DYNAMICS-BASED MOTION DE-BLURRING AND PANORAMIC IMAGE GENERATION FOR A COMPLIANT CAMERA ORIENTATION MECHANISM

A Thesis
Presented to
The Academic Faculty

by

Michael Duckjune Kim

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in
Mechanical Engineering

George W. Woodruff School of Mechanical Engineering
Georgia Institute of Technology
December 2016

DYNAMICS-BASED MOTION DE-BLURRING AND PANORAMIC IMAGE GENERATION FOR A COMPLIANT CAMERA ORIENTATION MECHANISM

Approved by:

Dr. Jun Ueda, Advisor
George W. Woodruff School of
Mechanical Engineering
Georgia Institute of Technology

Dr. Irfan Essa
School of Interactive Computing
Georgia Institute of Technology

Dr. Kok-Meng Lee
George W. Woodruff School of
Mechanical Engineering
Georgia Institute of Technology

Dr. Wayne Daley
Georgia Tech Research Institute

Dr. YongTae (Tony) Kim
George W. Woodruff School of
Mechanical Engineering
Georgia Institute of Technology

Date Approved: 30 August 2016

To my wife Hana and my son Nathan

ACKNOWLEDGEMENTS

I am pleased to thank many people who have supported me.

First and foremost, I would like to thank my Lord and Savior Jesus Christ for the wonderful blessings. He has given me the strength and wisdom throughout this journey and my life.

I would like to thank my adivsor, Dr. Jun Ueda, for his immeasurable guidance throughout my doctoral study. His passion and excellence for research and teaching has inspired and helped me become a professional researcher. He has been a true mentor and it has been a great pleasure being his student. I will always be grateful for his insight and support.

My sincere thanks to my reading committee, Dr. Kok-Meng Lee, Dr. YongTae (Tony) Kim, Dr. Irfan Essa, and Dr. Wayne Daley. I truly appreciate all the helpful suggestions you have given, both with regard to this research, and for my career in general.

I would like to thank my colleagues at Georgia Tech for their encouragement, valuable suggestions, and support. In particular, I would like to acknowledge Bio-Robotics and Human Modeling Laboratory. I would like to thank to the undergraduate who helped this work: Sean Kilbride. I would also like to express my gracious thanks to Dr. Sangil Lee, Dr. Jungyoul Lim, Dr. Yongjin Kim, and Minseok Ha.

Thank you to my friends in Atlanta. In particular, the friends I met at church. Your support has been invaluable to my family.

I would like to express my gratitude to my parents, Dr. Sangdae Kim and Joyce Kim, and my brother, Brian Kim. Thank you for your unending love, encouragement, and prayers.

Finally, I would like to thank my wife Hana and my son Nathan for your love, trust, support, and sacrifice. This accomplishment would not have been possible without you. Thank you for the prayers and being with me.

This study was made possible through generous funding and support in part by the National Science Foundation (Grant: Dynamical Systems CMMI-1300019) and Georgia Institute of Technology (Award: President's Fellowship) for the financial support on this study.

TABLE OF CONTENTS

DEDICATION . . . . . **iii**

ACKNOWLEDGEMENTS . . . . . **iv**

LIST OF TABLES . . . . . **x**

LIST OF FIGURES . . . . . **xi**

SUMMARY . . . . . **xix**

I INTRODUCTION **1**

1.1 Motivation 1

1.2 Overview 4

1.3 Dynamics-based Visual Processing 5

1.4 Testbed Device: Camera Orientation System 6

1.4.1 Mechanical Design 6

1.4.2 Quantization and Compliance in Actuation 8

1.4.3 Range of Motion 9

II LITERATURE SURVEY **11**

2.1 Camera Positioning Systems 11

2.2 Control of a Quantized or Complaint Actuation System 12

2.2.1 Compliant Actuation System 12

2.2.2 Quantized Actuation System 14

2.3 Image De-blurring 15

2.3.1 Image-based Approaches 15

2.3.2 Sensor-based Approaches 17

2.4 Generation of Panoramic View 18

2.5 Human Ocular System 19

III DYNAMICS-BASED IMAGE DE-BLURRING **20**

3.1 Saccade-like Motion in a Robotic Eye 20

3.1.1 Discrete Switching Input and Step Input 22
3.2 Image De-blurring 24
3.2.1 Dynamics-based Blur Kernel Estimation 24
3.2.2 Implementation of Dynamics-based Estimation 27
3.3 Sensitivity Analysis of the Dynamics-based Approach 29
3.4 Deconvolution 30
3.5 Evaluation Method 31
3.5.1 Compared Algorithms 31
3.5.2 Quantitative Evaluation 31
3.6 Experimental Results 32
3.6.1 Vibration Suppression Input and Step Input 32
3.6.2 Oscillatory Input 34
3.6.3 Quantitative Evaluation 42
3.7 Computation Time 43
3.8 Experiments with Different Scenes 45
3.9 Evaluation by OCR 55
3.10 Summary 61

IV SYSTEM ARCHITECTURE FOR VISUAL-MOTOR COORDINATION AND REAL-TIME PERFORMANCE 62

4.1 Methodology and System Architecture 63
4.1.1 Vision Sensor 64
4.1.2 FPGA 65
4.1.3 GPU 67
4.1.4 Real-Time Controller and Host PC 67
4.2 Real-Time Image de-blurring 68
4.2.1 Spatially invariant and deterministic kernel from pure rotational system 68
4.2.2 Instant dynamics-based blur kernel estimation 70
4.2.3 Real-Time Deconvolution 72

4.3 Experimental Results 74
4.4 Summary 77

V DYNAMICS-BASED PANORAMA GENERATION 78

5.1 Visual-Motor Coordination 78
5.2 Dynamics-based Stitching 80
5.2.1 Homography Estimation 81
5.2.2 Modified Deconvolution 82
5.3 Experimental Setup 83
5.4 Experimental Results 84
5.4.1 Evaluation 84
5.4.2 Computation Time 89
5.5 Response to External Disturbance 89
5.6 Summary 96

VI DISCRETE SWITCHING CONTROLLER FOR SMOOTH-PURSUIT LIKE MOVEMENTS 97

6.1 Cellular Actuator: Compliance and Quantization 97
6.2 Quantization Effects 98
6.3 Discrete Switching Commands 103
6.3.1 System Model and Modified Input 104
6.3.2 Tracking Commands 105
6.3.3 Validation of the Controller for Tracking 112
6.3.4 Vibration Suppression Commands 116
6.3.5 Validation of the Controller for Vibration Suppression 120
6.4 Experimental Setup for Tracking 122
6.5 Results and Discussions 124
6.5.1 Tracking a Single Object 124
6.5.2 Tracking Two Objects 126
6.6 Summary 128

VII CONCLUDING REMARKS 130

APPENDIX A — MODIFIED BLUR KERNEL FOR TRANSLATIONAL MOTION 134

APPENDIX B — EVALUATION OF THE DE-BLURRING METHODS USING A CHECKERBOARD 139

APPENDIX C — IMPROVED IMAGE PROCESSING BY DE-BLURRING 143

APPENDIX D — PANORAMIC IMAGE GENERATION 147

APPENDIX E — ROBOTIC VISION SYSTEM FIRMWARE . . 150

APPENDIX F — DESIGN OF A 2-DOF CAMERA POSITIONER 155

REFERENCES 157

LIST OF TABLES

1 Comparison between the biological system and the robotic vision system 22

2 Comparisons of Computation Times [sec] 44

3 Image sizes that were used to estimate computation times 45

4 Comparisons of Computation Times [sec] 46

5 Improvement in computation time by single channel approach 74

6 Conditions of the test sets 75

7 Comparisons of Computation Times using GPU for Color Images [ms] 75

8 Comparisons of Computation Times using GPU for Grayscale Images [ms] 75

9 Comparisons of Stitching Times per Image (GPU Computation) . . . 89

10 Comparisons of Stitching Times per Image (CPU Computation) . . . 89

11 Conditions of the external force applied on the system 94

12 Conditions of the external force applied on the system 94

13 RMS Position Errors [deg] 114

14 Test Conditions 122

15 Experimental Results of the Proposed Method 122

16 Experimental Conditions for Tracking Objects 124

17 Mean Peak Values of Image Gradients (Tracking a Single Object) . . 126

18 Mean Peak Values of Image Gradients (Tracking Two Objects) 128

19 Sampling rates of the key functions in FPGAs [kHz] 151

LIST OF FIGURES

1 Saccades (a) Conceptual drawing of motion blur due to saccades (b) Eye movements captured by a high-speed camera. 2

2 Robotic vision system 6

3 Piezoelectrically driven camera positioner system. The orientation of the camera mounted on the moving platen is controlled by a total of 32 PZT actuators. A tilting motion is generated when the rod is positioned by the antagonistic pair of the cellular actuators. 7

4 Motor unit recruitment. Each motor unit is operated in a binary manner and the action is generated by recruitment of individual units activated. (a) The cellular actuator consists of a total of 16 PZT actuators. A total of 16 PZT actuators is nested in a multi-layer fashion and the thin layers allow amplification. (b) The muscle consists of individual motor units. 8

5 Single-DOF motion of the camera orientation system: (a) camera at extreme left (b) camera at extreme right (c) camera at extreme left (d) camera at extreme right 10

6 A saccade-like motion with an open loop vibration suppression input. (a) Time response of a rapid point-to-point motion of the camera orientation system. (b) Discrete switching commands to suppress vibration. 21

7 Time responses of the camera orientation system. (a) Time response of the camera orientation device with vibration suppression input. (b) Time response of the camera orientation device with step input. . . . 23

8 The image frame of interest obtained for both inputs, as indicated in Figure 7. (a) An image obtained with the discrete switching commands. (b) An image obtained with the step command. 23

9 Time response of the system by step input 25

10 Dynamics-based Estimation of the PSF for the robotic camera orientation system. 27

11 Comparisons between the actual motion and the estimated motion. (a) Time response of the camera orientation system given the step command. (b) Time response of the camera orientation system given the discrete switching commands. 28

12 Sensitivity plots (a) Sensitivity plot for the frequency (b) Sensitivity plot for the damping coefficient 29

13 Experimental setup for image de-blurring 33

14 Oscillatory responses of the camera orientation system with various amplitudes. Images are obtained at 10.4fps. 36

15 Test set #1. The obtained blurry image and estimated latent images by various methods. Images are taken at 29fps with Vibration Suppresion input command. (a) - (g) are cropped local images. (a) Proposed Method (b) Goldstein *et al*. (c) Fergus *et al*. (d) Whyte *et al*. (e) Shan *et al*. (f) Xu *et al*. (g) Cropped obtained blurry image (h) Obtained blurry image 37

16 Test set #1. Estimated blur kernels. The size of the PSFs is 21 × 21. (a) Proposed Method (b) Goldstein *et al*. (c) Fergus *et al*. (d) Whyte *et al*. (e) Shan *et al*. (f) Xu *et al*. 37

17 Test set #2. The obtained blurry image and estimated latent images by various methods. Images are taken at 29fps with Step input. (a) - (g) are cropped local images. (a) Proposed Method (b) Goldstein *et al*. (c) Fergus *et al*. (d) Whyte *et al*. (e) Shan *et al*. (f) Xu *et al*. (g) Cropped obtained blurry image (h) Obtained blurry image 38

18 Test set #2. Estimated blur kernels. The size of the PSFs is 21 × 21. (a) Proposed Method (b) Goldstein *et al*. (c) Fergus *et al*. (d) Whyte *et al*. (e) Shan *et al*. (f) Xu *et al*. 38

19 Test set #3. The obtained blurry image and estimated latent images by various methods. Images are taken at 10.4fps with periodic oscillatory input. One PZT step was used to create oscillatory response. (a) - (g) are cropped local images. (a) Proposed Method (b) Goldstein *et al*. (c) Fergus *et al*. (d) Whyte *et al*. (e) Shan *et al*. (f) Xu *et al*. (g) Cropped obtained blurry image (h) Obtained blurry image 39

20 Test set #3. Estimated blur kernels. The size of the PSFs is 11 × 11. (a) Proposed Method (b) Goldstein *et al*. (c) Fergus *et al*. (d) Whyte *et al*. (e) Shan *et al*. (f) Xu *et al*. 39

21 Test set #4. The obtained blurry image and estimated latent images by various methods. Images are taken at 10.4fps with periodic oscillatory input. Two PZT steps were used to create oscillatory response. (a) - (g) are cropped local images. (a) Proposed Method (b) Goldstein *et al*. (c) Fergus *et al*. (d) Whyte *et al*. (e) Shan *et al*. (f) Xu *et al*. (g) Cropped obtained blurry image (h) Obtained blurry image 40

22 Test set #4. Estimated blur kernels. The size of the PSFs is 21 × 21. (a) Proposed Method (b) Goldstein *et al*. (c) Fergus *et al*. (d) Whyte *et al*. (e) Shan *et al*. (f) Xu *et al*. 40

23 Test set #6. The obtained blurry image and estimated latent images by various methods. Images are taken at 10.4fps with periodic oscillatory input. Four PZT steps were used to create oscillatory response. (a) - (g) are cropped local images. (a) Proposed Method (b) Goldstein *et al*. (c) Fergus *et al*. (d) Whyte *et al*. (e) Shan *et al*. (f) Xu *et al*. (g) Cropped obtained blurry image (h) Obtained blurry image 41

24 Test set #6. Estimated blur kernels. The size of the PSFs is 35 × 35. (a) Proposed Method (b) Goldstein *et al*. (c) Fergus *et al*. (d) Whyte *et al*. (e) Shan *et al*. (f) Xu *et al*. 41

25 Quantitative evaluation of the de-blurred results 42

26 Comparisons of computation times 44

27 Experimental results of the test sets #6 − #9. (a) Quantitative Evaluation. (b) Comparisons of computation times. 47

28 Comparisons of the de-blurring results. Test set #6: Figures 28(a) - 28(e). Test set #7: Figures 28(f) - 28(j). Test set #8: Figures 28(k) - 28(o). Test set #9: Figures 28(p) - 28(t). 48

29 Test set #6. Cropped local images from Figures 28(a) - 28(e). . . . 49

30 Test set #7. Cropped local images from Figures 28(f) - 28(j). . . . 50

31 Test set #8. Cropped local images from Figures 28(k) - 28(o). . . . 51

32 Test set #9. Cropped local images from Figures 28(p) - 28(t). . . . 52

33 Test set #10. The obtained blurry image and estimated latent images by various methods. Images are taken at 29fps with Vibration Suppresion input command. (a) - (g) are cropped local images. 53

34 Test set #10. The obtained blurry image and estimated latent images by various methods. Images are taken at 29fps with Vibration Suppresion input command. (a) - (g) are cropped local images. 54

35 Test set #10. Estimated blur kernels. The size of the PSFs is 21 × 21. (a) Proposed method (b) Goldstein *etal*. (c) Fergus *etal*. (d) Whyte *etal*. (e) Shan *etal*. (f) Xu *etal*. 54

36 Experimental setup for evaluation by OCR. 55

37 Input blurry images for evaluating the de-blurring methods by OCR. (a) 10 alphabet characters. (b) 10 numerical digits. (c) A sentence (100 characters). 56

