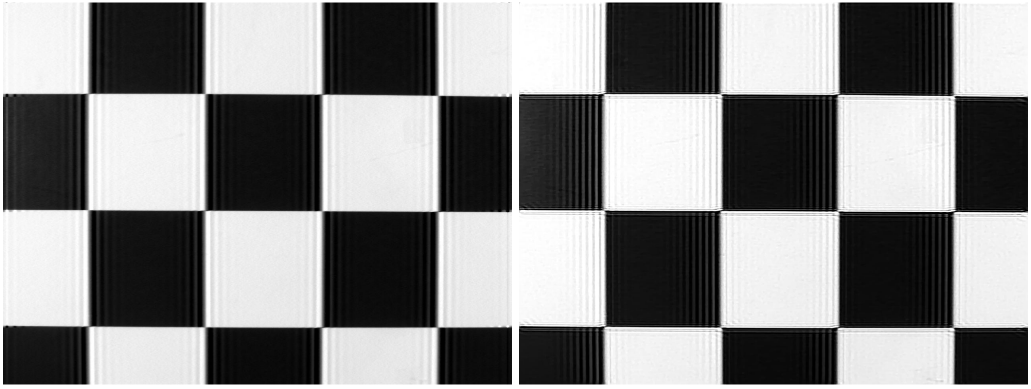
The results obtained using the SensoRHex motion blur dataset is shown in Fig. 5.7. Similar to the results obtained using the controlled motion blur dataset, the feature detection performance of Harris is higher than the FAST with or without the application of any deblurring method. For both of the detectors, Shan, Whyte and Krishnan's deblurring methods improve the detection performance. However, for Harris corner detector, Shan yields the highest performance while Whyte performs the best for the FAST detector.

It may seem surprising that Whyte have a higher performance in SensoRHex dataset than the controlled motion blur dataset. Actually, the success of Whyte over other methods in SensoRHex dataset is not by chance since it is the only method that handles nonuniform motion blur caused by the 3D camera rotations as explained in Section 1.1. The success of Shan for Harris detector is similar to its behavior in previous dataset examples. Although it does not handle nonuniform motion blur, the ringing artifact suppression property of Shan seems to work well with Harris corner detector even in nonuniform motion blur cases.

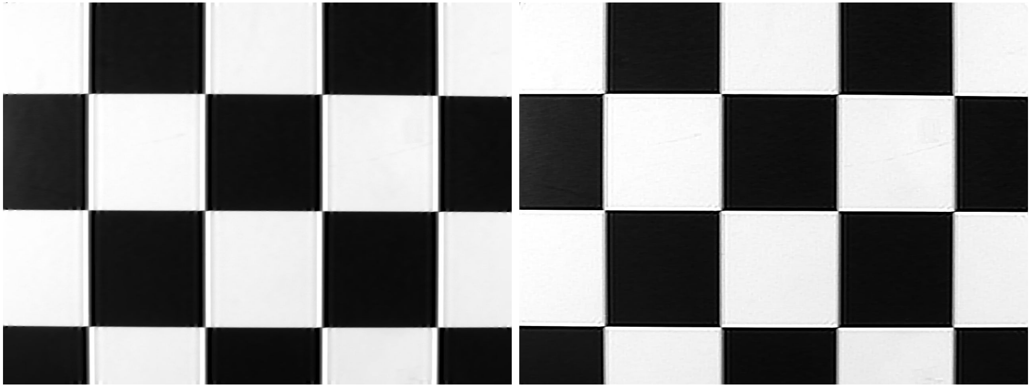
The performances of the methods illustrated by the given ROC curves can be quantified as a scalar number using the proposed FD-AROC metric. The ROC curves for the three chosen scenarios from the larger dataset are illustrated above. The FD-AROC scores for the deblurring methods for the entire content of the dataset are summarized in Table 5.1 and Table 5.2 for the Harris corner feature detector and FAST feature detector respectively.



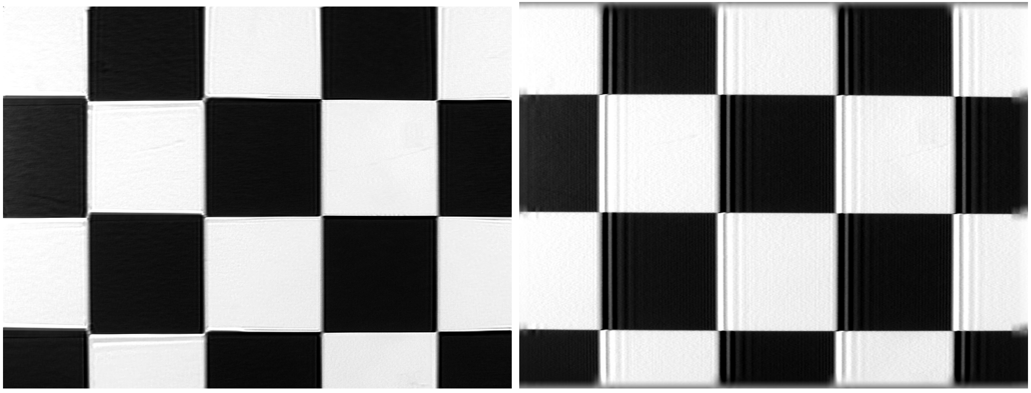
(a) No-DB (b) R-L



(c) Wiener (d) Goldstein

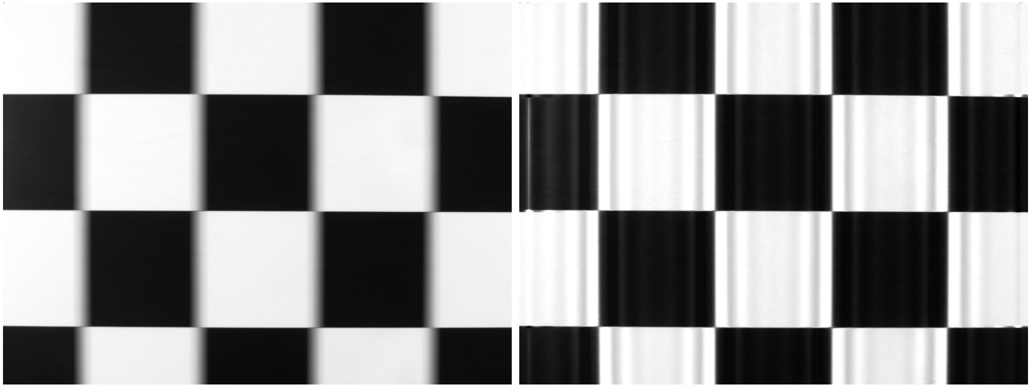


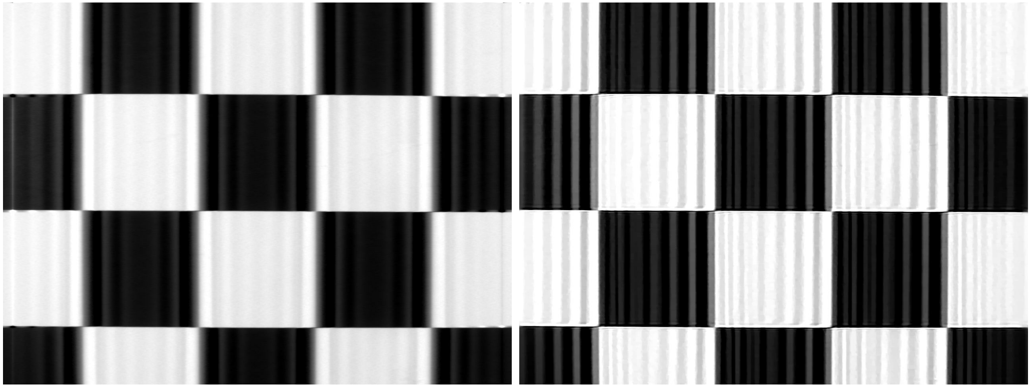
(e) Shan (f) Krishnan

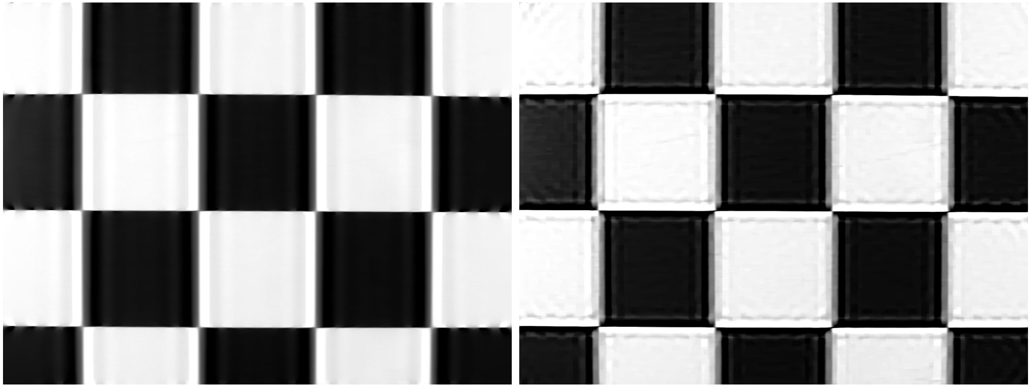


(g) Whyte (h) MID

Figure 5.8: (a) 20 pixels motion blurred checkerboard image. (b-h) Deblurring results obtained using the methods written on top of the frames given above.