38 Text recognition results from the de-blurred image. The input blurry image is Figure 37(a). 57

39 Evaluation of the de-blurring methods by OCR. (a) Number of recognized text. (b) Number of inaccurately recognized text. 58

40 Text recognition results from the de-blurred image. The input blurry image is Figure 37(b). 58

41 Evaluation of the de-blurring methods by OCR. (a) Number of recognized numerical digit. (b) Number of inaccurately recognized numerical digit. 59

42 De-blurred images of 37(c) by various methods. 59

43 Recognized texts by OCR from images shown in Figure 42. 60

44 Recognition rates by OCR from the de-blurred images containing a sentence. 60

45 Comparison of pipelines. (a) Pipeline of conventional de-blurring methods. Image acquisition, motion estimation, and kernel generation are processed in a serial manner. (b) Pipeline of the proposed de-blurring method. Image acquisition, motion estimation, and kernel generation are processed in a parallel manner. 63

46 Function blocks and data flows of the proposed system architecture for a general robotic system with the real-time image de-blurring capability. The FPGA is used to control a multi-actuator system and execute various functions in a parallel manner. The GPU is used to enhance process times of matrix computations. 66

47 Pinhole camera model for rotational motion 69

48 Comparison of data transfer times per pixel from CPU to GPU for various kernel dimensions 72

49 Comparison between Richardson-Lucy and Wiener deconvolution methods. The image was taken under low light condition to introduce noise and to test robustness. The results shows that the Richardson-Lucy method is robust to noise while the Wiener method is not. (a) Original blurry input (b) Result obtained by the Richardson-Lucy method (c) Result obtained by the Wiener method 74

50 Single channel approach when performing the deconvolution process on the GPU to improve computation time. 74

51 Improvement in the computation times. (a) Color images (b) Grayscale images 76

52 Visual-motor coordination (a) Vibration suppression input (b) Step input 79

53 Results of visual-motor coordination. The camera orientation device scans four different positions. The total travel distance is 16 PZT inputs. 80

54 Timeline to create a panoramic image from VGA images. The 'move' block represents the settling time for a point-to-point motion using the robotic vision system. The 'image' block represent the time of image acquisition. The 'de' block represents the computation time of image de-blurring that set to 16.5 ms based on the results from section 4.3. 81

55 Deconvolution results. The original method has artifacts around the edges due to zero padding. (a) Original blurry input. (b) Obtained by general deconvolution. (c) Obtained by modified deconvolution. . . . 83

56 An input image padded with flipped images to avoid ripple artifacts. 84

57 Experimental setup for generating a panoramic image while quickly scanning the environment by visual-motor coordination. 85

58 Panoramic results of the proposed method. (a) Dynamics-based method incorporating with real-time image de-blurring. (b) Dynamics-based method incorporating without image de-blurring. (c) Clean static image for reference. 86

59 Panoramic results of the compared methods. (a) MS ICE (b) PTGui (c) Panoweaver 87

60 Quantitative evaluation of the panorama results. 88

61 Transmissibility analysis of the camera orientation system. 90

62 Transmissibility analysis of the camera orientation system. 91

63 Responses of the entire system and camera due to the impact force. 92

64 Experimental setup. The camera orientation system is supported in a vertical manner. A rubber hammer is used to apply external disturbances to the camera orientation system. 93

65 Peak values of the displacements due to the impact force applied on the camera orientation system. 94

66 Experimental setup. The entire frame of the camera orientation system is firmly fixed. 95

67 Peak values of the displacements due to the impact force applied on the camera orientation system. 95

68 The cellular actuator exhibits quantization and flexibility. A total of 16 PZT actuators is nested in a multi-layer fashion and each PZT is operated in an ON-OFF manner. 99

69 The camera orientation system. It has a 1 DOF motion driven by an antagonistic pair of cellular actuators. 99

70 The cellular actuator produces discrete output positions due to the binary operation in the PZT actuators. 100

71 Angular displacements of the camera orientation system with respect to the number of individual PZTs activated. 100

72 Step Response of the camera orientation system. This study is interested in generation of motion with velocity slower than the early stage of step response. 102

73 Using a conventional feedback controller with PWM quantization, the camera orientation system can track a ramp signal that is 10 times slower than the maximum velocity in response to the step input. However, it causes high-frequency switching in the individual PZTs. . . . 103

74 Block diagrams. (a) Open-loop controller for a third-order system. (b) Open-loop controller for a second-order system with modified desired profile. 105

75 Example of pulse generation. The amplitude of the response can be adjusted by changing Δt. 106

76 Generation of discrete commands for velocity matching. The timings of primitive movements are determined to achieve the target positions indicated by blue circles at the time when the next primitive is generated. 108

77 Discrete switching commands for generating an arbitrary movement. The desired trajectory is the same as that in Figure 73. 110

78 Failure of command generation in achieving the target position. . . . 110

79 Conceptual depiction of the change in the reference position when the desired motion crosses the midpoint. 111

80 Comparison between with and without changing the reference. . . . 111

81 Experimental setup for measuring the responses. 113

82 Comparisons between the proposed discrete controller and the conventional PWM controller. (a) Test #1. Ramp signal. (b) Test #2. Half period of a triangle wave. (c) Test #3. Half period of a sinusoidal signal. (d) Test #4. Absolute period of a sinusoidal signal. 115

83 Representation of residual vibrations in the complex plane. The solid lines represent an example for an undamped system. The dashed lines represent an example for an underdamped system. 117

84 Possible combination of saccade-like movements and smooth pursuit when the desired velocity is constant. 120

85 Comparisons between the proposed method, consisting of both the tracking and vibration suppression controllers, and the tracking only controller. Tests #5 and #6 have the same desired motion with different image acquisition times. Tests #7 and #8 have the same desired motion with different image acquisition times. Tests #9 and #10 have the same desired motion with different image acquisition times. . . . 121

86 Experimental setup. The proposed discrete switching commands are given to the system for tracking the object of interest positioned by the linear stage. 123

87 Generated movements to track objects of interest. The discrete switching commands for vibration suppression were generated after the exposure window. (a) Tracking Test #1. One image is obtained. (b) Tracking Test #2. Two images are obtained 124

88 Tracking test #1. The exposure time was 35 ms. The images were obtained under various conditions for comparison. (a) Obtained image while tracking. (b) Object in motion with camera at rest. (c) Reference image. 125

89 Comparisons of the absolute of the gradient values. Gradients of the middle line of the images are depicted. 126

90 Tracking test #2. The exposure time was 80 ms. The system tracked two moving objects in a short period of time and two images were obtained. (a) 'Object 1' captured in 'Exposure Window 1'. (b) 'Object 1' captured when the camera was at rest. (c) Reference image of 'Object 1'. (d) 'Object 2' captured in 'Exposure Window 1'. (e) 'Object 2' captured when the camera was at rest. (f) Reference image of 'Object 2'. 127

91 Pinhole camera model for translational motion 134

92 Experimental setup to test the de-blurring method for translational motion. 136

93 Scene to test motion blur caused by translational motion. Two checkerboards are placed with different distances to the vision system. . . . 137

94 Image with depth information. 137

95 Obtained blurry images. 138

96 De-blurred images acquired using a spatially invariant kernel. The blur kernel size was 27×27. 138

97 De-blurred images acquired using modified blur kernels. The blur kernel size for the red checkerboard was 27 × 27. The blur kernel size for the blur checkerboard was 21 × 21. 138

98 A red checkerboard is used to evaluate the de-blurring methods. (a) clean image obtained at rest. (b) blurry image obtained in the course of motion 139

99 De-blurred images. (a) Proposed method (b) Goldstein *et al.* (c) Fergus *et al.* (d) Whyte *et al.* (e) Shan *et al.* (f) Xu *et al.* (g) Cho *et al.* . . 141

100 Number of white pixels on the checkerboard region. 142

101 Examples of image stitching by SIFT and RANSAC. (a) Matched feature points from a de-blurred image set. (b) Matched feature points from a blurry image set. 144

102 Examples of image stitching by SIFT and RANSAC. (a) Matched feature points from a de-blurred image set. (b) Matched feature points from a blurry image set. 145

103 Panoramic image generated by MS ICE given the de-blurred images. 146

104 Panoramic image generated by PTGui given the de-blurred images. . 146

105 Panoramic image generated by Panoweaver given the de-blurred images. 146

106 Experimental setup for panorama generation. 147

107 Panoramic results of the proposed method. (a) Dynamics-based method incorporating with real-time image de-blurring. (b) Clean static image for reference. 148

108 Panoramic results of the compared method. (a) MS ICE (b) PTGui (c) Panoweaver 149

109 GUI for the robotic vision system 151

110 State diagram representation of functions in the FPGAs. 152

111 State diagram representation of functions in the CPU. 153

112 The dynamics-based image processing methods are executed in a pipelining fashion. 153

113 GUI for operating two linear stages 154

114 State diagram representation of functions in the FPGAs for operating linear stages. 154

115 CAD model of a 2-DOF mechanism 155

116 Exploded view of the CAD model 156

SUMMARY

This thesis presents a mathematical framework and methods to coordinate motion control and image processing in a fast moving robotic system. The human eye has two representative movements, a saccade and smooth pursuit. Saccade is one of the fastest and most accurate movements in the human eye. However, the human visual system receives blurry environmental information due to a finite integration time when a saccade takes place. Smooth pursuit is another eye movement that continuously follows an object with relatively slow velocity.

This thesis presents dynamics-based image processing methods for a fast-moving robotic camera system, inspired by the observation of the physiological evidences. Real-time panoramic image stitching is presented with simultaneous motion de-blurring in a dynamic vision system, allowing for generic image sensors with a standard frame rate and significantly less computational load, and requiring no motion sensors. The proposed methods are based on the dynamic model of the robotic eye and movements induced by the control system while conventional methods analyze inherent image properties. In the neuromotor system, the movements are generated by muscles that are essentially quantized, compliant actuators. To reproduce smooth pursuit-like movements, an open-loop discrete switching controller accounts for dynamics is introduced to generate an arbitrary velocity profiles for a robotic eye system driven by quantized, compliant actuators, while avoiding high-frequency switching in individual motor units. The effectiveness of the dynamics-based methods and the discrete switching controller are presented and validated on a fast-moving robotic orientation system.

CHAPTER I

INTRODUCTION

1.1 Motivation

The human visual system is an important contributor to human perception providing a large part of environmental cognition. Because the eye has a limited field of view (FOV), the human visual system scans the environment or changes the region of interest (ROI) by changing the direction of the person's gaze. The human eye has a total of three degrees of freedom (DOF), each generated by antagonistic pairs of extraocular muscles. Therefore, the human gaze is controlled by a total of six extraocular muscles, including four recti muscles and two oblique muscles [13].

It is known that the human eyes have two representative movement patterns: saccade and smooth pursuit [24]. A saccade is known to be the fast and rapid eye movements, enabling quick scanning the field of view and high-resolution fovea at the region of interest. Several physiology studies have shown that the movement time is within 50 ms for the 10 degree saccade or 100 ms for the 30 degree saccade and the velocity is ranged 250 - 500 degree per second [40, 5, 7].

Saccades occur much more rapidly than proprioceptive, vestibular, or visual feedback that can be returned to the brain [91]. This indicates that saccades are performed in an open-loop manner and may be implemented as feedforward control based on the oculomotor dynamics. Inversely, motion control of the image acquisition device must be planned to ease this masking process.

Figure 1 shows a conceptual illustration of the saccade and saccadic eye movements captured by a high-speed camera. Due to a finite integration time in the visual system and rapid point-to-point motion, the visual system may perceive blurry information;

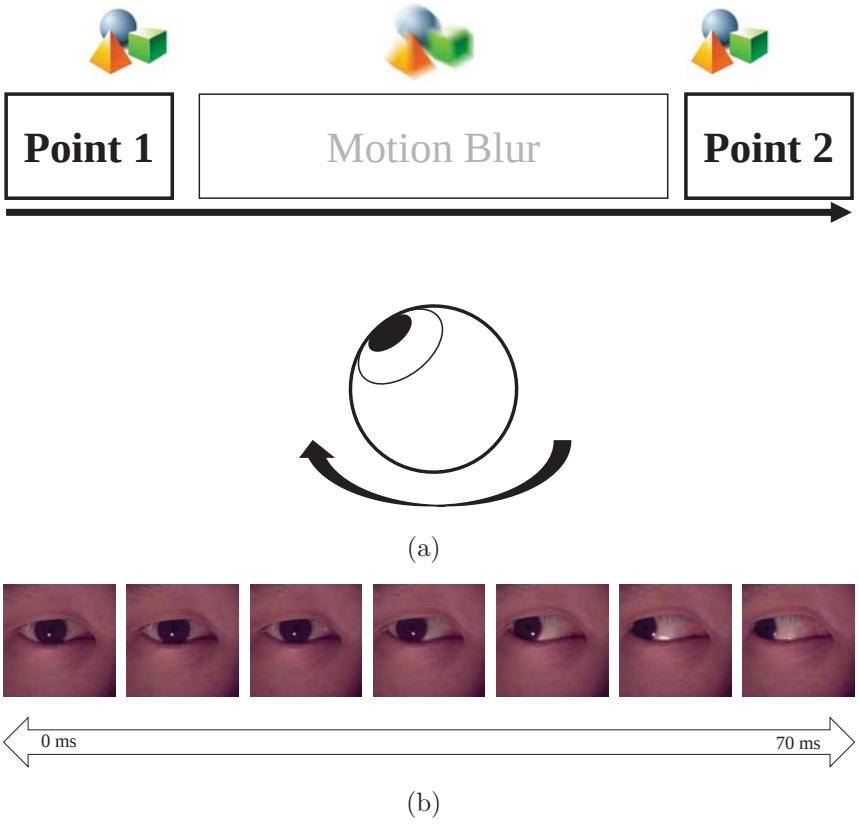


Figure 1: Saccades (a) Conceptual drawing of motion blur due to saccades (b) Eye movements captured by a high-speed camera.

the equivalent frame rate of the human eye is reported to be below 24 frame per second (fps) or 42ms/frame, which varies depending on situation [42].

Despite of saccadic eye motion, we are almost unaware of potentially blurry visual input. The neural system blocks images with substantial motion blur in the course of the eye motion, termed saccadic masking [46, 70]. Saccadic masking plays a key role when the motion of the eye is not negligible in visual perception.

Instead of saccadic masking, the visual information is recovered in the course of motion. Other Evidences indicate that this blurred information, called motion smear, is partially compensated by neural processes [8, 11, 17]. The human brain predicts or uses information on eye movements to reduce blurring [114, 92].

Smooth pursuit is another known eye movement. Unlike saccades, the eye continuously follows an object with relatively slow velocity. Smooth pursuit is the key component for tracking and takes place when no discontinuity exists in the position of the target [91].

Several camera positioning devices inspired by the human ocular system have been introduced [61, 109, 58]. Mechanisms of prior studies are actuated by various means and successfully generated a fast motion that is comparable with saccade [60, 119, 95, 97]. However, existing studies on camera positioning systems are limited to mechanical design and control, without image processing, although a vision sensor provides the richest information in various robotic applications. In addition, they have focused only on generation of saccade-like rapid movements. Since many prior studies have employed electromagnetic servomotors as a primary actuation system, it may be fairly easy to generate slowly moving profiles like smooth pursuit.