(a) No-DB (b) R-L


(c) Wiener (d) Goldstein


(e) Shan (f) Krishnan

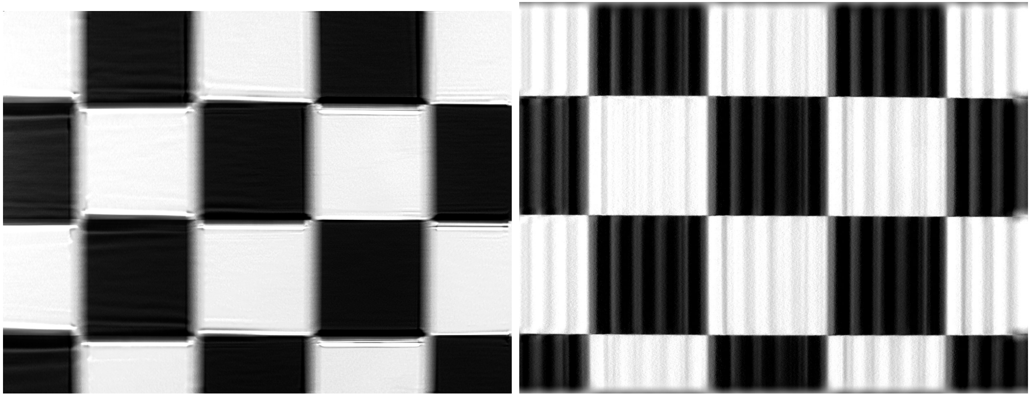
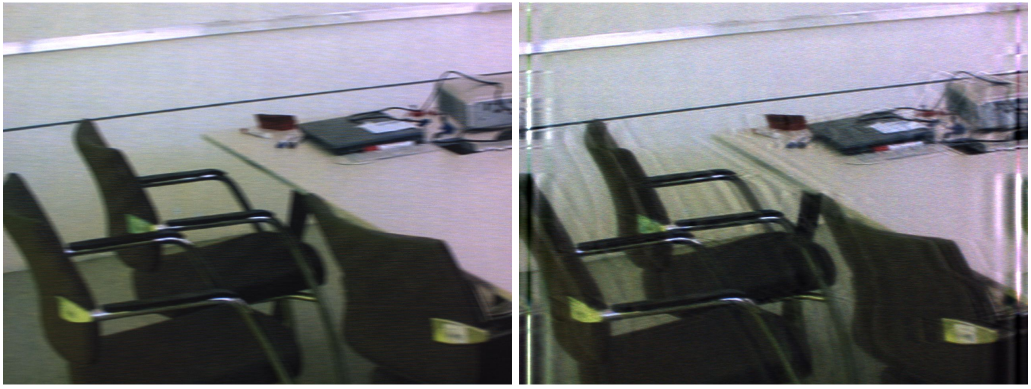
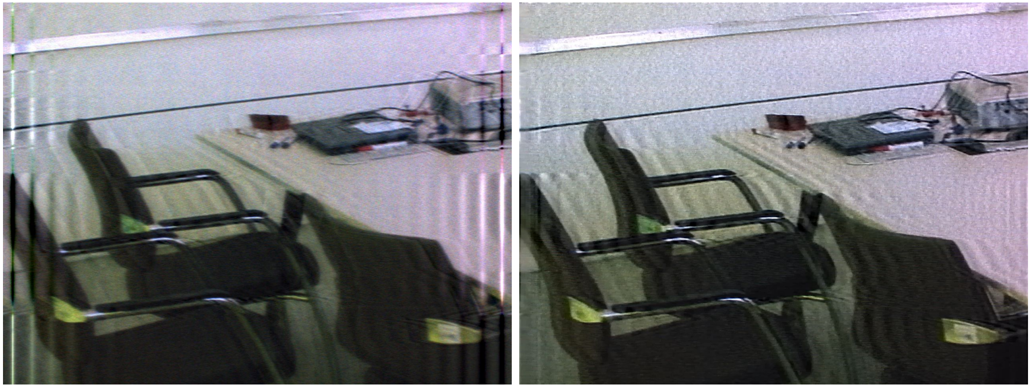

(g) Whyte (h) MID

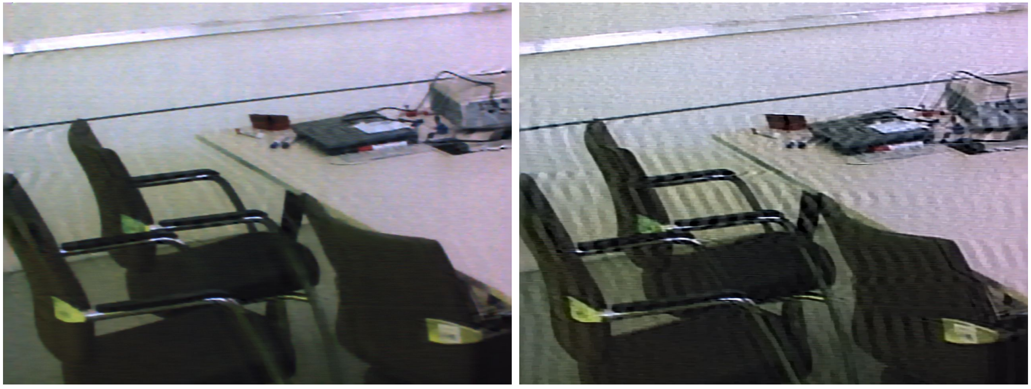
Figure 5.9: (a) 50 pixels motion blurred checkerboard image. (b-h) Deblurring results obtained using the methods written on top of the frames given above.



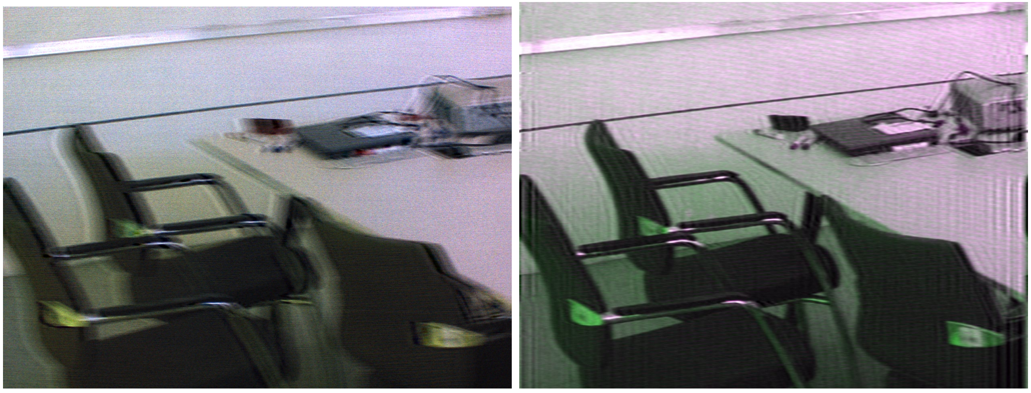
(a) No-DB (b) R-L



(c) Wiener (d) Goldstein

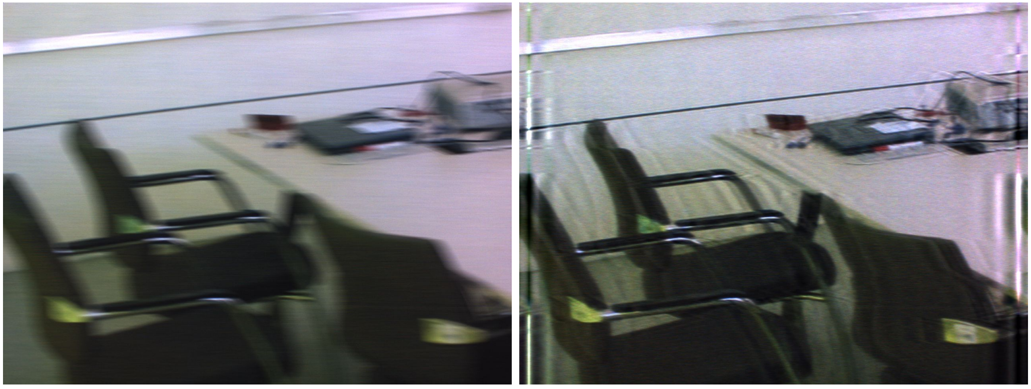


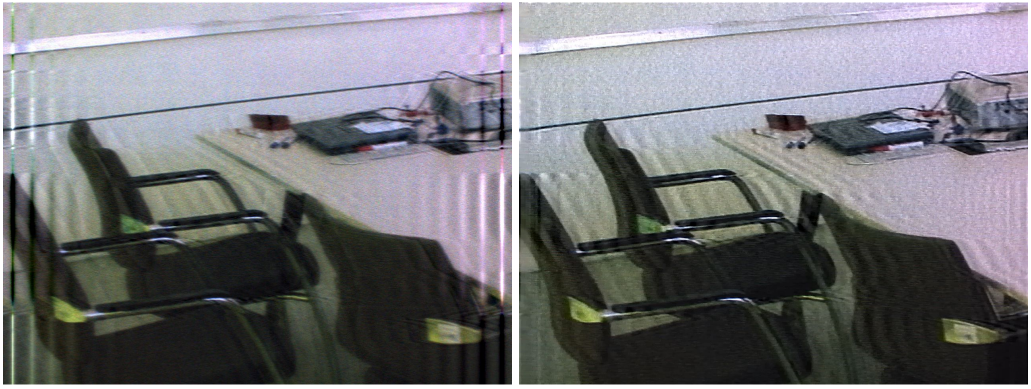
(e) Shan (f) Krishnan

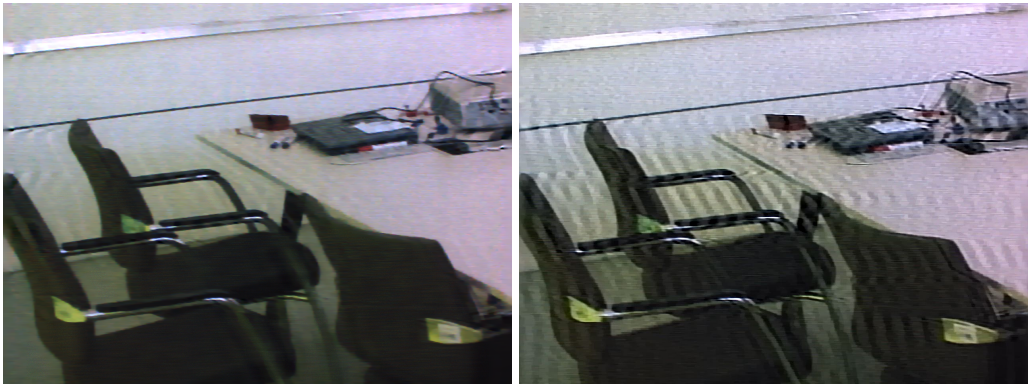


(g) Whyte (h) MID

Figure 5.10: (a) 20 pixels motion blurred Office Scene image. (b-h) Deblurring results obtained using the methods written on top of the frames given above.


(a) No-DB (b) R-L


(c) Wiener (d) Goldstein


(e) Shan (f) Krishnan

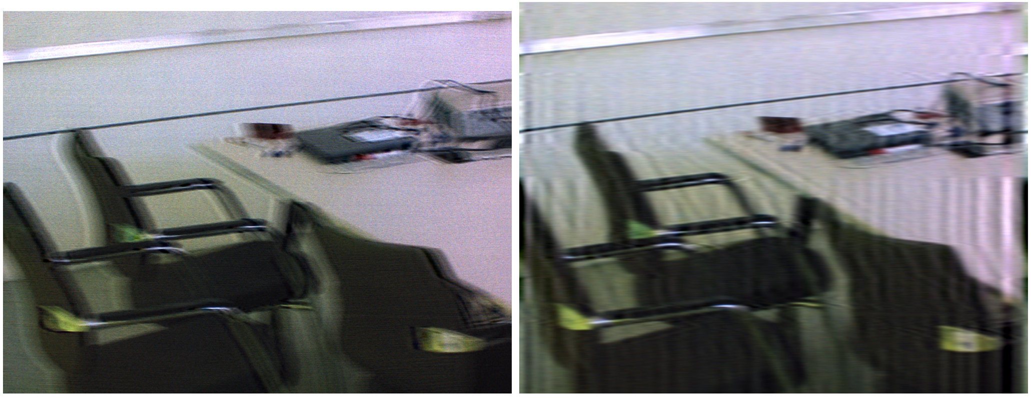

(g) Whyte (h) MID

Figure 5.11: (a) 50 pixels motion blurred Office Scene image. (b-h) Deblurring results obtained using the methods written on top of the frames given above.

Table 5.1: Metric scores of motion deblurring methods with Harris corner detector.

DATA	Metric	Metric Scores (Ordered from best to worst)						
Sp02	MinDist	Krishnan	No-DB	Shan	Whyte	Goldstein	R-L	Wiener
		0.0003	0.0023	0.0028	0.0046	0.0113	0.0298	0.0922
	ROC Area	Krishnan	Shan	Whyte	No-DB	Goldstein	R-L	Wiener
		0.9972	0.9704	0.9462	0.9444	0.9370	0.8889	0.8655
Sp03	MinDist	Krishnan	R-L	Shan	Goldstein	Whyte	Wiener	No-DB
		0.0018	0.0169	0.0342	0.0684	0.0698	0.1931	0.2412
	ROC Area	Krishnan	R-L	Shan	Whyte	Goldstein	Wiener	No-DB
		0.9639	0.9046	0.8602	0.8375	0.8339	0.6297	0.5911
Sp05	MinDist	Krishnan	R-L	Goldstein	Shan	No-DB	Whyte	Wiener
		0.0032	0.0318	0.0856	0.1165	0.2526	0.3032	0.3470
	ROC Area	Krishnan	R-L	Goldstein	Shan	No-DB	Whyte	Wiener
		0.9645	0.8460	0.8424	0.7674	0.5628	0.5343	0.5034
Sp06	MinDist	Krishnan	R-L	Goldstein	Shan	No-DB	Wiener	Whyte
		0.0012	0.0246	0.1011	0.1023	0.2464	0.2895	0.2923
	ROC Area	Krishnan	R-L	Goldstein	Shan	No-DB	Whyte	Wiener
		0.9861	0.8135	0.8088	0.7862	0.6032	0.5515	0.5206
Sp07	MinDist	Krishnan	R-L	Goldstein	Shan	No-DB	Wiener	Whyte
		0.0039	0.0202	0.1643	0.2512	0.2725	0.2762	0.3377
	ROC Area	Krishnan	R-L	Goldstein	Shan	No-DB	Whyte	Wiener
		0.9578	0.8317	0.7305	0.6811	0.5592	0.5422	0.5235
Sp09	MinDist	Krishnan	R-L	Whyte	Goldstein	Shan	No-DB	Wiener
		0.0799	0.0852	0.2215	0.2276	0.2638	0.2651	0.4522
	ROC Area	Krishnan	Goldstein	R-L	No-DB	Whyte	Shan	Wiener
		0.8526	0.6444	0.6419	0.6196	0.6038	0.5903	0.3808
S.Rhex	MinDist	Shan	Whyte	Krishnan	No-DB	Goldstein	Wiener	R-L
		0.2293	0.3036	0.4428	0.4789	0.4989	0.5437	0.7174
	ROC Area	Shan	Whyte	Krishnan	No-DB	Goldstein	Wiener	R-L
		0.7041	0.6311	0.4593	0.4425	0.3526	0.3494	0.3106

Table 5.2: Metric scores of motion deblurring methods with FAST corner detector.