One might think that such velocity control of the eye is fairly easy in an engineering standpoint. However, this is not the case in the neuromotor system primarily due to recruitment, or discrete activation of muscle bundles [22, 47]. Muscles are essentially quantized actuators hence cannot produce continuous force. Despite of quantized force production, it has not yet been discovered this discrete actuation in the eye and able to perceive the features of a moving object. Although possible neuromotor mechanisms in smooth-pursuit are debatable, from an engineering point of view, this problem can be seen as velocity matching control of an impulsive dynamical system.

In a robotic vision system, a camera may receive blurry images when a saccade-like rapid motion takes place unless a high-speed camera system is employed [86, 110]. Blurry images are lack of substantial information in particular on the high frequencies which makes it difficult to analyze their properties or features. Therefore, motion of the robotic system is often limited by a frame rate of the vision sensors to avoid blurry images for post-processing. If rapid motion is necessary, the vision system may block

or abandon degraded images as if saccadic masking [70] plays a key role when the motion of the eye is not negligible in visual perception. An alternative approach would be to recover blurry images in real-time and make use of them, which could benefit performance or productivity.

1.2 Overview

This research aims to apply principles of the ocular physiology to the control of a robotic vision system. The concept behind this research is to merge the system dynamics area and image processing area for a robotic camera positioner while previous studies have focused solely on either mechanical design and control or image processing. Although detailed neural mechanisms behind saccadic masking and smooth-pursuit are not known, it is a natural consequence that motion control and image processing must be tightly synchronized. Image processing methods in coordination with inherently discrete and rapid ocular movements will reproduce saccades and smooth-pursuit in a robotic vision system.

This thesis will establish dynamics-based motion de-blurring for a camera orientation mechanism exhibiting quantization and compliance. The developed systems will be capable of recovering degraded blurry images and stitching them to create a panoramic image using a robotic vision system. The robotic vision system used in this study is able to generate rapid motion comparable with saccades. Image de-blurring will be accomplished by dynamics-based estimation of the blur kernel. A parallel system architecture will be designed and parallel computing will be exploited to achieve real-time processing of the image de-blurring algorithm. In general, image stitching is computationally easy when a set of *clean* static images is given. For a set of blurry images, however, image stitching is a extremely challenging task due to lack of high-frequency information. This problem will be solved by a dynamics-based homography estimation method using the robotic vision system.

To generate biologically inspired movements using the camera orientation system that is quantized and compliant, an open-loop discrete switching controller will be designed. It will generate an arbitrary velocity profile to reproduce smooth pursuit-like movements and track objects of interest. A feedback controller with PWM (Pulse-Width Modulation) method may be effective to generate desired tracking profiles. However, this causes high-frequency switching in individual motor units, which could lead to damage in flexible mechanisms. The proposed controller will be designed to generate velocity profiles for tracking an object of interest only during exposure windows of a camera. Additional discrete switching commands will be generated to suppress residual vibrations after acquiring images, which contributes to reduce the number of switching. To generate a saccade-like motion, existing minimum switching discrete switching vibration suppression (MSDSVS) controller will be used [96].

1.3 Dynamics-based Visual Processing

Eye movements are primarily self-induced, i.e., generated by the extraocular muscles receiving commands from the oculomotor system. It may be hypothesized that there is a neural mechanism that makes a prediction of such self-induced eye movements based on the dynamic model of the eye, enabling effective image de-blurring and panoramic image stitching, or dynamics-based visual processing.

Physiological evidences indicate visual processing and motor control are tightly coordinated in the human visual system. It has been reported that visual stimuli even before the saccade is masked, which indicates saccades are planned by the central system. [46]. Others have reported that the human brain predicts or uses information on eye movements to reduce blurring when saccades take place [114, 92]. The suppression of motion smear also occurs during smooth pursuit [8].

This possible mechanism is the key enabler of real-time panoramic image stitching with simultaneous motion de-blurring in a robotic vision system, allowing for generic

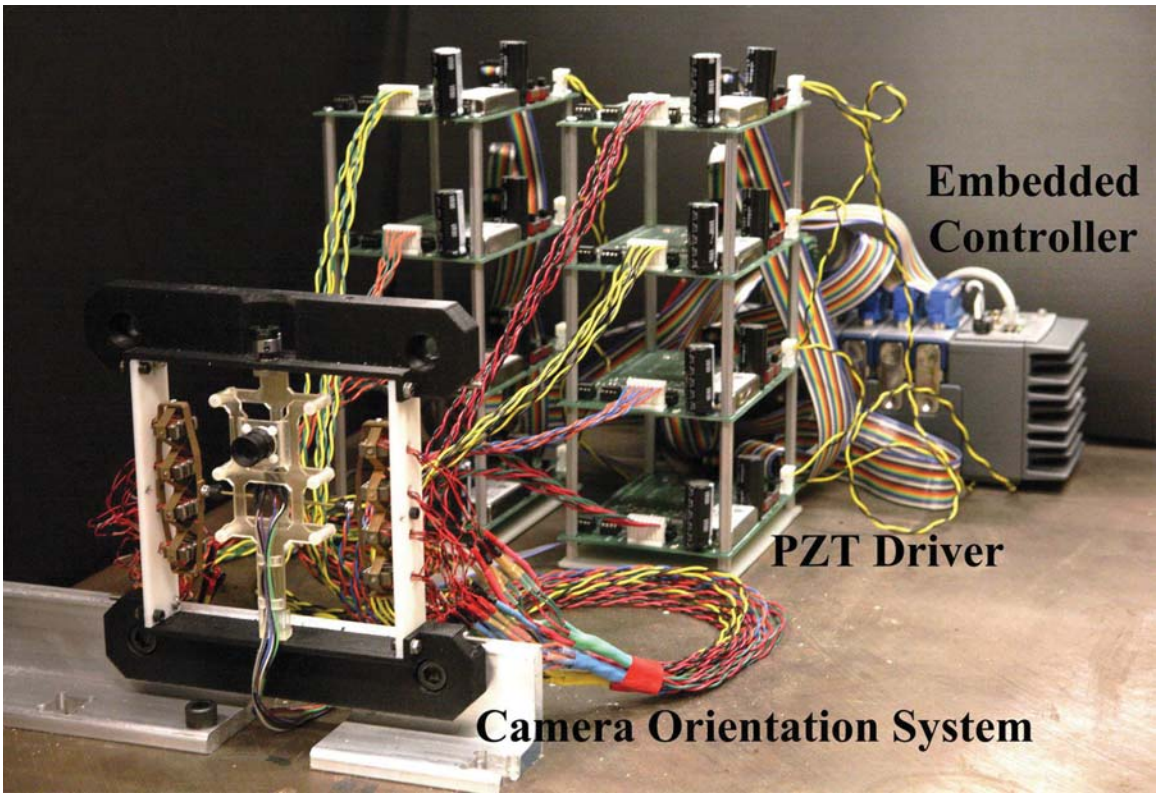


Figure 2: Robotic vision system

image sensors with a standard frame rate and significantly less computational load, and requiring no motion sensors. Similarly, the dynamic model allows for inverse calculation of a smooth velocity profile.

1.4 Testbed Device: Camera Orientation System

The methods and algorithms developed in this study will be tested and validated on the robotic vision system shown in Figure 2.

1.4.1 Mechanical Design

The camera orientation system inspired by the human ocular system has a single DOF (degree of freedom) and is orientated by an antagonistic pair of cellular actuators located on both sides as shown in Figure 3 [98]. The moving platen is connected to a rod that transmits the force from the antagonistic pair of the cellular actuators.

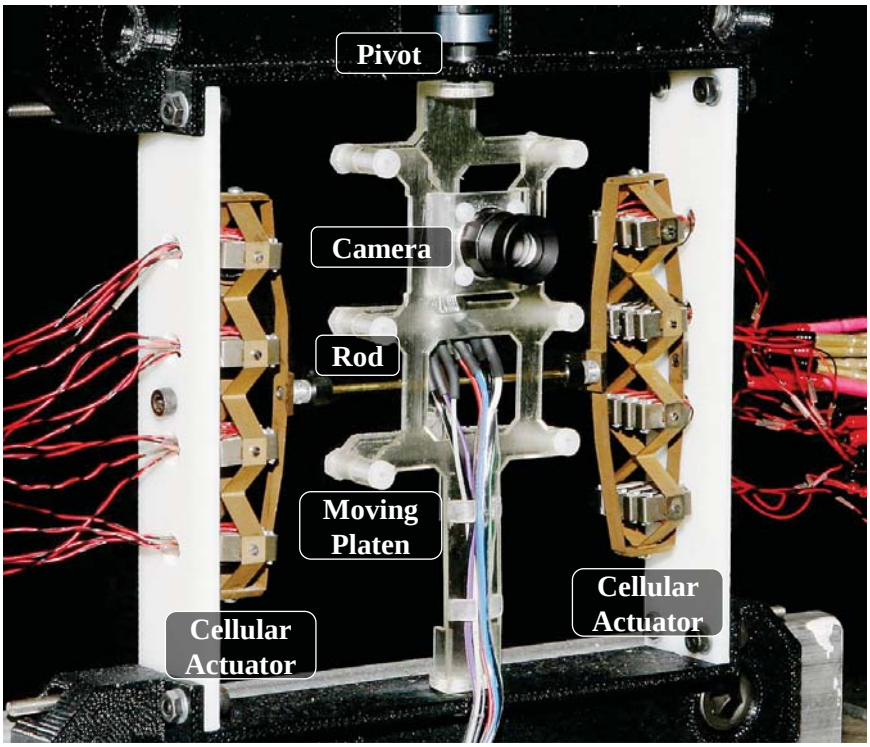
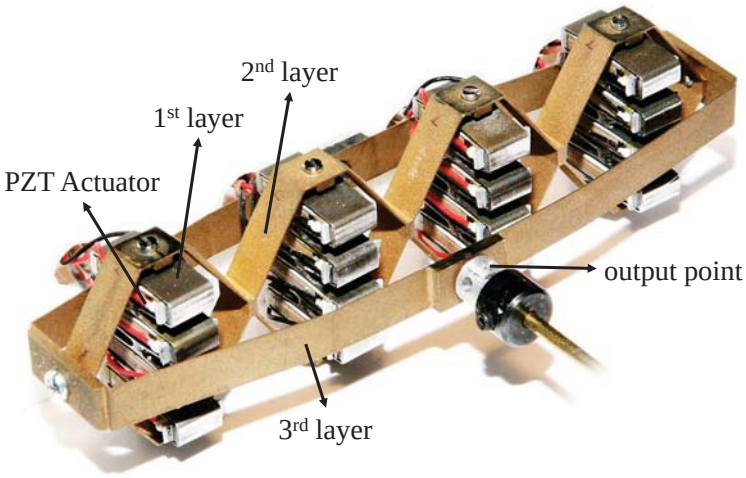


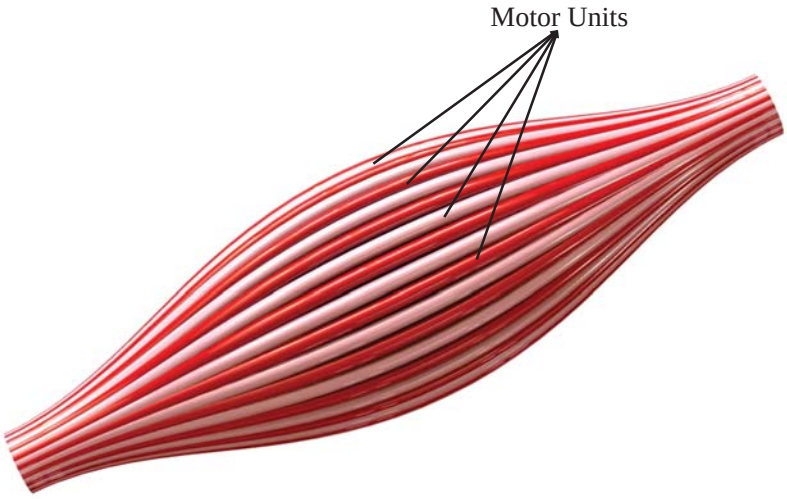
Figure 3: Piezoelectrically driven camera positioner system. The orientation of the camera mounted on the moving platen is controlled by a total of 32 PZT actuators. A tilting motion is generated when the rod is positioned by the antagonistic pair of the cellular actuators.

Each cellular actuator consists of 16 Lead Zirconate Titanate (PZT) stack actuators with deformable amplification mechanisms [1, 116]. Since no gears or sliding mechanisms are used in the structures, this mechanism exhibits zero backlash and noise-less operation while extremely fast movement is achieved. The axis of the rod is positioned perpendicular but not orthogonal to the pivot axis. Therefore, when the rod is pushed or pulled by the antagonistic pair of the cellular actuators, a moment is applied to the moving platen, resulting in a tilting motion of the camera positioning system.

The robotic vision system employ two different cameras in this research: Ximea subminiature camera MU9PC-MH and Logitech C270 HD webcam. The plastic casings of the cameras are removed and only a board-level camera is installed on a moving platen. The rubber shielding of a USB cable is also removed to avoid introducing



(a)



(b)

Figure 4: Motor unit recruitment. Each motor unit is operated in a binary manner and the action is generated by recruitment of individual units activated. (a) The cellular actuator consists of a total of 16 PZT actuators. A total of 16 PZT actuators is nested in a multi-layer fashion and the thin layers allow amplification. (b) The muscle consists of individual motor units.

mechanical resistance to the camera orientation system.

1.4.2 Quantization and Compliance in Actuation

A total of 32 APA50XS (Cedrat corporation) PZT actuators are used to position the camera. The cellular actuator can be controlled continuously by adjusting the

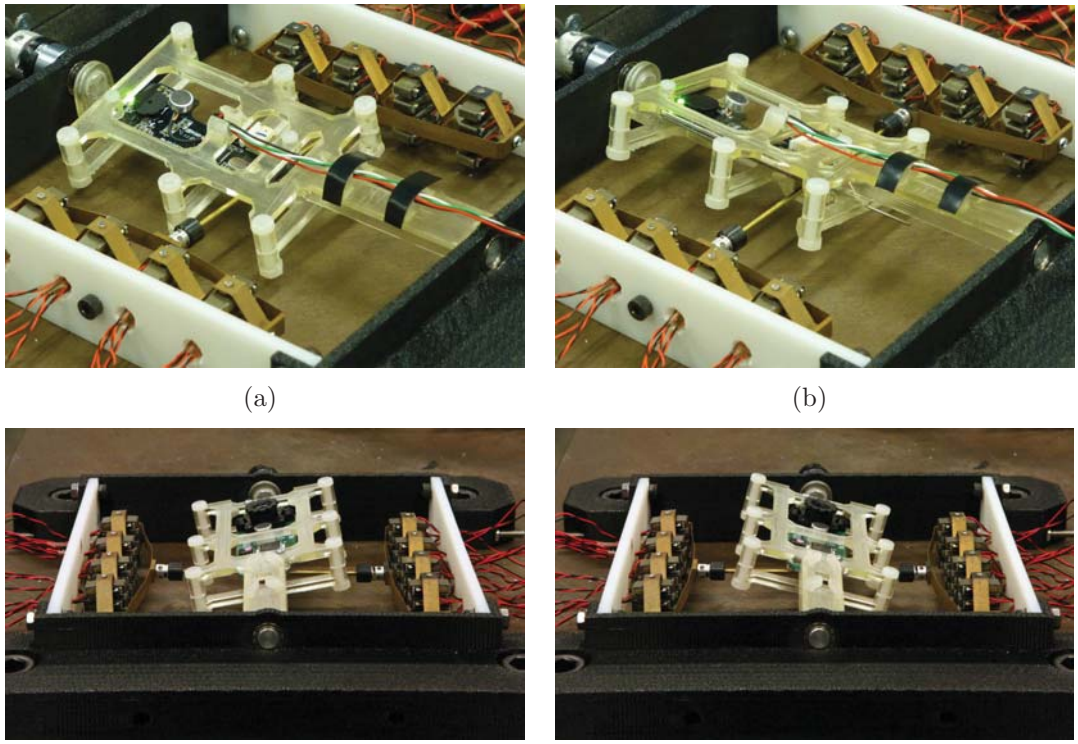
input voltage from 0V to 150V. The PZT actuator has a hysteresis response requiring additional sensors to estimate the position or to control the system. In this study, each PZT actuator is operated in an on-off manner to avoid hysteresis. In this way, the actuation system exhibits quantization.