DATA	**Metric**	**Metric Scores (Ordered from best to worst)**						
Sp02	MinDist	Goldstein 0.2298	Krishnan 0.2839	R-L 0.4322	No-DB 0.4578	Shan 0.5788	Wiener 0.6437	Whyte 0.7118
	ROC Area	Goldstein 0.7877	Krishnan 0.6980	No-DB 0.6036	Shan 0.5693	R-L 0.4836	Whyte 0.3142	Wiener 0.2658
Sp03	MinDist	Goldstein 0.2642	Krishnan 0.2690	No-DB 0.4251	R-L 0.4286	Shan 0.5746	Whyte 0.7166	Wiener 0.7180
	ROC Area	Krishnan 0.6904	Goldstein 0.6858	No-DB 0.5678	R-L 0.3446	Shan 0.2516	Whyte 0.2027	Wiener 0.1805
Sp05	MinDist	Krishnan 0.2527	Goldstein 0.2868	No-DB 0.4135	R-L 0.5660	Shan 0.6210	Wiener 0.7800	Whyte 0.8190
	ROC Area	Krishnan 0.6662	Goldstein 0.6074	No-DB 0.5092	R-L 0.2495	Shan 0.2286	Wiener 0.1389	Whyte 0.1343
Sp06	MinDist	Krishnan 0.2222	Goldstein 0.3351	No-DB 0.4545	R-L 0.5773	Shan 0.6313	Wiener 0.7785	Whyte 0.8913
	ROC Area	Krishnan 0.6808	Goldstein 0.6671	No-DB 0.4882	Shan 0.2569	R-L 0.2294	Wiener 0.1349	Whyte 0.1064
Sp07	MinDist	Krishnan 0.2261	Goldstein 0.3329	No-DB 0.4292	R-L 0.5864	Shan 0.6999	Wiener 0.7477	Whyte 0.7779
	ROC Area	Krishnan 0.7125	Goldstein 0.5994	No-DB 0.5352	Shan 0.2925	Whyte 0.2477	R-L 0.2318	Wiener 0.1419
Sp09	MinDist	Goldstein 0.2681	Krishnan 0.3267	No-DB 0.4784	R-L 0.6221	Whyte 0.6260	Shan 0.7301	Wiener 0.7649
	ROC Area	Goldstein 0.6318	Krishnan 0.6003	No-DB 0.4599	Whyte 0.3066	R-L 0.2445	Shan 0.2191	Wiener 0.1585
S.Rhex	MinDist	Whyte 0.3644	Shan 0.5443	Krishnan 0.5796	Goldstein 0.6153	No-DB 0.6841	Wiener 0.7146	R-L 0.7946
	ROC Area	Whyte 0.4985	Shan 0.3731	Krishnan 0.3232	No-DB 0.2928	Goldstein 0.2906	Wiener 0.2461	R-L 0.2200

For all the motion blur magnitudes of 1-D uniform dataset, Krishnan's deblurring method is the most successful method in combination with the Harris corner feature detector. When compared with respect to the no deblurring case, Krishnan's method achieves an increase between 0.05 to 0.40 in the FD-AROC metric. However, for the SensoRHex dataset with non-uniform blur, the performance increase with Krishnan deblurring is not more than 0.02. When combined with the FAST detector in 1-D uniform dataset, Krishnan competes with Goldstein. When compared with the no deblurring case, Krishnan improves the FAST detector performance from 0.09 to 0.19 and Goldstein from 0.06 to 0.18 in terms of the performance metric proposed. However, similar to the Harris detector case, for the SensoRHex robot data, Krishnan can only contribute 0.03 where Goldstein shows no improvement when combined with the FAST detector. This shows that, the FD-AROC performance of Krishnan degrades dramatically for nonuniform motion blur cases with both Harris and FAST detectors. Similarly, although Goldstein's method work well with FAST for 1-D uniform motion blur dataset, its performance drops slightly below the no deblurring case. On the other hand, Shan and Whyte's methods are distinguished well over the other methods for the nonuniform motion blur dataset.

To summarize, the results discussed above show that the selection of a motion deblurring method for improving the performance of a feature detector depends on both the properties of motion blur (hence the various parts of the dataset) and the motion blur magnitude. Therefore the selected methods are tested on the collected motion blur dataset that includes controlled 1-D linear uniform together with nonlinear, nonuniform motion blurred video. It is also demonstrated that a motion deblurring method that favors a feature detector may not perform as well for other feature detectors showing that there is no one universal deblurring method for the application. We believe that the obtained results can be used as a selection guide for deciding on the motion deblurring method to use for a particular feature based computer vision application in legged robotics and other legged platforms. The proposed feature based FD-AROC metric we believe can be used as a reliable performance measure that can be utilized in this field for benchmarking for all proposed motion deblurring methods. Finally, we would like to note that the dataset includes a multi-sensor aspect that we are in the process of using in our ongoing research. In particular, we believe that for video processing on mobile platforms, the motion data from inertial sensors, which is part of our presented dataset, can very successfully be used for aiding image processing and computer vision algorithms in general and motion deblurring in particular.

5.2 Multi-Frame Motion Deblurring Results

The comparison of selected single frame methods based on the FD-ROC analysis were presented in Section 5.1. In this section, the multi-frame deblurring method explained in Chapter 2 is tested on real test images obtained from our 1-D motion setup and compared according to the FD-ROC based feature detection performance metric.

Similar to the experiments on single frame deblurring methods, the proposed multi-frame deblurring method is tested with different motion blur magnitudes. The results are compared with the best performing single frame deblurring method and no deblurring applied images in the linear uniform motion blur dataset.

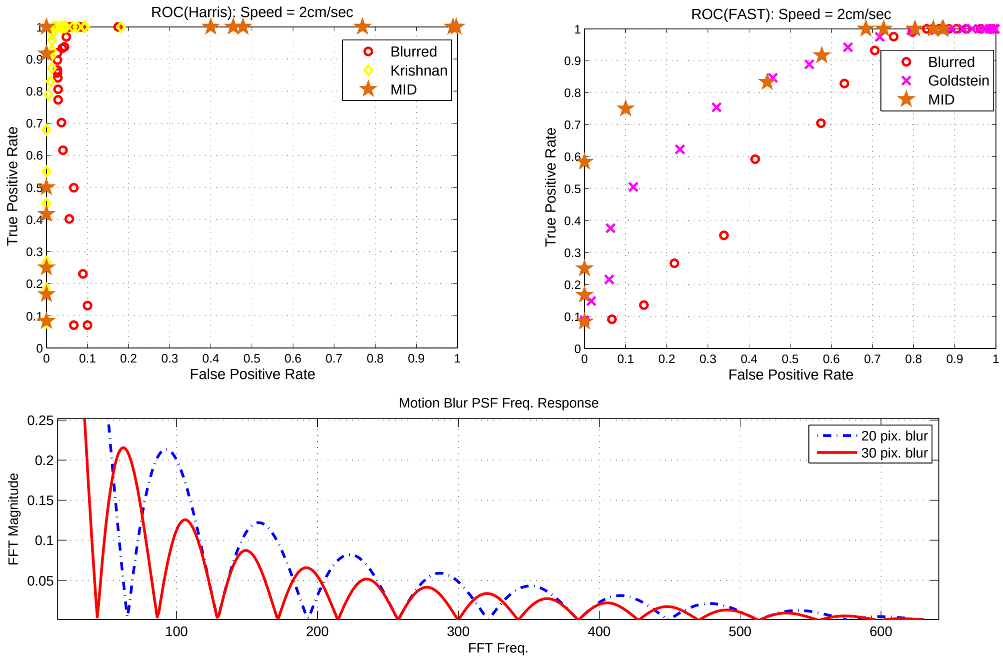


Figure 5.12: ROC analysis of Harris(left) and FAST(right) feature detectors under multi-frame deblurring method, best performing single frame motion deblurring method and no deblurring applied images in the linear uniform 20 pixel motion blur dataset.

The results obtained with 20 pixel blurred images are presented in Fig. 5.12. ROC analysis of Harris feature detector is shown on top-right graph and FAST feature detector is shown on top-left graph. The multi-frame deblurring method is applied using two image frames; one with 20 pixels and the other with 30 pixels motion blur magnitudes. The frequency response spectrum of 20 and 30 pixels motion blur PSFs are shown on bottom graph in Fig. 5.12. As can be seen from the frequency spectrum, the 20 pixel motion blur PSF has zero response at 10 frequencies and 30 pixel motion blur PSF has zero response at 15 frequencies. However, when multi-frame deblurring is used, then half of the zeros of 20 pixel motion blur PSF is compensated with 30 pixel motion blur PSF hence resulting in zero responses at only 5 frequencies. This leads to less image information to be lost and a low amplification factor for restoration which increases the signal to noise ratio (SNR). Therefore, a perfect detection rate (FD-AROC=1) could be achieved with Harris feature detector which is better than both the no deblurring case and deblurring with the best performing single frame deblurring method by Krishnan. For FAST feature detector, the multi-frame deblurring method performs better than the best performing single frame motion deblurring method by Goldstein. Although the detection rates for FAST detector seem to have a lower performance than the Harris, the application of deblurring yields a significant increase in the detection performance when compared to the no deblurring case.

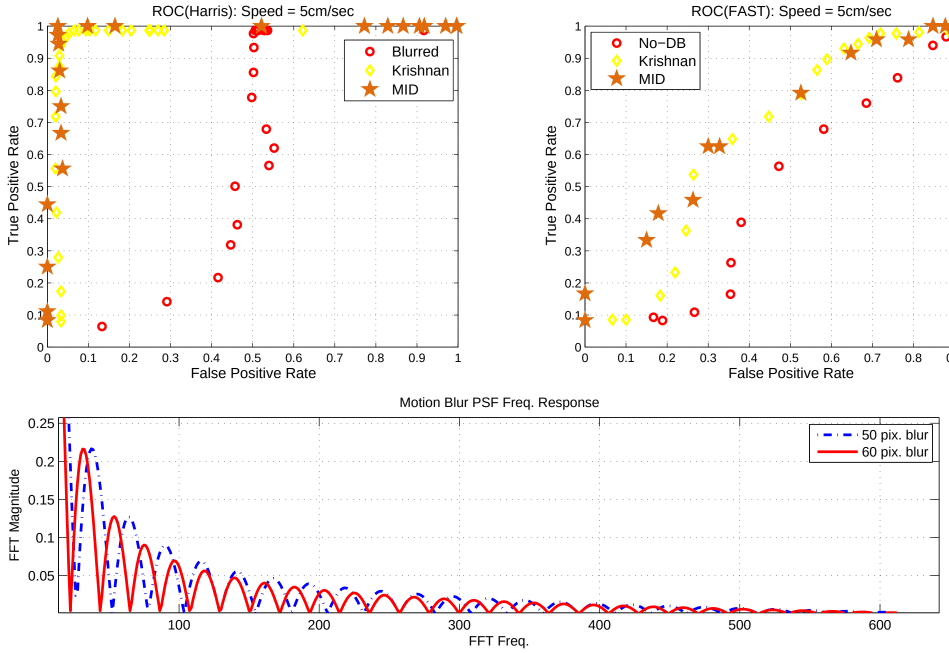


Figure 5.13: ROC analysis of Harris(left) and FAST(right) feature detectors under multi-frame deblurring method, best performing single frame motion deblurring method and no deblurring applied images in the linear uniform 50 pixel motion blur dataset.

Another example result obtained with relatively higher motion blur magnitude of 50 pixels is presented in Fig. 5.13. The multi-frame deblurring method is applied using two image frames; one with 50 pixels and the other with 60 pixels motion blur magnitudes. The frequency response spectrum of 50 and 60 pixels motion blur PSFs are shown on bottom graph in Fig. 5.13. It can be noticed that the number of zero response frequencies of PSFs are increasing with higher motion blur magnitudes as explained in Section 2.2. The response amplitudes are also getting lower which requires higher amplification factor in restoration. The obtained results show that the proposed multi-frame deblurring method improves the feature detection performance of both Harris and FAST detectors when compared to the no deblurring case. For Harris and FAST feature detectors, the multi-frame deblurring method performs slightly better than the best performing single frame motion deblurring method by Krishnan.

The application of multi-frame deblurring method with linear PSF estimation on the 1D linear motion blur dataset yields satisfactory results as presented above. How-

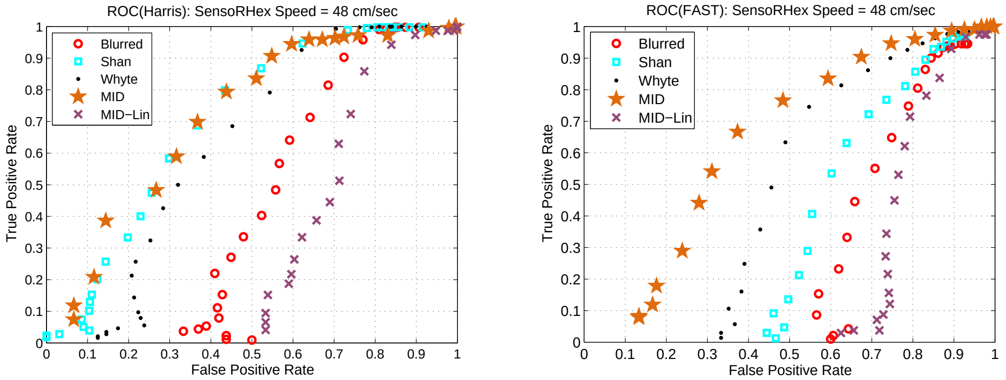


Figure 5.14: ROC analysis of Harris(left) and FAST(right) feature detectors under multi-frame deblurring method(with linear and nonlinear PSF estimation), two of the best performing single frame motion deblurring methods and no deblurring applied images in the SensoRHex motion blur dataset.

ever, when it is tested on the nonlinear SensoRHex motion blur dataset, the obtained results were even worse than the no-deblurring cases as shown in Fig. 5.14. Therefore, instead of linear PSF estimation, we made a nonlinear PSF estimation from the inertial measurement sensors. The combination of multi-frame deblurring together with the nonlinear PSF estimation improved the feature detection performance results dramatically both for Harris and FAST feature detectors. In Harris, multi-frame deblurring performs as good as Shan's method. In FAST detector, the improvement is much more considerable. The performance of multi-frame deblurring method with nonlinear PSF estimation is considerably better than the best performing single-frame deblurring method by Whyte et.al.

5.3 The Effect of Motion Blur and Deblurring on Feature Descriptors

In the previous section, we presented the effect of motion blur and deblurring on the detection of corner features. An important indicator which is as important as the detection of a feature is to measure how much a feature descriptor is effected(distorted or recovered) from motion blur and deblurring. Especially in tracking kind of applications the matching performance of features depends on the descriptor of a feature not to be changed(distorted) too much. Therefore, our aim in this section is to analyze the effect of motion blur and deblurring on the feature descriptors.

There are a number of feature descriptors available in literature such as SURF, SIFT. SIFT(Scale Invariant Feature Transform) is one of the most popular methods for detecting and describing image features that are invariant to image translation, scaling and rotation. Therefore, we choose SIFT features for analyzing the effect of motion

Table 5.3: Number of SIFT keypoints that can be detected on sharp, blurred(No-DB) and deblurred images with the given motion deblurring methods.

DATA	Number of Detected SIFT Keypoints							
	Sharp	**No-DB**	**R-L**	**Wiener**	**Goldst.**	**Shan**	**Krishn.**	**Whyte**
Sp02	3690	163	1224	2297	4239	639	2465	2129
Sp03	3690	115	677	1399	4568	385	1399	1524
Sp05	3690	61	352	768	3607	269	981	1609
Sp06	3690	47	306	620	2743	300	830	1331
Sp07	3690	40	276	519	2547	305	740	689
Sp09	3690	30	211	325	2728	187	664	822

blur and deblurring on a feature descriptor using the SIFT Matlab codes provided by the author of [31]. Firstly, we analyze how the number of detected SIFT features varies under different motion blur magnitudes. Our 1D linear motion blur dataset with a real office scene is used as the testing data. The number of detected SIFT features on sharp, blurred(No-DB) and deblurred images using different deblurring methods are summarized in Table 5.3. The parameters of the method is kept constant at default values. As can be seen from the table, the blurring decreases the number of detected SIFT keypoints dramatically as the blur magnitude increases. Another consequence is the effect of deblurring methods on the number of SIFT keypoints that can be detected. It can easily seen that all of the deblurring methods increases the number of SIFT keypoints. However, it is not possible to say that the deblurring methods recover the exact SIFT keypoints as detected from the sharp image by just looking at the increased number of detected keypoints. Therefore, we need to compare the vector distance between the descriptor vectors to determine the amount of distortion in the descriptors introduced by the blurring/deblurring process. For this, we selected the similar keypoints from both the sharp image and the blurred/deblurred image that matches. The mean distances between the descriptor vectors of the SIFT keypoints are shown in Table 5.4. The first observation from the table is the increasing mean error of SIFT keypoint descriptors as the motion blur magnitude increases as expected. Similar behavior is valid for the deblurred images. The mean error increases for the deblurred images with increasing blur. However, when compared to the no deblurring case, it can easily be observed that deblurring reduces the mean error for most of the cases. There are also cases where deblurring increases mean error. In fact, there is no deblurring method except Shan's that decreases the mean error for all blur magnitudes in the test data set. This result (descriptor mean error), when considered together with the number of keypoints detected given in Table 5.3, concludes that although the deblurring methods help finding more keypoints than the no deblurring case, the mean error of the keypoint descriptors are increased in high blur cases because of the high amount of distortion introduced by these deblurring methods. High descriptor error means that the descriptor is distorted and it differs more from the original descriptor of the keypoint which is detected on the sharp image. However, increasing mean error should not be considered alone for evaluating a deblurring method. Because, although the descriptor error is increased it may be still possible to match the keypoints. Therefore, the mean error should be considered together with the number of keypoints that can be detected.