A PZT actuator has an extremely fast response and high accuracy on the nanometer scale with a low energy consumption. However, its small strain and small stroke distance limit the application to robotics. To overcome these limitations, the PZT actuators are nested in a multi-layer structure as shown in Figure 4(a). The multi-layered strain amplification structure creates a larger displacement at the output point when forces are generated by the PZT stacks at the input [98]. The thin layers in the mechanism introduce flexibility and allow amplification at the output point.

These approaches share principles with the biological muscle system. Muscles are quantized system consisting of individual motor units activated in a binary manner as shown in Figure 4(b): each motor unit is either 'On' for contraction or 'Off' for relaxation [69, 59]. The cellular actuators and constituent PZT actuators mimic human muscle physiology where whole muscle activity is a summation of motor unit activity, respectively [97]. The binary operation in the individual PZT actuators is similar to physiological bang-bang type of inputs given in the human saccadic eye movement [26].

1.4.3 Range of Motion

The range of motion of the camera orientation system is ± 16 PZT inputs or a tilting angle of approximately ± 13 degrees. Since each cellular actuator has a total of 16 PZT actuators, the camera positioner can create 32 discrete angles. The discretized desired angle can be achieved by defining the number of activated PZT actuators. Figure 5 shows the operation range of the camera orientation system.



(a) (b)

(c) (d)

Figure 5: Single-DOF motion of the camera orientation system: (a) camera at extreme left (b) camera at extreme right (c) camera at extreme left (d) camera at extreme right

CHAPTER II

LITERATURE SURVEY

This chapter summarizes the contributions from prior studies that provide principles relevant to the work conducted in this study. To apply principles of the human ocular system, contributions of this study are related to image processing for images obtained in the course of motion of a quantized, complaint actuation system. A review of robotic vision systems is conducted with respect to motion control and visual processing areas. Part of the survey includes resources on control methods for a quantized and flexible actuation system. In addition, representative studies from motion de-blurring and generation of panoramic images are included with discussions on image processing in the human ocular system.

2.1 Camera Positioning Systems

There have been various approaches using compliant actuators or soft materials to design camera positioning systems, which are inspired by the human ocular system. A pan-tilt orienting mechanism using flexible beams is designed by Lee *et al*. [58]. A camera is connected to thin elastic-plastics beams driven by servo motors. Robotic eyes using cable-driven mechanisms connected to servo motors are proposed by Cannata *et al*. and Schulz *et al*. [12, 100]. A robotic eye that emulated vestibulo-ocular reflex (VOR) using Pneumatic Artificial Muscles (PAMs) is developed by Lenz *et al*. [60]. Although soft actuators are not used, a 3 DOF camera-orientation system driven by piezoelectric actuators via rigid push rods is designed by Villgrattner and Ulbrich [118]. However, actuation systems in the existing studies have little in common with

biological muscle actuation because traditional servo motors have a marked discrepancy from contractile, compliant extraocular muscles. Although forces were transmitted through flexible mechanisms in some studies, their actuation systems still have discrepancy from the biological muscle system whose action is generated by motor recruitment. In addition, mechanisms in prior studies [58, 109, 119, 60] are controlled in a closed-loop manner, whereas it is reported that the saccade is completed in an open-loop manner [48, 124, 91]. Schultz and Ueda developed a single DOF camera positioner oriented by an antagonistic pair of quantized, compliant actuators that are controlled in an open-loop manner [98, 97].

These robotic eyes have proposed notable mechanisms and generated a fast motion that is comparable to saccades. However, smooth pursuit, which is the other representative mode of operation in the human ocular system, has not been addressed. This is because an electromagnetic servomotor is employed in most of the studies [58, 109, 119, 60, 12, 100]. In addition, prior studies on robotic camera positioners are limited to mechanical design and control. Image processing was not studied.

2.2 Control of a Quantized or Complaint Actuation System

In contrast to the conventional servomotor systems, the biological system is known to be controlled in an impulsive manner. Observations indicate that its compliant actuation system is operated in a binary manner. To control such a system, from an engineering point of view, an open-loop discrete switching controller that accounts for flexible mechanisms is required.

2.2.1 Compliant Actuation System

Compliant actuators have shown potentials in various applications [37, 56, 115]. Soft actuators posses safety, high energy efficiency, zero backlash, and power density [37, 102, 57], and hence they have been studied for various applications such as human-robot interaction (HRI) system, and medical and biomimetic robotic systems [20,

68, 75, 115]. However, the control of soft actuators may be challenging because they are inherently flexible and have a physical configuration different from that of stiff actuation system [108, 80]. In fact, stiff actuators are preferred in traditional applications for precision and repeatability. While notable actuators fabricated using soft materials have been introduced in studies, most of these studies lack results on precise motion control. Some studies utilized sliding mode control, but focused only on point-to-point movements. In addition, their controllers used feedback information which has a little common with the biological system. A closed-loop controller may be effective to generate an arbitrary velocity profile, but may cause high-frequency switching in the actuation system used in this study which could possibly damage the flexible mechanism.

Several command shaping methods have been proposed to suppress residual vibrations in a flexible system in an open-loop manner [43, 107]. A set of input commands are given to the system with different timings that are determined by the natural frequencies of the system. The commands induce residual vibrations to the system with different phases to suppress the resonant modes. However, prior studies [43, 107] have focused on point-to-point movements; there has been little discussions of generating arbitrary velocity profiles. An arbitrary velocity profile may be achieved by closing the control loop, but it may damage the compliant actuation mechanism due to high-frequency switching in the individual control units. In addition, shaping commands are generated for a system employing servo motors, which are suitable for generating continuously variable movements. Thus, existing approaches [43, 107] for controlling a flexible mechanism are not applicable to the quantized, compliant mechanism used in this study.

2.2.2 Quantized Actuation System

Various efforts have been made to control a quantized actuation system. A discrete switching vibration suppression (DSVS) controller is proposed by Schultz and Ueda to suppress residual vibrations for a quantized system [96]. An algorithm consisting of iterative nested loops is developed to determined commands from a set of nonlinear equations. Katoch and Ueda investigate an evolutionary optimization approach to control shape memory alloy (SMA)-driven, quantized actuator system [49]. A non-uniform array distribution with motor variability is determined by an floating point quantization (FPQ) schema. Richter *et al*. investigate the absolute stability of a discrete linear system with a full-state quantized feedback controller using a discrete positive-realness (DPR) [90]. Although these quantized controllers have shown notable results for generating point-to-point movements, smooth pursuit-like movements cannot be achieved for the camera orientation system to track an object of interest.

A discrete switching controller to achieve an arbitrary trajectory have been studied. Ueda *et al*. propose an FPQ method demonstrating variability using complaint, quantized actuator arrays [117]. This study shows a quantized control can replace the signal dependent noise (SDN) approach for replacing motor noise reported in neuromotor systems. Schultz and Ueda propose a controller that can continuously adjust the outputs for a quantized, flexible system [99]. A candidate control function is chosen from a PWM signal and timing of switching commands is modified by an inter-sampling strategy. These approaches demonstrate promising results but the controller include a feedback loop and its implementation is not addressed.

2.3 Image De-blurring

2.3.1 Image-based Approaches

A camera system perceives blurry information when rapid motion takes place during the exposure window due to a finite integration time. There are various approaches in the computer vision field to restore blurry degraded images due to motion blur. To restore an image that is degraded by motion blur, the motion path must be estimated. The estimated path can be represented as a point spread function (PSF). The PSF is a blur kernel that describes the camera motion during an exposure window. If the motion blur is shift invariant, an un-blurred or latent image can be estimated with a deconvolution algorithm, given the estimated PSF and blurry image. Some studies propose the estimation of the blur kernel from a single image [27, 44, 105], while other studies use multiple images for blur kernel estimation [63, 88].

Fergus *et al*. propose a method to estimate the blur kernel by maximizing the probability in the image gradient domain [27]. Using the mixture of zero-mean Gaussians as the prior and given the measured image gradient, the posterior distribution can be defined. The method approximates the full posterior distribution and then estimates the blur kernel with maximum marginal probability, which is the maximum a-posteriori (MAP) solution. The approximation is made by a variation method, proposed by Miskin and MacKay [74]. The cost function, which is the difference between the approximating distribution and the true posterior, is minimized iteratively by the variation method.

Shan *et al*. observe that the majority of ringing artifacts in the latent image are caused by image noise and errors in kernel estimation [105]. Hence, they propose a unified probabilistic model for both blur kernel estimation and latent image estimation to avoid ringing artifacts. Both estimations are unified into a single MAP problem.

Xu and Jia observe that strong edges do not necessarily benefit kernel estimation, but could deteriorate kernel estimation under certain conditions. They propose a

two-phase blur kernel estimation to avoid this issue [121]. In the first phase, critical edges are defined by using Gaussian filtering and solving the shock filtering PDF problem. Then, the cost function, which is a function of noise, spatial prior, and Gaussian regularization weight, is optimized to find a coarse blur kernel. In the second phase, a refined blur kernel is defined by iterative support detection (ISD) method to minimize a non-convex cost function.

Goldstein and Fattal propose a method to estimate a blur kernel from the irregularities in the statistics of blurry images [33]. A power-law model is introduced to describe the power spectrum of original images. After several processes such as filtering, approximation and whitening are performed, the one-dimensional auto-correlation function of the differentiated projections of the blurry image is computed. Then, the blur kernel is iteratively estimated by a phase-retrieval algorithm given the minimized full 2D power spectrum function.

Whyte *et al*. propose an estimation of a non-uniform blur kernel that accounts for camera motion [120]. The non-uniform blur kernel includes a geometric model of camera motions: translation and rotation. Then, the algorithm proposed by Fergus *et al*. [27] is adopted and the algorithm proposed by Miskin and MacKay [74] is used to approximate the posterior distributions. A cost function is optimized by optimization of the Kullback-Leibler (KL) divergence between the posterior and the approximating distribution [10]. Since this method assumes that the rotational motion is a dominant factor in motion blur.

Cho and Lee propose a de-blurring method that estimates both latent image and kernel in an iterative manner. A kernel is estimated by searching image derivatives to predict strong edges. Therefore, the estimation process is dependent on local features to enhance processing times. This method assumes that a latent image contains strong edges to estimate a kernel. After a kernel is estimated, an input image is estimated by a deconvolution process. The de-blurred image is given to the kernel

estimation process again to refine the estimated kernels, and this process is repeated.

These image-based approaches successfully estimate the latent image; however much computation time is required [27, 44, 105, 106, 4, 33, 18, 120]. It takes at least 10 seconds to de-blur a VGA-size image [55], which is not preferable for a robotic vision system. In particular, the computation time greatly increases as the image size increases or as the blur kernel size increases for image-based methods. To avoid this problem, part of the image could be analyzed to estimate the PSF, but it may be inaccurate. The key parameters such as the kernel dimension need to be manually determined by users, which is not suitable for a fast-scanning robotic vision system. Thus, prior studies are limited to a hand-held camera system. In addition, the image-based approaches generally require an image having strong contrast between the foreground and background [44, 104].

2.3.2 Sensor-based Approaches

There are different approaches that do not analyze inherent properties of a blurry image but use external devices to estimate the motion blur. One study use an additional high-speed camera with a low resolution to track camera motion [9]. However, this approach also requires much computation time for post-image processing with the secondary camera. Another study use a gyro sensor and an accelerometer to track camera shake or motion [45]. However, it involves computations that estimate the path of a camera shake from noisy gyro and accelerometer signals.

Apparatus-based methods are also proposed [36, 87, 82]. Raskar *et al*. propose a fluttered shutter approach for controlling the exposure time and preserving high-frequency information. Park *et al*. extend this coded exposure technique [87] to image streams for a mobile system [82]. However, the coded exposure inevitably lowers the frame rate and image brightness. In addition, this fluttering does not occur in the human vision system and is not applicable to a standard camera.

2.4 Generation of Panoramic View

The field of view of a generic camera is smaller than that of a human [84]. Therefore, so there has been a major interest in panoramic image generation. A panoramic image is generated by quick scanning the environment using the fast-moving camera orientation system.

To relate neighbor images to each other, a homography matrix should be obtained to warp images on the same plane. In general, existing stitching methods successfully produce notable results by first extracting feature points by Scale-invariant Feature Transform (SIFT) or other feature detection algorithms [66, 39]. Some studies utilize Speeded Up Robust Features (SURF) to enhance the computational speed [6]. After the feature points are extracted, they are matched by Random Sample Consensus (RANSAC) [28] or other means [31]. Additional processes can be performed such as color adjustment to refine the final images.

Lopez *et al*. use a mobile phone to generate a panoramic image where the obtained images may be blurry [64]. However, degraded blurry images are simply abandoned. Yang *et al*. show promising results by incorporating inertial sensors embedded in a mobile phone. However, the degree of blurriness is required to be minor, and the resultant images still exhibit artifacts [123].

Existing methods for panoramic image generation are developed for clean images exhibiting no motion blur. Therefore, existing methods are not robust to blurry images due to a lack of information on high frequencies, while, in this study, images are obtained in the course of rapid motion.

In contrast with software-based approaches, camera platforms specialized in panoramic image generation are introduced. Various multi-camera systems are developed to generate a wider FOV than a single camera [41, 85]. A motorized stage is utilized to slowly scan the environment using a generic camera [32]. These platforms also require clean images to generate panoramic image. In addition, they are limited to a

single function and the motorized systems are simply used to position the camera at different angles and obtain images at rest, while this study aims to reproduce two representative modes of the human ocular system. An omnidirectional camera that consists of a single sensor and a mirror system can achieve a 360-degree FOV in the horizontal plane, but the entire image is distroted [72].