Table 5.4: The mean error of matching SIFT descriptor vectors on blurred and deblurred images with the given motion deblurring methods.

DATA	**SIFT Descriptor Vectors Mean Error (Lower is better)**						
	No-DB	**R-L**	**Wiener**	**Goldst.**	**Shan**	**Krishn.**	**Whyte**
Sp02	0.112	0.101	0.100	0.083	0.100	0.122	0.114
Sp03	0.116	0.110	0.110	0.111	0.109	0.135	0.125
Sp05	0.146	0.128	0.130	0.120	0.126	0.144	0.139
Sp06	0.175	0.157	0.165	0.157	0.130	0.160	0.147
Sp07	0.194	0.239	0.247	0.222	0.184	0.255	0.179
Sp09	0.339	0.302	0.297	0.429	0.318	0.634	0.320

For detecting corner features using the Harris corner detector, every pixel in the image is assigned a cornerness score. A higher cornerness score means a more likelihood a pixel to be a corner. However, when the image is motion blurred we have shown in previous sections that some of the corner features cannot be detected. When deblurring methods are applied to a motion blurred image, then some of the undetectable corner features can be recovered. In this section, we analyze how the cornerness scores of corner features are effected due to motion blurring and deblurring methods applied. In Table 5.5, the cornerness scores of ground truth corner pixels are shown for sharp, blurred and deblurred checkerboard images with the given motion deblurring methods. The tests are repeated for different motion blur magnitudes. It can be seen from the table that sharp image corner pixels have a mean cornerness score of 1.654. However, the scores decreases dramatically as motion blur magnitude is increased. When deblurring methods are applied, the cornerness scores are increased for all of the deblurring methods.

5.4 Computational Complexities of the Tested Motion Deblurring Methods

Computation times of motion deblurring methods are important especially for applications where real-time operation or fast computation is required and the computation resources are limited. Robotic applications are generally among these type of applications. Therefore in this section we present the measured computation times of the tested motion deblurring methods. For an accurate computation time measurement

Table 5.5: The cornerness scores of sharp, blurred and deblurred checkerboard images with the given motion deblurring methods.

DATA	Cornerness Scores (Higher is better)							
	Sharp	**No-DB**	**R-L**	**Wiener**	**Goldst.**	**Shan**	**Krishn.**	**Whyte**
Sp02	1.654	0.043	0.569	0.172	1.017	0.304	1.711	0.450
Sp03	1.654	0.002	0.421	0.076	0.849	0.173	1.798	0.027
Sp05	1.654	0.000	0.141	0.010	0.613	0.069	1.148	0.013
Sp06	1.654	-0.002	0.097	0.002	0.113	0.037	0.382	0.004
Sp07	1.654	-0.002	0.049	0.001	0.003	0.003	0.160	0.002
Sp09	1.654	-0.003	0.013	-0.001	0.001	0.001	0.007	0.001

and comparison, all of the methods should be implemented with same programming language and run on same computation platform. Instead of implementing all of the methods ourselves using the same software language, we prefer using the codes provided by the authors of the methods since reimplementing all of the methods requires too much time and effort which is not the main focus of this study. However, the implementations of the deblurring methods used are done using different software languages. Some of them are implemented in Matlab and others are provided only in precompiled executables. Therefore, the computation time analysis provided in this section are coarse and can only be used for gaining insight about the computation time of each method individually.

Table 5.6: The computation times required for the tested motion deblurring methods to run on 1280x960 resolution images.

	Computation Times (seconds)						
	R-L (Matlab)	**Wiener** (Matlab)	**Goldstein** (Matlab)	**Shan** (Exe)	**Krishnan** (Matlab)	**Whyte** (Exe)	**MID** (Matlab)
Sp03	5.2	0.36	31.8	26.1	169.2	1017.2	0.28
Sp05	5.2	0.36	80.4	35.8	224.0	1052.7	0.29

The measured computation times of the tested motion deblurring methods used are presented in Table 5.6. The computation times are measured on 1280x960 pixels resolution images having two different motion blur magnitudes. It can be easily recognized that the computation time of some of the methods depend on the motion blur magnitude whereas the others are independent of how much the image is blurred. For example, Richardson-Lucy, Wiener, Whyte methods are computed on both 30 pixels and 50 pixels motion blurred images at equal or close time durations whereas Goldstein, Shan and Krishnan's methods require significantly longer computation times for the higher motion blur magnitude. For Multi-image deblurring (MID), the presented method is tested with pairs of motion blurred frames. For Sp03 case, a pair of images one with 30 pixels and the other with 50 pixels motion blur are used. Similarly, for Sp05 case, the reported computation time involves computation with 50 and 60 pixels motion blurred image pairs. Since the presented method does not depend much on the blur magnitude, the computation times of Sp03 and Sp05 pairs are nearly the same. All of the computation complexity tests are carried out on the same computer having Intel i7-4790 3.6GHz CPU and 16GB RAM running Windows 8 operating system.

CHAPTER 6

CONCLUSIONS AND FUTURE STUDIES

6.1 Conclusions

In this study, we focused on the motion blur and its compensation in legged mobile robotic platforms in order to increase the feature detection performance. The properties of motion blur in SensoRHex robot is analyzed and possible deblurring methods are explored. We begin with accurately and objectively evaluating the performance of single-frame motion deblurring methods for legged moving platforms. We are motivated by the significant presence of motion-blur for camera systems on such platforms and the resulting negative effects for computer vision applications particularly for dexterous legged robots.

We identified that feature detection performance is crucial for most higher level computer vision algorithms and the fact that existing performance measures for deblurring do not measure this. We hence proposed a direct feature detection ROC based metric and demonstrated the inaccuracy of available image quality metrics from the literature for indicating the quality of an image in terms of feature detection performance and so as the computer vision applications that utilize them. The ROC analysis approach is adapted to the problem and the “FD-AROC” metric is proposed for benchmarking.

Due to the lack of a motion blur dataset in legged robotics in literature, a comprehensive dataset is constructed, with motion-blurred images having both controlled linear, uniform and realistic nonuniform (robotic) motion components. The dataset includes measurements of camera motion from FOG and MEMS gyroscope sensors together with an external motion tracking data as the ground truth. Two feature detectors (Harris and FAST) are considered for experiments. The performances of six deblurring methods, proposed by Goldstein, Krishnan, Richardson-Lucy, Shan, Whyte and Wiener are compared in terms of the FD-AROC metric.

It is a fact that motion blur blur PSFs have zero response frequencies in frequency domain that causes the corresponding frequency information to be totally lost in the blurred image. From the properties of motion blur obtained from our robot, we have discovered that consecutive image frames have different motion blur PSFs. We analyzed the zero response frequencies of linear PSFs and derived the conditions of having overlapping and non-overlapping zero response frequencies. Utilizing this fact, we proposed a multi frame motion deblurring method for motion blur in our legged mobile platform. The proposed method is tested on 1-D linear uniform dataset using two image frames having different amounts of motion blur. The results are compared with the best performing single frame motion deblurring method and the no deblurring applied images in terms of the proposed FD-AROC based feature detection performance metric. The obtained results show that multi-frame deblurring method significantly increases the performances of Harris and FAST corner feature detector performances. It also slightly improves the performance obtained from the best performing single frame motion deblurring methods. However, the application of multi-frame deblurring method to the nonlinear and nonuniform motion blurred images does not yield good results. Therefore we studied on 2-D nonlinear PSF estimation from motion data and could obtain accurate 2-D nonlinear PSFs. We believe that the usage of nonlinear PSF estimation together with multi-frame deblurring can yield better results for the motion blurred images taken from the legged mobile robotic platform. Another factor that limits the performance of the deblurring method in legged robot image dataset is the existence of non-uniformity (spatially variance) of PSFs on the image. Our notion on this issue is that, non-uniformity of the PSF is the fundamental effect that limits the performance of deblurring. We also state in our future works in Section 6.2 that the future studies on extending the multi-frame deblurring method to non-uniform motion blur cases may significantly contribute to the performance of the method.