2.5 Human Ocular System

The human eye represents a good platform for demonstrating engineering challenges inherent in the neuromotor system. Physiological evidences indicate that the visual perception in the brain is tightly synchronized to eye movements [114, 92, 46]. In the biological system, it is known from various neuroscience studies that 50% - 80% of the human brain's processing power is dedicated to visual perception [30]. The human brain processes the vast visual information in a parallel manner [93] and it takes at least 100 ms - 150 ms to process a complex natural image [113, 23]. Similarly in robotic studies, image processing requires a large amount of computational effort to process images. In particular, de-blurring algorithms that rely on inherent information are known to be computationally expensive.

Little engineering study has been made on the oculomotor control for fast dynamic vision. Inspired by the observation of the human visual system, the use of dynamics information may benefit the computation time in a fast moving robotic system. In addition, it become possible to accelerate computation times for complex or heavy algorithms by means of parallel computing such as GPU (Graphics Processing Unit) or FPGA (Field Programmable Gate Array) as reported in various studies [2, 19]. Therefore, in this study, these aspects will be addressed and utilized for the camera orientation system to achieve real-time performance.

CHAPTER III

DYNAMICS-BASED IMAGE DE-BLURRING

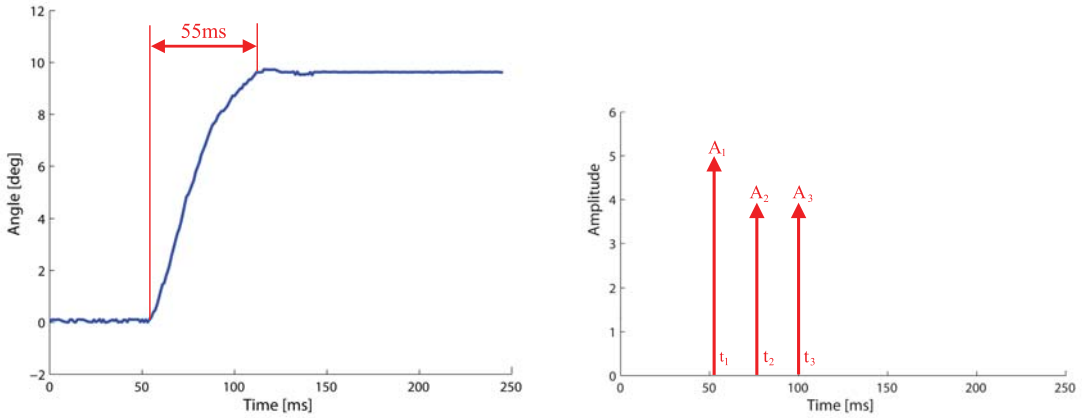
This chapter presents a method for removing motion blur from images captured by a fast-moving robot eye. Existing image techniques focused on recovering blurry images due to camera shake with long exposure time. In addition, previous studies relied solely on properties of the images or used external sensors to estimate a blur kernel, or point spread function (PSF).

This study focuses on estimating a latent image from the blur images taken by the robotic camera orientation system. A PZT-driven, compliant camera orientation system was employed to demonstrate the effectiveness of this approach. Discrete switching commands were given to the robotic system to create a rapid point-to-point motion while suppressing the vibration with a faster response. The blurry images were obtained when the robotic system created a rapid point-to-point motion, like human saccadic motion.

This chater presents proposes a method for estimating the PSF in knowledge of system dynamics and input commands, resulting in a faster estimation. The proposed method will be investigated under various motion conditions using the single degree-of-freedom camera orientation system to verify the effectiveness and be compared to other approaches quantitatively and qualitatively.

3.1 Saccade-like Motion in a Robotic Eye

Human saccadic eye motions are rapid eye movements within 50 ms settling time at the maximum angular velocity of 250 - 500 deg/sec. In order for the camera orientation system to generate the saccade-like motion, discrete switching commands



(a) (b)

Figure 6: A saccade-like motion with an open loop vibration suppression input. (a) Time response of a rapid point-to-point motion of the camera orientation system. (b) Discrete switching commands to suppress vibration.

must be given. The quantized discrete input commands can be determined by understanding the mechanical properties of the actuation system such as the natural frequency and the damping ratio. Figure 6(b) shows the discrete switching commands given to the system at appropriates time with appropriate amplitudes by applying a phase-vector analysis of mechanism control [96]. Figure 6(a) shows the time response of the camera orientation system as a result of the discretized switching commands. The results show that the vibration has been successfully suppressed and the settling time is within 55ms. If inappropriate commands or a step command are given to the system, however, the time response shows considerable oscillation resulting in a longer settling time.

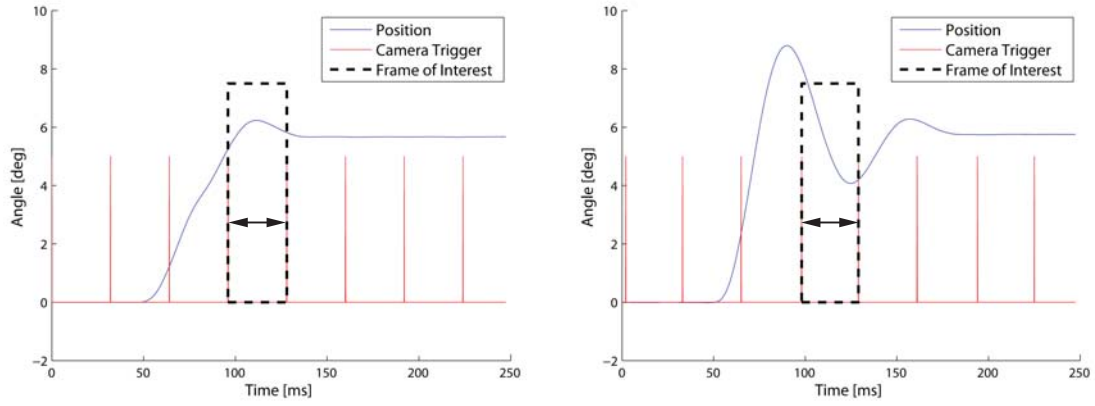
To apply principles of the human ocular system, the robotic vision system employed a camera whose frame rate was set to 30 fps that is comparable with the equivalent frame rate of the human eye. Table 1 demonstrates comparison of motion and visual systems between the biological eye and the camera orientation system used in this study. It can be checked that both specifications are comparable and the camera may perceive blurry images in the course of rapid motion.

Table 1: Comparison between the biological system and the robotic vision system

	Biological Eye	Robotic Vision System
Settling Time of Point-to-Point Movements	50 - 100 ms	55 - 65 ms
Velocity	250 - 500 deg/sec	200 - 350 deg/sec
Frame Rate	24 fps (equivalent)	30 fps

3.1.1 Discrete Switching Input and Step Input

Figures 7(a) and 7(b) show the time responses of the camera orientation system given the vibration suppression input and the step input, respectively. When saccade takes place, the brain blocks images with substantial motion blur in the course of eye motion which is called saccadic masking [70]. Therefore, the image frame of interest is the image obtained in the settling region. The image of interest from the time response with vibration suppression input is the third frame after the commands are given, as shown in Figure 7(a). For comparison, the same image frame was selected from the time response with step input, or vibration uncompensated input, as shown in Figure 7(b). The system starts obtaining images without motion blur after the fourth frame is given the vibration suppression commands, and the system starts obtaining images without motion blur after the sixth frame is given the step input. It can be easily seen that the camera orientation system experiences less vibration and has a faster settling time given the vibration suppression commands. It can also be seen from the obtained images that the system will receive less motion blur given the vibration suppression commands as shown in Figure 8.



(a) (b)

Figure 7: Time responses of the camera orientation system. (a) Time response of the camera orientation device with vibration suppression input. (b) Time response of the camera orientation device with step input.

(a) (b)

Figure 8: The image frame of interest obtained for both inputs, as indicated in Figure 7. (a) An image obtained with the discrete switching commands. (b) An image obtained with the step command.

3.2 Image De-blurring

The blur kernel, or the PSF, represents a trajectory of the motion. In order to estimate the latent image accurately, the blur kernel must be estimated accurately. This study developed a method for the estimating of PSFs using the system dynamics of the camera orientation system.

3.2.1 Dynamics-based Blur Kernel Estimation

In the absence of external disturbances, the motion of a robotic camera orientation system is predictable since the dynamics of the system and input commands are known and given. Therefore, the motion of the system is estimated in an open-loop manner without the use of additional sensors. This approach is inspired by evidence that a saccade is completed in an open-loop manner [91] and the human brain predicts or uses information of eye movements to reduce motion smear [92, 114].

The state-space representation of a m-DOF, n-th order, linear system can be written as:

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \sum_{i=0}^{N-1}[\mathbf{B}\mathbf{u}_i(t - t_i)] \tag{1}$$

$$\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) \tag{2}$$

$$\mathbf{u}_i(t - t_i) = A_i \cdot 1(t - t_i) \quad (i \in 0, \cdots, N) \tag{3}$$

$$t_0 = 0 \tag{4}$$

$$\mathbf{x}(t) \in \mathbb{R}^{n\times 1}$$

$$\mathbf{u}(t), \mathbf{y}(t) \in \mathbb{R}^{m\times 1}$$

$$\mathbf{A} \in \mathbb{R}^{n\times n}, \mathbf{B} \in \mathbb{R}^{n\times m}, \mathbf{C} \in \mathbb{R}^{m\times n}$$

where $\mathbf{x}(t)$ is the state vector, $\mathbf{u}(t)$ is the control input vector, $\mathbf{y}(t)$ is the output vector, $\mathbf{A}$ is the system matrix, $\mathbf{B}$ is the input matrix, $\mathbf{C}$ is the ouput matrix, $1(t)$ is

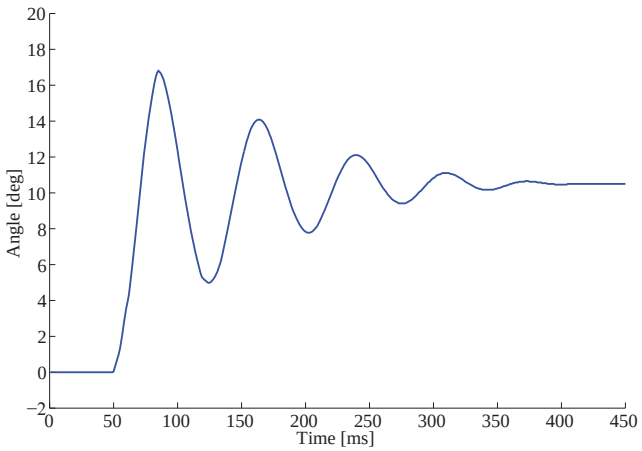


Figure 9: Time response of the system by step input

the unit step function, A_i is the amplitude of i-th step input, and N is the number of inputs given to the system.

Therefore, given the dynamics and control information, the motion of the camera orientation system can be estimated from (2). The linearity is assumed for simplicity in this study, but nonlinear dynamics can also be used since the dynamics calculation is open-loop.

The camera orientation mechanism, employed in this study, exhibits a single dominant frequency at 14.4 Hz with a damping ratio of 0.19. However, the dominant frequency was different at 16.1Hz when the desired distance was full stroke. The system parameters were identified experimentally by the observation of a step response as shown in Figure 9. Although nonlinearity was observed, an input of full stroke is not the range of the motion of interest in this study. Therefore, the system can be represented as a linear second-order system given as:

$$G(s) = \frac{K}{s^2 + 2\zeta\omega_n s + \omega_n^2} \tag{5}$$

where K is the residue, ω_n is the natural frequency, and ζ is the damping coefficient. The value of K is 6650.3.

Since the robotic camera orientation mechanism has a single dominant natural

frequency, it can be modeled as a linear, time-invariant (LTI), second-order, underdamped system. Then, the time response of the single DOF camera orientation system can be given as:

$$\theta(t) = \sum_{i=1}^{j} \frac{A_i}{\omega_n^2}[1 - e^{-\zeta\omega_n(t-t_i)} \cdot (\cos\omega_d(t-t_i) + \frac{\zeta}{\sqrt{1-\zeta^2}} \cdot \sin\omega_d(t-t_i))] \cdot 1(t-t_i) \quad (6)$$

where A_i is the amplitude of the i-th step input, t_i is the time of the given i-th step input, j is the number of amplitudes given to the system, ω_n is the natural frequency, and ζ is the damping coefficient.

The camera orientation system in this study has a single DOF and thus the output vector is:

$$\mathbf{y}(t) = \theta(t)\hat{x} \quad (7)$$

where $\hat{x}$ is a unit vector in the horizontal axis.

A PSF can be given as

$$k_{(x,y)} = h \quad (8)$$

where h is an energy level.

The pixel of the kernel (x, y) is

$$(x, y) = L \int_{t_a}^{t_b} \theta(t)dt \quad (9)$$

where L is a conversion factor dependent on the size of the blur kernel, and the energy function is

$$h = \frac{t_b - t_a}{t_{acq}} \quad (10)$$

where t_{acq} is the acquisition time of a single image.

Thus, the energy level at the location (x, y) of the PSF pixel is proportional to the time remaining at the location (x, y). The PSF is an energy distribution function for which the energy conservation constraint must hold:

$$\iint k_{(x,y)}dxdy = 1 \quad (11)$$

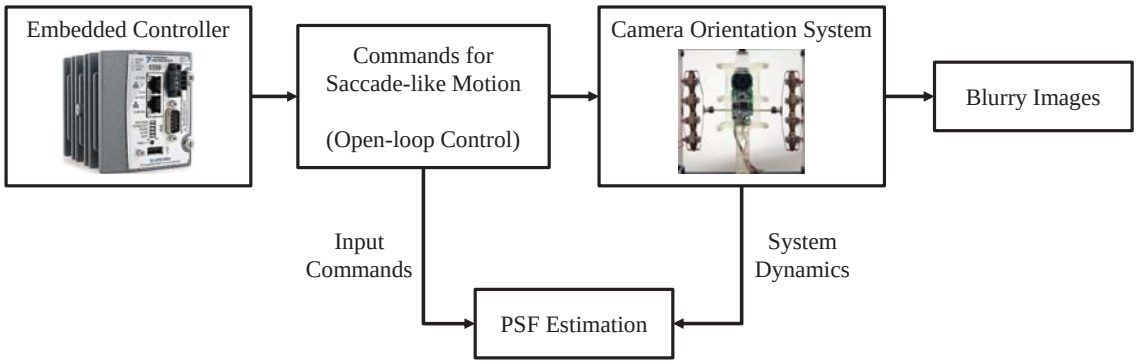


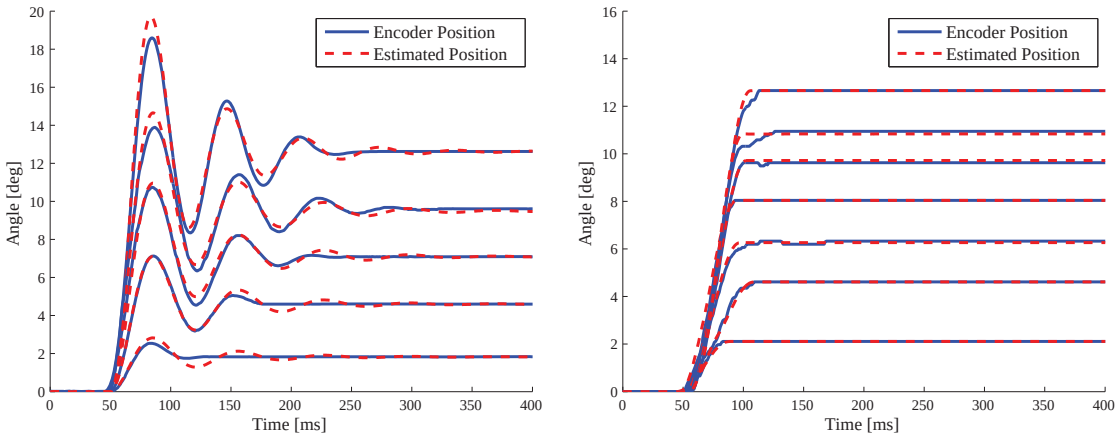
Figure 10: Dynamics-based Estimation of the PSF for the robotic camera orientation system.