6.2 Future Work

This thesis presents our initial studies on motion blur and its compensation on a legged mobile robotic platform in order to increase the performances of computer vision applications that utilize image features such as corners. The proposed FD-AROC metric is used to demonstrate the performances of a number of deblurring methods together with two fundamental corner feature detectors which are Harris and FAST. This metric can be further used for benchmarking all of the other motion deblurring metrics that are already available in literature or will be proposed in the future.

The proposed metric is also appropriate to measure the performance under motion blurred and deblurred images using other future detectors. SUSAN, SIFT, SURF, MSER are such examples to other feature detectors. Some of these detectors are used for corner detection similar to the feature detectors we used in this study and others are used for blob detection or detection of higher dimensional feature vectors. A matching based performance comparison can be used for those detectors that do not detect a visual feature such as a corner.

We obtained a comprehensive motion blur dataset for current and future studies of motion blur, deblurring and other computer vision applications on a legged mobile robot platform. Therefore, it is possible to utilize this dataset to develop and test other computer vision applications in the future. Although the dataset includes many types of controlled and uncontrolled motion blur obtained using different kinds of scenes and speeds, all of the data are collected indoors due to the difficulties to establish the current data collection setup outdoors. All of our tests are carried out on a flat and rigid laboratory floor. Although it is possible to operate the SensoRHex robot outdoors together with its onboard sensors such as camera and gyro; OptiTrack, the external motion tracking system, is not appropriate for outdoor operation. An outdoor motion tracking method can be used if such a dataset is needed.

We compared the single frame motion deblurring methods and proposed a multi frame deblurring method for legged robotics applications. However, the performance of the multi frame deblurring method can be increased if the spatially invariant (uniform) motion blur assumption can be extended to nonuniform cases. Our analysis revealed that the images taken from a legged robotic platform includes a considerable amount of nonuniform motion blurred frames. Therefore, we think that this improvement can increase the feature detection performance of motion blurred frames in legged robotics vision applications.

Our study considers the images captured from the SensoRHex robot without interfering with the image capturing process. The proposed multi frame deblurring method utilizes the fact that most of the consecutive image frames have different motion blur magnitudes and therefore different motion blur PSFs. However, this assumption may not be applicable for all the frames. In that cases the multi frame deblurring method can not be utilized or will result in a poor performance. We have discussed in Section 1.1 that some of the methods in literature modifies the image capturing process to obtain a motion blur PSF having favorable properties for the motion deblurring process. Instead of keeping the exposure constant as in our case, the exposure can be modified in realtime according to the motion of the robot to obtain more suitable image frames for multi frame motion deblurring.

REFERENCES

[1] A. Agrawal and R. Raskar. Resolving objects at higher resolution from a single motion-blurred image. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, June 2007.

[2] A. Agrawal, Y. Xu, and R. Raskar. Invertible motion blur in video. *ACM Trans. Graph.*, 28(3):95:1–95:8, July 2009.

[3] Y. Bando, B.-Y. Chen, and T. Nishita. Motion deblurring from a single image using circular sensor motion. *Computer Graphics Forum*, 30(7):1869–1878, 2011.

[4] Y. Bando, H. Holtzman, and R. Raskar. Near-invariant blur for depth and 2d motion via time-varying light field analysis. *ACM Trans. Graph.*, 32(2):13:1–13:15, Apr. 2013.

[5] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. Speeded-up robust features (surf). *Computer vision and image understanding*, 110(3):346–359, 2008.

[6] M. Ben-Ezra and S. K. Nayar. Motion-based motion deblurring. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(6):689–698, June 2004.

[7] M. O. Öğücü. *Modeling and Control of the Three Degrees-of-freedom Parallel Manipulated Robotic Sensor Head*. PhD thesis, Middle East Technical University, 2014.

[8] Canon. *What is optical shift image stabilizer?* Canon, 2013.

[9] D. M. Chandler and S. S. Hemami. Vsnr: A wavelet-based visual signal-to-noise ratio for natural images. *IEEE Transactions on Image Processing*, 16(9):2284–2298, Sept 2007.

[10] S. Dai and Y. Wu. Motion from blur. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8, June 2008.

[11] P. Favaro and S. Soatto. *3D Shape Estimation and Image Restoration: Exploiting Defocus and Motion-Blur*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[12] R. Fergus, B. Singh, A. Hertzmann, S. T. Roweis, and W. T. Freeman. Removing camera shake from a single photograph. *ACM Trans. Graph.*, 25(3):787–794, July 2006.

[13] R. Ferzli and L. J. Karam. A no-reference objective image sharpness metric based on the notion of just noticeable blur (jnb). *Trans. Img. Proc.*, 18(4):717–728, Apr. 2009.

[14] A. Goldstein and R. Fattal. Blur-kernel estimation from spectral irregularities. In *European Conference on Computer Vision (ECCV)*, pages 622–635. Springer, 2012.

[15] J. R. Goulding. Biologically-inspired image-based sensor fusion approach to compensate gyro sensor drift in mobile robot systems that balance. In *Multisensor Fusion and Integration for Intelligent Systems (MFI), 2010 IEEE Conference on*, pages 102–108, Sept 2010.

[16] C. Harris and M. Stephens. A combined corner and edge detector. In *Alvey vision conference*, volume 15, page 50. Citeseer, 1988.

[17] A. Huster, E. W. Frew, and S. M. Rock. Relative position estimation for auvs by fusing bearing and inertial rate sensor measurements. In *OCEANS '02 MTS/IEEE*, volume 3, pages 1863–1870 vol.3, Oct 2002.

[18] A. Huster and S. M. Rock. Relative position estimation for manipulation tasks by fusing vision and inertial measurements. In *OCEANS, 2001. MTS/IEEE Conference and Exhibition*, volume 2, pages 1025–1031 vol.2, 2001.

[19] Q. Huynh-Thu and M. Ghanbari. Scope of validity of psnr in image/video quality assessment. *Electronics letters*, 44(13):800–801, 2008.

[20] W. Jiang, D. Zhang, and H. Yu. Sensor-assisted image deblurring of consumer photos on smartphones. *Multimedia and Expo (ICME), 2014 ...*, 2014.

[21] M. C. P. Joseph L. Fleiss, Bruce Levin. *Statistical Methods for Rates and Proportions*. Wiley, 2004.

[22] N. Joshi, S. B. Kang, C. L. Zitnick, and R. Szeliski. Image deblurring using inertial measurement sensors. *ACM Trans. Graph.*, 29(4):30:1–30:9, July 2010.

[23] M. D. Kim and J. Ueda. Dynamics-based motion de-blurring for a pzt-driven, compliant camera orientation mechanism. *Int. J. Rob. Res.*, 34(4-5):653–673, Apr. 2015.

[24] R. Köhler, M. Hirsch, B. Mohler, B. Schölkopf, and S. Harmeling. Recording and playback of camera shake: Benchmarking blind deconvolution with a real-world database. In *Proceedings of the 12th European Conference on Computer Vision - Volume Part VII*, ECCV'12, pages 27–40, Berlin, Heidelberg, 2012. Springer-Verlag.

[25] D. Krishnan and R. Fergus. Fast image deconvolution using hyper-laplacian priors. In Y. Bengio, D. Schuurmans, J. D. Lafferty, C. K. I. Williams, and

A. Culotta, editors, *Advances in Neural Information Processing Systems 22*, pages 1033–1041. Curran Associates, Inc., 2009.

[26] A. Levin, P. Sand, T. S. Cho, F. Durand, and W. T. Freeman. Motion-invariant photography. *ACM Trans. Graph.*, 27(3):71:1–71:9, Aug. 2008.

[27] J. Lewis. Fast normalized cross-correlation. In *Vision interface*, volume 10, pages 120–123, 1995.