3.2.2 Implementation of Dynamics-based Estimation

The proposed method requires an accurate estimation of the motion given the system dynamics and control commands to estimate the PSF. Given the mechanical properties and input commands to the system, a discretized PSF can be estimated in an open-loop manner, without the use of external sensors as shown in Figure 10. This approach results in the fast estimation of the PSFs using the robotic camera orientation system because it is estimated simultaneously with the image acquisition in an embedded motion controller.

The mechanical system was modeled as a linear second-order system. It has been reported by Schultz and Ueda that a linear second-order model is sufficient in the range of the motion of interest and any significant nonlinearity was not observed in the camera orientation mechanism when it was controlled by discrete switching commands [97]. Also, it has been reported by Schultz and Ueda that a linear dynamic model is sufficient for model-based control [96]. In addition, the experiment results comparing the real and calculated responses clarify the validity of the linear model as shown in Figure 11. To apply the proposed control method to a genral mechanism, issues associated with possible nonlinearity in the structure may need to be resolved.

Figure 11 shows the experiment results of a comparison between the actual motion

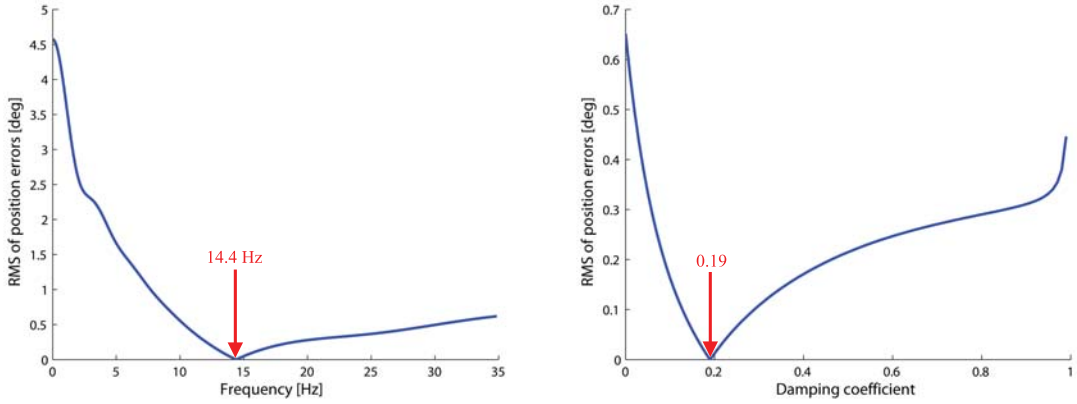


(a) (b)

Figure 11: Comparisons between the actual motion and the estimated motion. (a) Time response of the camera orientation system given the step command. (b) Time response of the camera orientation system given the discrete switching commands.

and the estimated motion. The actual motion was recorded by an encoder and the estimated motion was calculated by (6) implement on the embedded controller.

Figure 11(a) shows a comparison for step commands. For this particular mechanism, nonlinearity was not observed when the desired step was 10 degrees or less. A slight nonlinearity was observed where the natural frequency was 16.1 Hz when the desired step input was a full stoke step while the natural frequency was 14.4 Hz when the desired step input was below 77% of the full stroke. This was due to a saturation effect. The overshoot caused the third layer to collide with the first layer units [96]. Although the nonlinearity was observed, approximately a variation of 12% of the natural frequency, when the desired step was beyond 77% of the full stroke, the repeatability of the response was very high and the motion could be accurately estimated by using the predefined values of natural frequency accordingly as shown in 11(a). In this study, the range of step input where any significant nonlinearity was not observed was used as an input. The estimated motions show a good agreement with the actual motion overall with only minor errors after settling possibly due to flexibility of the cellular actuators and mechanical frictions.



(a) (b)

Figure 12: Sensitivity plots (a) Sensitivity plot for the frequency (b) Sensitivity plot for the damping coefficient

Figure 11(b) shows the results for vibration suppression discrete switching commands where a good agreement is observed. When the discrete switching commands were used to generate a motion, any significant nonlinearity in the mechanism was not observed.

3.3 Sensitivity Analysis of the Dynamics-based Approach

This approach assumes that the system dynamics is modeled accurately and the motion has no unpredictable disturbances. Figure 12(a) and 12(b) show sensitivity analysis for system parameters given the discrete switching commands shown in Figure 6(b). The sensitivity analysis was performed because the performance of the proposed method depends on the accuracy of system modeling. The system has no RMS position errors if the system parameters are calibrated accurately at frequency of 14.4Hz and damping coefficient of 0.19 as shown in Figure 12. The analysis shows that the motion is not estimated accurately with parameter errors, resulting in a poor estimation of the PSFs. In particular, the estimation model is not robust to the frequency parameter error. Although the performance is sensitive to the parameter calibration as expected, the calibration in general can be done with a reasonable

precision. Any issues regarding the parameter calibration were not observed in this study. It is possible to add command constraints on robustness to variations in frequency or damping coefficient but it results in more switching of commands and a longer settling time.

While the system in this research was modeled as a linear second-order system, the dynamics-based motion de-blurring method is theoretically applicable to a higher order linear dynamic system or a non-linear system. The motion of the system can be estimated by knowing the accurate dynamics model and control commands. However, the current method is not applicable to a system experiencing significant external disturbances.

3.4 Deconvolution

A spatially-invariant blurred image $\boldsymbol{B}$ can be represented by a convolution between a shift-invariant blur kernel $\boldsymbol{K}$ and a latent sharp image $\boldsymbol{I}$ plus noise $\boldsymbol{N}$, as follows:

$$\boldsymbol{B} = \boldsymbol{K} \otimes \boldsymbol{I} + \boldsymbol{N} \tag{12}$$

$$\boldsymbol{B}, \boldsymbol{K}, \boldsymbol{I} \in \mathbb{R}^{m \times n}$$

$$\boldsymbol{K} \in \mathbb{R}^{l \times l}$$

where m is the width of the image in pixels, n is the height of the image in pixels, l is the length or height of the blur kernel, and $\otimes$ is the convolution operator.

As the PSF is obtained, the latent image can be estimated by means of a deconvolution algorithm. In this study, a widely-used Richardson-Lucy deconvolution method was selected [89]. This method is known to be robust in the presence of high noise levels. It estimates the latent image iteratively given the blurry input image and the estimated PSF [29].

$$\boldsymbol{I}^{n+1} = ((\boldsymbol{B} \oslash (\boldsymbol{I}^n \otimes \boldsymbol{K})) \otimes \hat{\boldsymbol{K}}) \odot \boldsymbol{I}^n \tag{13}$$

where $\boldsymbol{B}$ is the obtained blurry image, $\boldsymbol{K}$ is the blur kernel, $\hat{\boldsymbol{K}}$ is the flipped blur kernel, $\boldsymbol{I}^n$ is the estimated image at n^{th} iteration, $\odot$ is the elementwise multiplication operator, and $\oslash$ is the elementwise division operator.

For the deconvolution process, MATLAB's deconvlucy.m function was used.

3.5 Evaluation Method

3.5.1 Compared Algorithms

Images with motion blur were obtained under various conditions to verify the proposed method. The results were compared with those of other notable and state-of the art algorithms proposed by: Fergus *et al.* [27], Shan *et al.* [105], Xu *et al.* [121], Goldstein *et al.* [33], Whyte *et al.* [120], and Cho *et al* [18]. These compared algorithms analyze inherent image properties and include parameters such as dimension of the blur kernel that need to be manually defined by the users. If the parameters are not properly determined, the resultant images would be deteriorated. For a fair comparison, the parameters were adjusted a number of times to find the best possible blur kernels. The results were compared qualitatively and quantitatively.

3.5.2 Quantitative Evaluation

In general, previous image de-blurring studies have qualitatively evaluated their results in comparison with those of other algorithms. Although the results of such evaluation were successfully discussed, it was a subjective evaluation. In this study, the results of the proposed de-blurring algorithm will be evaluated both qualitatively and quantitatively. The quantitative evaluation method subtracts the clean image with no motion blur from the estimated latent image and evaluates the pixel values of the difference image. The procedures of a quantitative evaluation method are as follows:

1. obtain a clean image with no motion blur at the closest discretized position to the blurry image;

2. convert both the de-blurred image and the clean image into grayscale;

3. pattern match between the estimated latent image and the clean image to find the translation matrix;

4. translate the clean image based on the previous step;

5. determine the absolute of the difference between the de-blurred image $\boldsymbol{I}_{de-blurred}$ and the translated clean image $\boldsymbol{I}_{clean}$;

$$\boldsymbol{I}_{diff} = |\boldsymbol{I}_{clean} - \boldsymbol{I}_{de-blurred}| \tag{14}$$

6. trim 10% of the image border to remove the boundary artifacts from the image deconvolution;

7. evaluate;

$$Difference\ Value = \frac{\sum_{j=1}^{n}\sum_{i=1}^{m}\boldsymbol{I}_{diff(i,j)}}{m \times n} \tag{15}$$

8. normalize $DifferenceValue$ by $DifferenceValue$ of the proposed method

where $\boldsymbol{I}_{diff(i,j)}$ is a pixel value at pixel (i, j).

The lower difference value results from the higher correlation between the estimated latent image and the clean image with no motion blur. Therefore, it can be concluded that the better de-blurring method will have a lower difference value.

3.6 Experimental Results

3.6.1 Vibration Suppression Input and Step Input

To investigate the effect of the motion controller on the image quality, the proposed de-blurring method was tested for two different command inputs: vibration suppression and step inputs.

Figure 15 shows the results of the proposed image de-blurring method and others when the system was oriented by vibration suppression commands. The input blurry

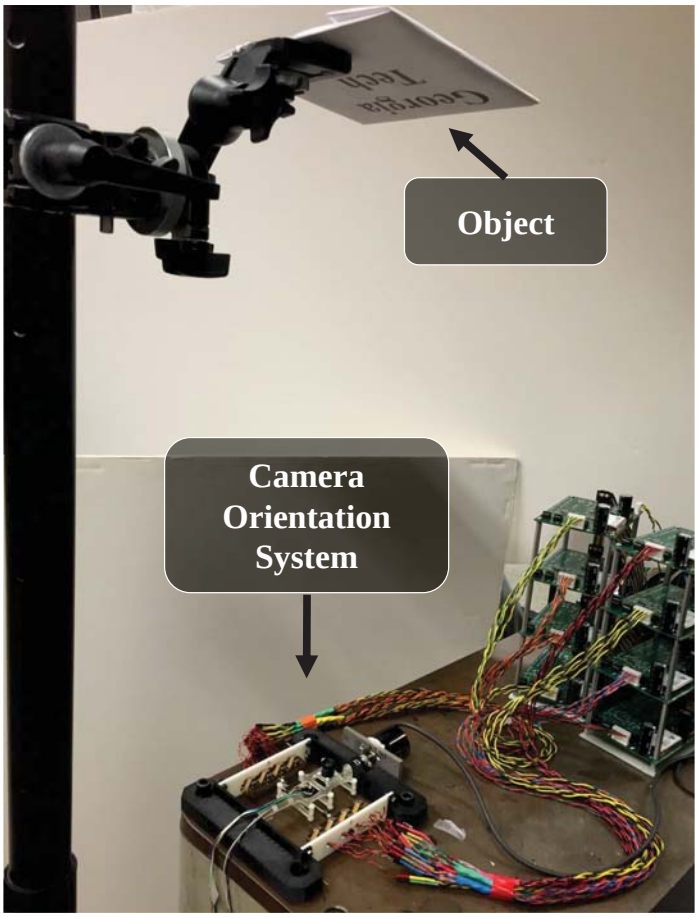


Figure 13: Experimental setup for image de-blurring

image is the frame of interest indicated in Figure 7(a). The estimated PSFs are shown in Figure 16. The intensity profile of the PSFs has a left-shifted center since the motion of the selected image frame was settled to the desired position. The amount of time that the device position was around the target position was dominant within a single exposure window, when the motion was created by the vibration suppression control as shown in Figure 7(a). Therefore, the intensity value in the PSF was accumulated relatively high around the target region, which corresponds to the left side of the PSF. The results show that all of the methods removed the blur because the vibration suppression technique reduced the motion blur. However, the results obtained by Goldstein *et al.*, Shan *et al.*, and Xu *et al.* present ringing artifacts. The results show that the proposed method, Whyte *et al.*s method, and

Fergus *et al*.s methods have successfully estimated the latent image.

Figure 17 shows the results of the proposed image de-blurring method and others when the system was oriented by step input. The input blurry image is the frame of interest indicated in Figure 7(b), which is the same frame in the previous case. The PSFs have a left-shifted center as shown in Figure 18. This is because the motion of the selected image frame was oscillated around the second peak where the intensity is relatively high. However, the intensity profile is more distributed than the previous case because the velocity profile is almost constant except the oscillatory region at the end of the interested frame as shown in Figure 7(b). The results show clearer comparisons than the previous case because the input image was more degraded. The results obtained by Goldstein *et al*., Fergus *et al*., and Whyte *et al*.s are still blurry. The results of Shan *et al*. and Xu *et al*. show an improvement, but the ringing artifacts still exist. The proposed method shows the most distinguishable letters and the best overall result.

Comparing the results, it can be concluded that it is preferable to give the discrete switching commands as an input because it has a faster settling time and less motion blur in the image, resulting in better estimation of the latent image regardless of the image de-blurring method.

3.6.2 Oscillatory Input

The proposed method has been discussed and compared to other approaches when the system experiences uni-directional movement. In this section, the proposed method was investigated when the system runs an oscillatory motion to verify that the proposed method performs effectively when the motion paths are overlapping or intersecting. For this experiment, it is preferable to have longer exposure times due to the limitation of working around the natural frequency of the camera orientation device. The longer exposure time ensures that the device experiences full-cycle of oscillation,

but also make image acquisition susceptible to excess light. Therefore, images were obtained while indoor lights are off. Blurry images were obtained under oscillatory conditions with three different amplitudes as shown in Figure 14. The amplitudes were one PZT step, two PZT steps, and four PZT steps. Images are obtained at 10.4 fps. The indicated image frame shown in Figure 14 was investigated for all three amplitudes.

Figures 20, 22 and 24 show the estimated PSFs with different oscillation amplitudes. The PSFs have considerably uniform intensity distribution with low-amplitude multiple peaks. The amount of time staying at each pixel of the blur kernel within a single exposure window has a small variation as the camera was positioned. Therefore, the PSFs exhibit multiple peaks that is a natural consequence as the system experienced full cycle of oscillation.

Figure 19 shows image de-blurring results when the system obtained images while having an oscillatory motion created by one PZT step. Shan *et al.*s result is degraded due to an existence of dark noise in the capture image. Most of the methods show an improvement over the input blurry images but have small discrepancies from the image with no motion blur. The results of Goldstein *et al.* and Xu *et al.* have a small amount of ringing artifacts and the results of the proposed method and Whyte *et al.*s method are locally blurry. The proposed method, however, shows less blur globally.

Figure 21 shows image de-blurring results when the system obtained images while having an oscillatory motion created by two PZT steps. It can be checked again that Shan *et al.*'s method is not robust to low ambient light. The results of Goldstein *et al.* and Xu *et al.* are degraded with ringing artifacts and the result Fergus *et al.* still presents a large amount of blur. Although the results of the proposed method and Whyte *et al.* have a discrepancy with the image with no motion blur, those have the best results compared to the others.

Figure 23 shows image de-blurring results when the system obtained images while

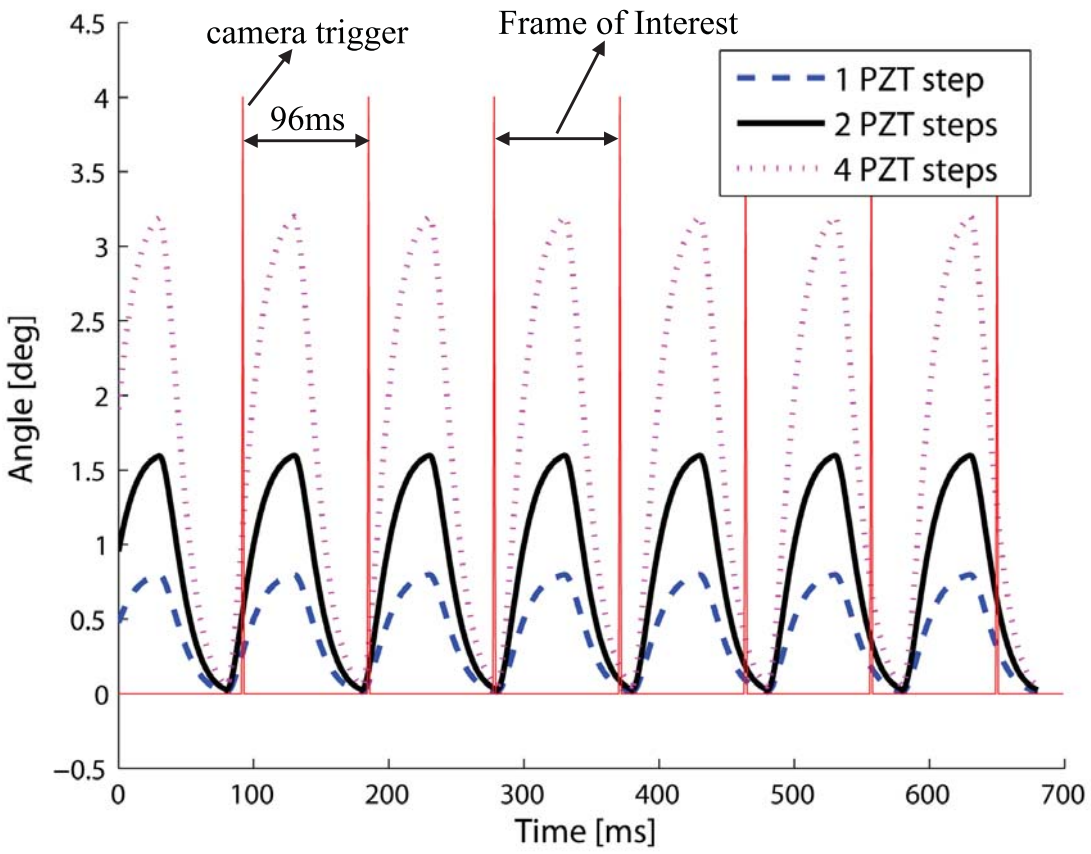


Figure 14: Oscillatory responses of the camera orientation system with various amplitudes. Images are obtained at 10.4fps.

having an oscillatory motion created by four PZT steps. The results of Fergus *et al.* and Shan *et al.* have not estimated the blur kernels properly such that the letters are not distinguishable. The results of Goldstein *et al.* and Whyte *et al.* show an improvement but outputs are still blurry. Xu *et al.*' method has greatly improved the image but some of the characters are not clear. The proposed method shows the best qualitative result in that all characters except 'e' are recognizable.

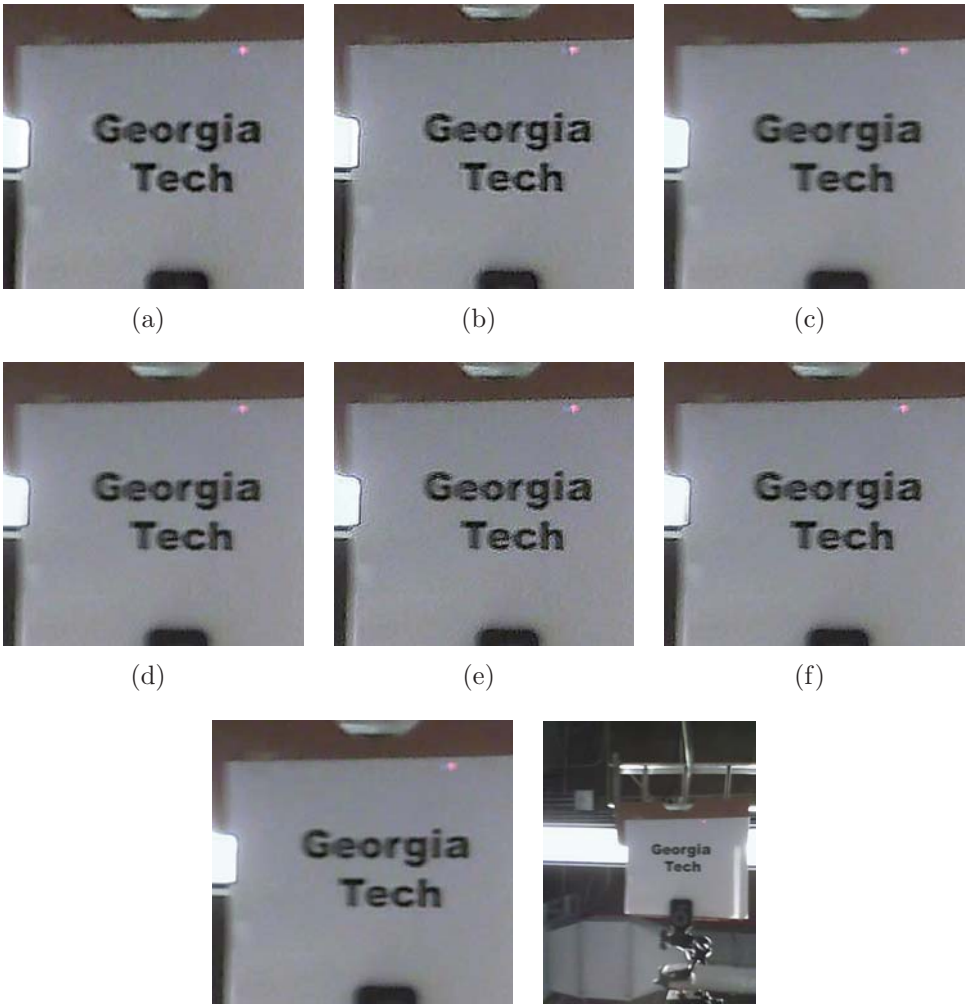


Figure 15: Test set #1. The obtained blurry image and estimated latent images by various methods. Images are taken at 29fps with Vibration Suppresion input command. (a) - (g) are cropped local images. (a) Proposed Method (b) Goldstein *et al*. (c) Fergus *et al*. (d) Whyte *et al*. (e) Shan *et al*. (f) Xu *et al*. (g) Cropped obtained blurry image (h) Obtained blurry image

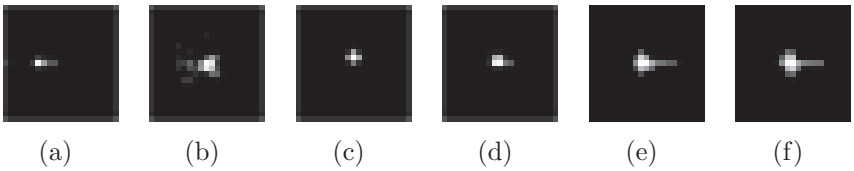


Figure 16: Test set #1. Estimated blur kernels. The size of the PSFs is 21 × 21. (a) Proposed Method (b) Goldstein *et al*. (c) Fergus *et al*. (d) Whyte *et al*. (e) Shan *et al*. (f) Xu *et al*.

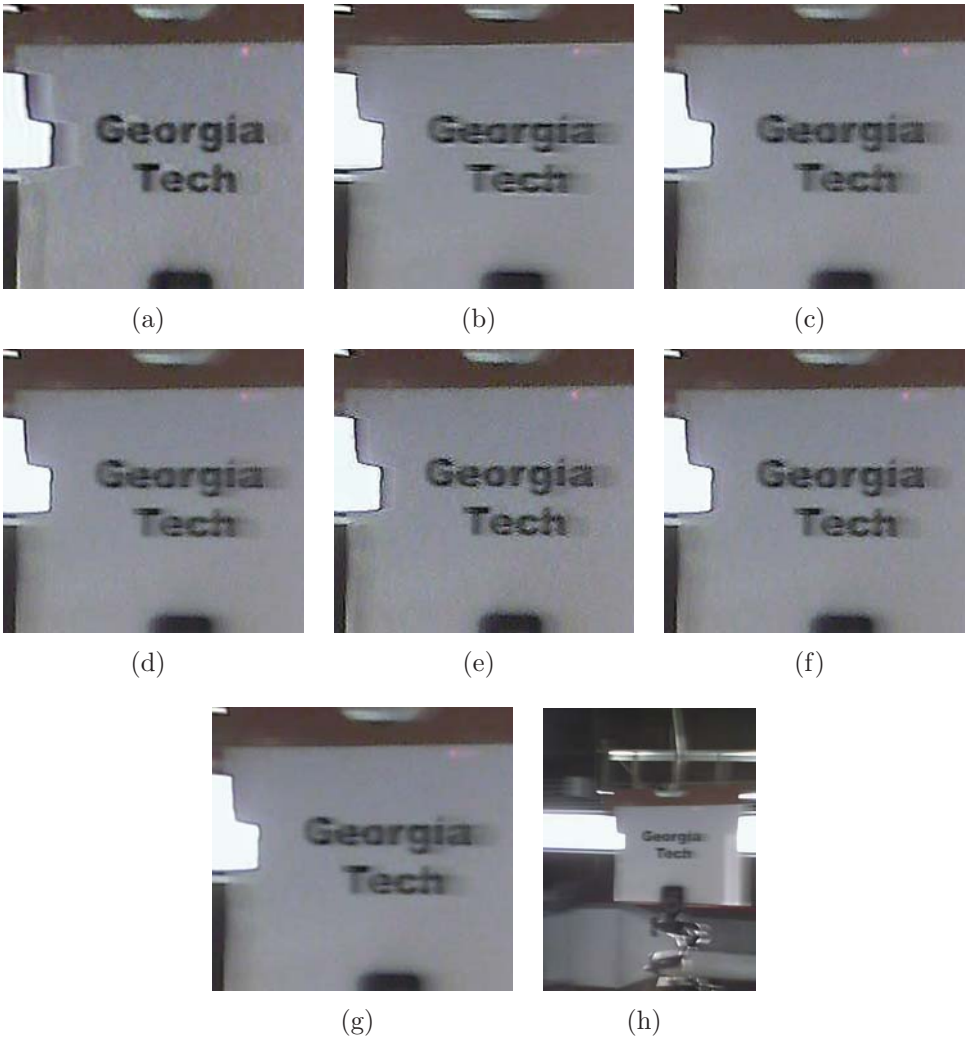


Figure 17: Test set #2. The obtained blurry image and estimated latent images by various methods. Images are taken at 29fps with Step input. (a) - (g) are cropped local images. (a) Proposed Method (b) Goldstein *et al*. (c) Fergus *et al*. (d) Whyte *et al*. (e) Shan *et al*. (f) Xu *et al*. (g) Cropped obtained blurry image (h) Obtained blurry image

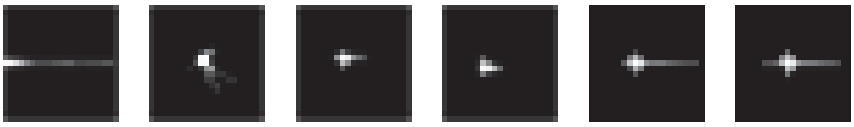


Figure 18: Test set #2. Estimated blur kernels. The size of the PSFs is 21 × 21. (a) Proposed Method (b) Goldstein *et al*. (c) Fergus *et al*. (d) Whyte *et al*. (e) Shan *et al*. (f) Xu *et al*.