[28] T. Li, D. W. Zhang, Y. Fu, and M. Q.-H. Meng. Motion blur removal for humanoid robots. In *2012 IEEE International Conference on Automation and Logistics*, pages 378–381. IEEE, 2012.

[29] L. Liang, J. Chen, S. Ma, D. Zhao, and W. Gao. A no-reference perceptual blur metric using histogram of gradient profile sharpness. In *2009 16th IEEE International Conference on Image Processing (ICIP)*, pages 4369–4372, Nov 2009.

[30] H. Y. Lin. Vehicle speed detection and identification from a single motion blurred image. In *Application of Computer Vision, 2005. WACV/MOTIONS '05 Volume 1. Seventh IEEE Workshops on*, volume 1, pages 461–467, Jan 2005.

[31] D. G. Lowe. Object recognition from local scale-invariant features. In *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, volume 2, pages 1150–1157. Ieee, 1999.

[32] L. B. Lucy. An iterative technique for the rectification of observed distributions. *Astronomical Journal*, 1974.

[33] P. Marziliano, F. Dufaux, S. Winkler, and T. Ebrahimi. A no-reference perceptual blur metric. In *Image Processing. 2002. Proceedings. 2002 International Conference on*, volume 3, pages III–57–III–60 vol.3, 2002.

[34] M. Mutlu. A novel real-time inertial motion blur metric with applications to motion blur compensation. Master's thesis, Middle East Technical University, 2014.

[35] N. D. Narvekar and L. J. Karam. A no-reference perceptual image sharpness metric based on a cumulative probability of blur detection. In *Quality of Multimedia Experience, 2009. QoMEx 2009. International Workshop on*, pages 87–91. IEEE, 2009.

[36] K. Ni, Z. Sun, and N. Bliss. *Real-time global motion blur detection*, pages 3101–3104. 2012.

[37] Nikon. *Vibration-Reduction (VR)*. Nikon, 2008.

[38] K. Park, S. Shin, H. G. Jeon, J. Y. Lee, and I. S. Kweon. Motion deblurring using coded exposure for a wheeled mobile robot. In *Ubiquitous Robots and Ambient Intelligence (URAI), 2014 11th International Conference on*, pages 665–671, Nov 2014.

[39] Quanser. Quanser linear servo base unit. `http://www.quanser.com/products/linear_servo_ip02`, accessed August 2016.

[40] R. Raskar, A. Agrawal, and J. Tumblin. Coded exposure photography: Motion deblurring using fluttered shutter. *ACM Trans. Graph.*, 25(3):795–804, July 2006.

[41] I. S. Reed and X. Yu. Adaptive multiple-band cfar detection of an optical pattern with unknown spectral distribution. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 38(10):1760–1770, 1990.

[42] W. H. Richardson. Bayesian-based iterative method of image restoration$*$. *J. Opt. Soc. Am.*, 62(1):55–59, Jan 1972.

[43] B. Rohani, Y. Yazicioglu, M. Mutlu, O. Ogucu, E. Akgul, and A. Saranli. Lagrangian based mathematical modeling and experimental validation of a planar stabilized platform for mobile systems. *Journal of Computational and Applied Mathematics*, 259, Part B:955 – 964, 2014. Recent Advances in Applied and Computational Mathematics: ICACM-IAM-METUOn the occasion of 10th anniversary of the foundation of Institute of Applied Mathematics, Middle East Technical University, Ankara, Turkey.

[44] E. Rosten and T. Drummond. Fusing points and lines for high performance tracking. In *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, volume 2, pages 1508–1515. IEEE, 2005.

[45] E. Rosten and T. Drummond. Machine learning for high-speed corner detection. In *Proceedings of the 9th European Conference on Computer Vision - Volume Part I*, ECCV'06, pages 430–443, Berlin, Heidelberg, 2006. Springer-Verlag.

[46] U. Saranli, M. Buehler, and D. E. Koditschek. Rhex: A simple and highly mobile hexapod robot. *The International Journal of Robotics Research*, 20(7):616–631, 2001.

[47] S. Schuon and K. Diepold. Comparison of motion de-blur algorithms and real world deployment. *Acta Astronautica*, 64(11–12):1050 – 1065, 2009.

[48] Q. Shan, J. Jia, and A. Agarwala. High-quality motion deblurring from a single image. *ACM Transactions on Graphics (SIGGRAPH)*, 2008.

[49] H. R. Sheikh and A. C. Bovik. A visual information fidelity approach to video quality assessment. In *The First International Workshop on Video Processing and Quality Metrics for Consumer Electronics*, pages 23–25. Citeseer, 2005.

[50] D. Sun, S. Roth, and M. J. Black. Secrets of optical flow estimation and their principles. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 2432–2439. IEEE, 2010.

[51] Z. Taojia, Z. Hong, and Y. Ding. A no-reference quality metric for blur image. In *Green Computing and Communications (GreenCom), 2013 IEEE and Internet of Things (iThings/CPSCom), IEEE International Conference on and IEEE Cyber, Physical and Social Computing*, pages 1813–1816, Aug 2013.

[52] F. Uzer. Camera motion blur and its effects on feature detectors. Master's thesis, Middle East Technical University, 2010.

[53] Z. Wang and A. C. Bovik. A universal image quality index. *IEEE Signal Processing Letters*, 9(3):81–84, March 2002.

[54] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

[55] Z. Wang and E. P. Simoncelli. Local phase coherence and the perception of blur. In *Advances in neural information processing systems*, page None, 2003.

[56] O. Whyte, J. Sivic, A. Zisserman, and J. Ponce. Non-uniform deblurring for shaken images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2010.

[57] Y. Yitzhaky, I. Mor, A. Lantzman, and N. S. Kopeika. Direct method for restoration of motion-blurred images. *J. Opt. Soc. Am. A*, 15(6):1512–1519, Jun 1998.

[58] S. Yousaf and S. Qin. Approach to metric and discrimination of blur based on its invariant features. In *2013 IEEE International Conference on Imaging Systems and Techniques (IST)*, pages 274–279, Oct 2013.

CURRICULUM VITAE

PERSONAL INFORMATION

Surname, Name: Gültekin, Gökhan Koray

Date and Place of Birth: 1984, Karşıyaka / İzmir

Nationality: Turkish (TC)

Language: Turkish(native), English(fluent), German(beginner)

e-mail: gultekin@eee.metu.edu.tr

EDUCATION

Degree	**Institution**	**Year of Graduation**
M.S.	Middle East Technical University Electrical and Electronics Engineering Thesis:An FPGA Based High Performance Optical Flow Hardware Design for Mobile Robotic Applications Advisor: Dr. Afşar Saranlı	2010
B.S.	Başkent University(Full Scholarship) Electrical and Electronics Engineering Graduation Ranking: 2^{nd}	2007
High School	Etimesgut Anatolian High School	2002

PROFESSIONAL EXPERIENCE

Year	**Place**	**Enrollment**
2009-2016	Middle East Technical University Department of Electrical and Electronics Engineering	Teaching & Research Assistant
2010-2013	TÜBİTAK 1001 Project ID:110E120 Acoustic perception assisted robust visual processing and perception for dynamically dextrous legged robots with application to autonomous navigation	Researcher
2008-2010	TÜBİTAK 1001 Project ID:106E089 Bacaklı robotlarda mobil arazi uygulamalarına yönelik öz durum kestirimi ve geri beslemeli dinamik davranışlar	Researcher
2008-2010	TÜBİTAK 1001 Multipurpose National Unmanned Underwater Vehicle	Researcher
2007-2008	Başkent University Department of Electrical and Electronics Engineering	Teaching & Research Assistant

AWARDS

1. ODTÜ GRW (Graduate Research Workshop) 2016
Best Research Poster and 2^{nd} Best Oral Presentation Awards

2. ODTÜ GRW (Graduate Research Workshop) 2015
Best Research Poster and 2^{nd} Best Oral Presentation Awards

3. Awards in International Robotics Contests:
  i. IEEE R8 Student Robotics Contest 2006, Netherlands, 1^{st} ranking

4. Awards in National Robotics Contests:
  i. MEB Robotics Contest 2007, Ankara, 3^{rd} ranking
  ii. ODTÜ Robotics Contest 2006, Ankara, 1^{st} ranking
  iii. ODTÜ Robotics Contest 2005, Ankara, 2^{nd} ranking
  iv. ABU-Robocon (Asia-Pacific Broadcasting Union Robotics Contest) 2005, İstanbul, 2^{nd} ranking