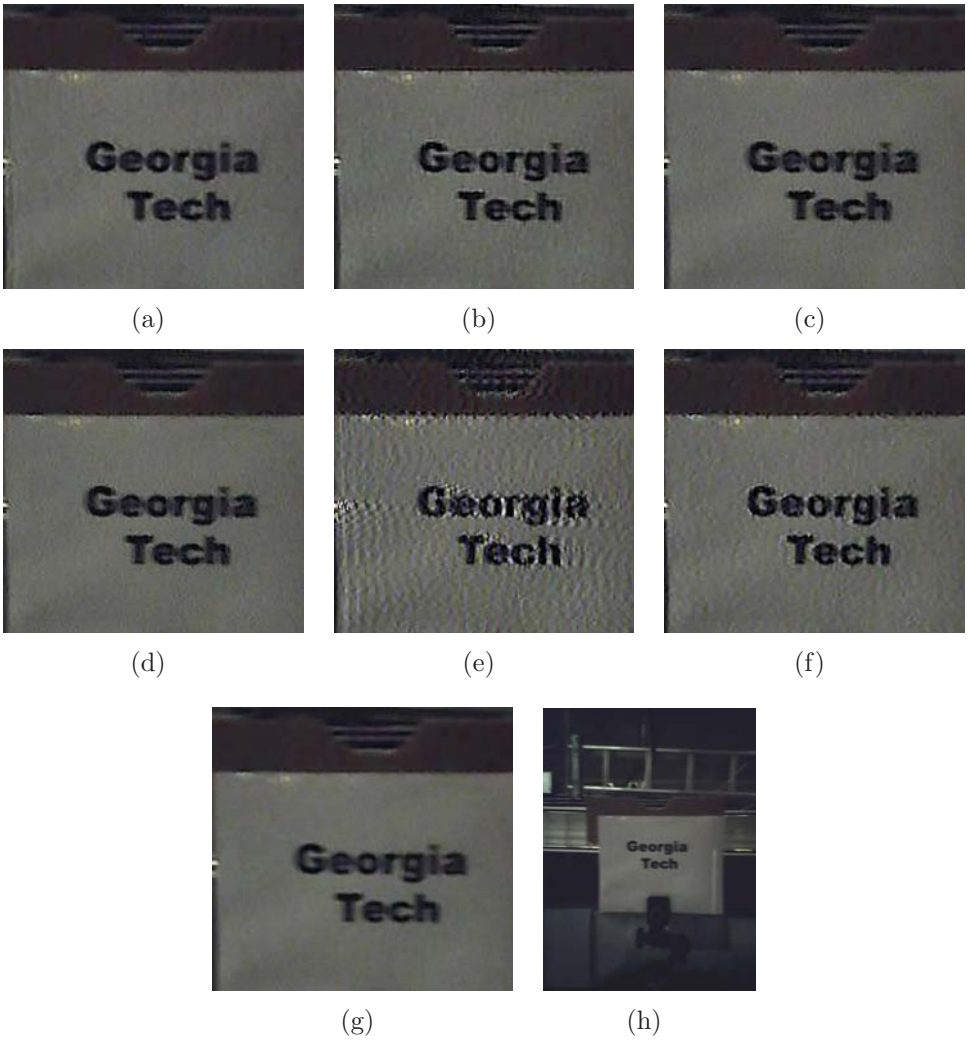


Figure 19: Test set #3. The obtained blurry image and estimated latent images by various methods. Images are taken at 10.4fps with periodic oscillatory input. One PZT step was used to create oscillatory response. (a) - (g) are cropped local images. (a) Proposed Method (b) Goldstein *et al*. (c) Fergus *et al*. (d) Whyte *et al*. (e) Shan *et al*. (f) Xu *et al*. (g) Cropped obtained blurry image (h) Obtained blurry image

Figure 20: Test set #3. Estimated blur kernels. The size of the PSFs is 11 × 11. (a) Proposed Method (b) Goldstein *et al*. (c) Fergus *et al*. (d) Whyte *et al*. (e) Shan *et al*. (f) Xu *et al*.

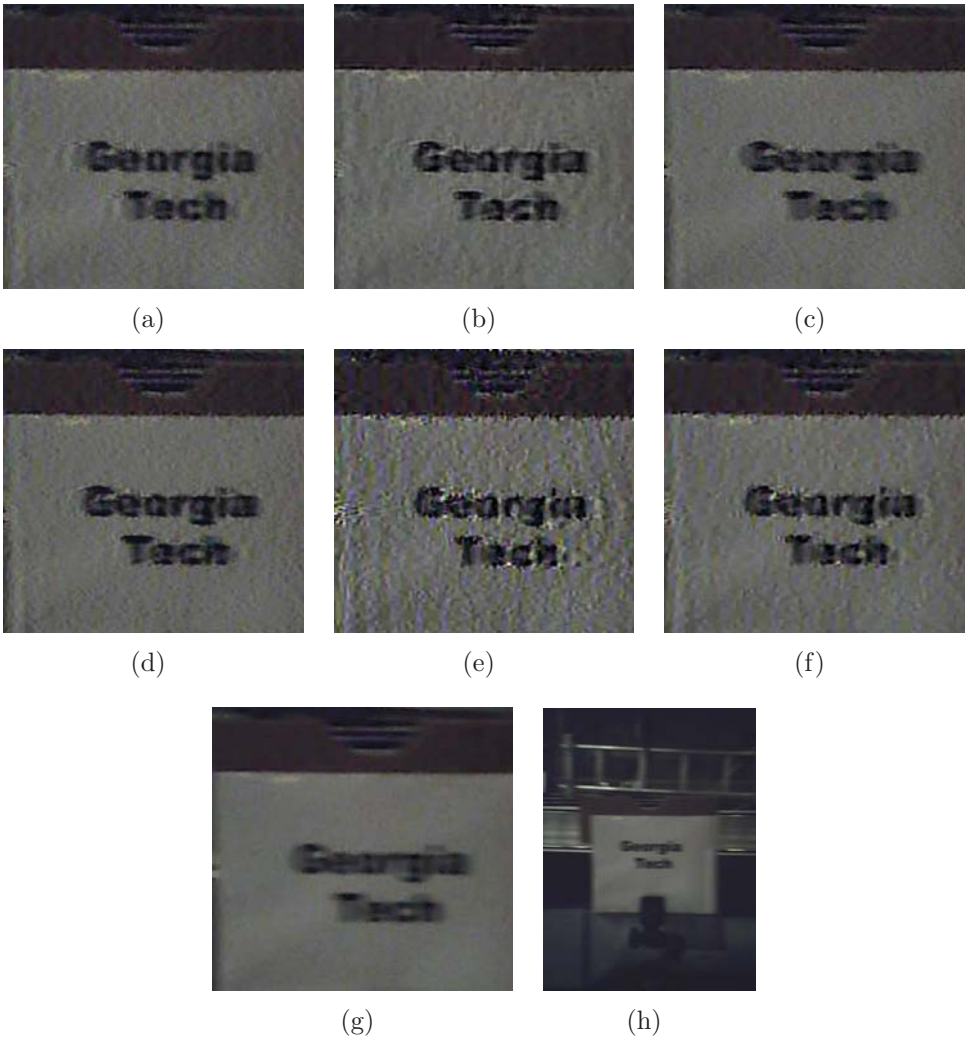


Figure 21: Test set #4. The obtained blurry image and estimated latent images by various methods. Images are taken at 10.4fps with periodic oscillatory input. Two PZT steps were used to create oscillatory response. (a) - (g) are cropped local images. (a) Proposed Method (b) Goldstein *et al*. (c) Fergus *et al*. (d) Whyte *et al*. (e) Shan *et al*. (f) Xu *et al*. (g) Cropped obtained blurry image (h) Obtained blurry image

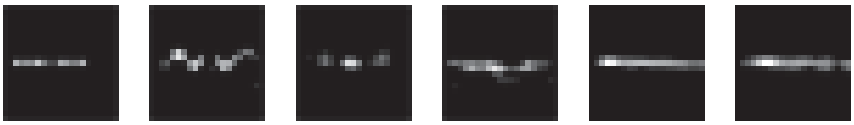


Figure 22: Test set #4. Estimated blur kernels. The size of the PSFs is 21 × 21. (a) Proposed Method (b) Goldstein *et al*. (c) Fergus *et al*. (d) Whyte *et al*. (e) Shan *et al*. (f) Xu *et al*.

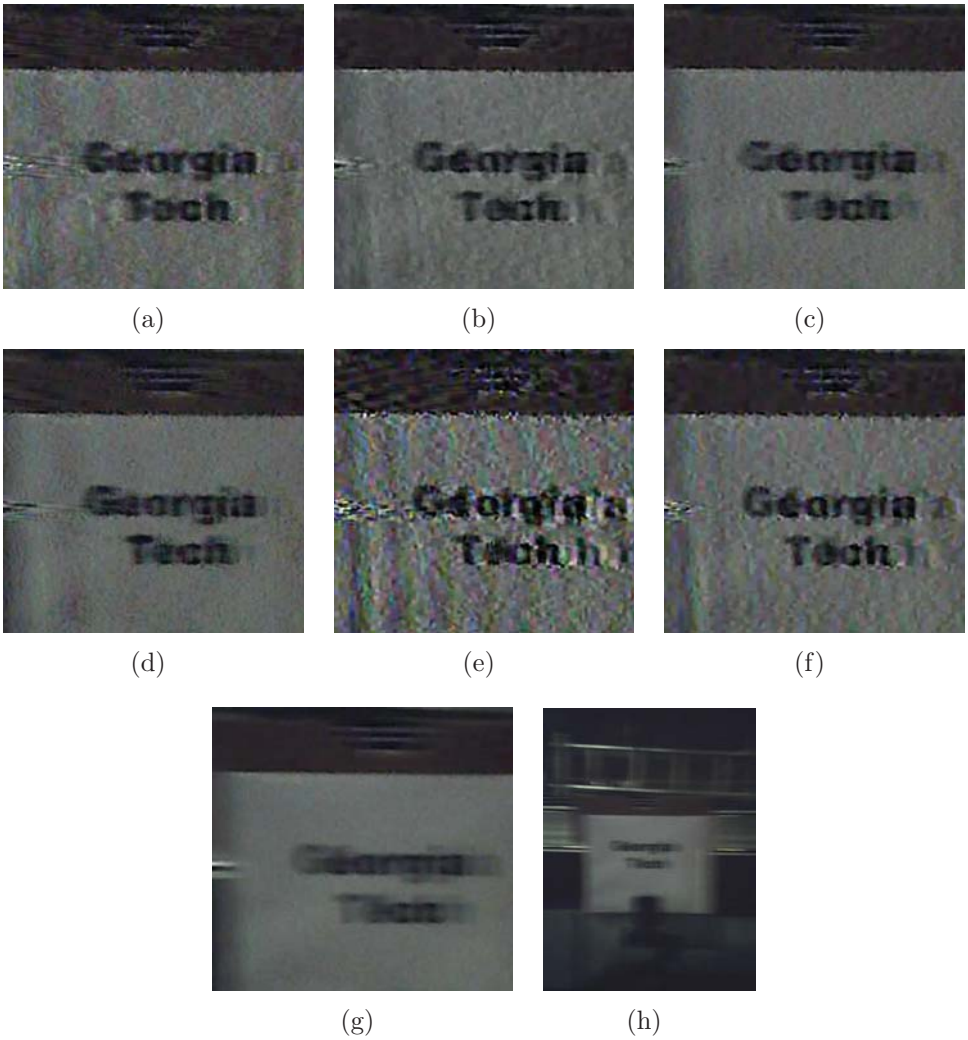


Figure 23: Test set #6. The obtained blurry image and estimated latent images by various methods. Images are taken at 10.4fps with periodic oscillatory input. Four PZT steps were used to create oscillatory response. (a) - (g) are cropped local images. (a) Proposed Method (b) Goldstein *et al*. (c) Fergus *et al*. (d) Whyte *et al*. (e) Shan *et al*. (f) Xu *et al*. (g) Cropped obtained blurry image (h) Obtained blurry image

Figure 24: Test set #6. Estimated blur kernels. The size of the PSFs is 35 × 35. (a) Proposed Method (b) Goldstein *et al*. (c) Fergus *et al*. (d) Whyte *et al*. (e) Shan *et al*. (f) Xu *et al*.

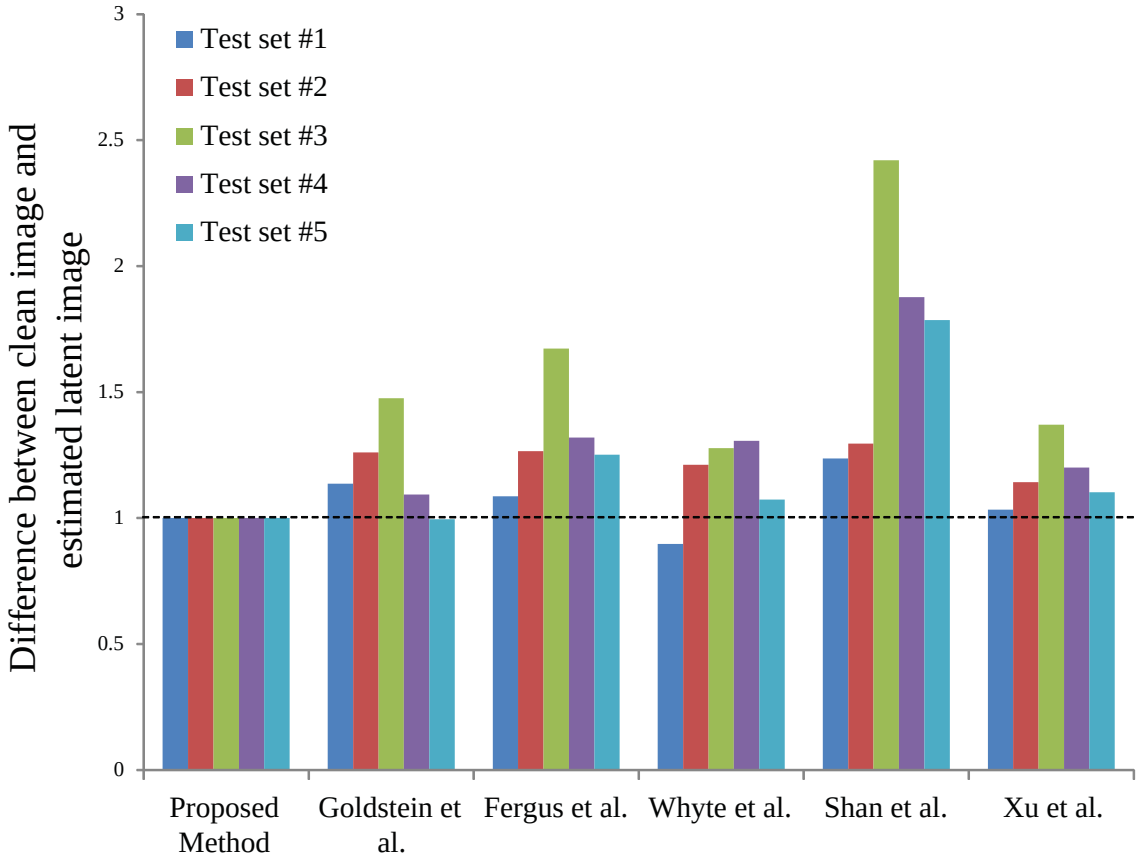


Figure 25: Quantitative evaluation of the de-blurred results

3.6.3 Quantitative Evaluation

The results of the de-blurring methods have been discussed in a qualitative manner. This section presents a quantitative evaluation of the de-blurred images by the method suggested in Section 3.5.2. Figure 25 shows quantitative evaluations for test sets #1 − #5. For the test set #1, the proposed method has the second best estimation according to the suggested quantitative evaluation. For the test set #2, the proposed method has the best quantitative result, which agrees with the qualitative evaluation that the proposed method shows the most distinguishable letters. The proposed also has the best quantitative results when de-blurring images that are obtained in the course of oscillatory movements as it can be checked from test sets #3 − #5.

The effectiveness of the proposed algorithm was investigated under various motion

conditions. The proposed algorithm estimated the PSFs overall better than conventional methods. The experiment results show that the performance metric of the proposed method is 16.96% better than the 2nd best algorithm and 31.14% better than the other algorithms on average. It can be concluded from the experimental results and the quantitative evaluations that the proposed algorithm has properly estimated the PSFs and showed better image de-blurring results than other approaches. However, residual blur or ringing exists in the image for some cases because it is theoretically impossible to recover high-frequency image from the images blurred with low-pass PSF.

3.7 Computation Time

Image de-blurring techniques is in general known to be computationally expensive. The proposed method, however, estimates the blur kernel in an open loop manner as the images are obtained. This approach results in a significant reduction of computation time. Figure 26 and Table 2 show computation times for each algorithm for each test presented in this study. A total of 10 trials are conducted for each of the algorithm and for each test to evaluate average computation times. Computation times include estimation of a PSF and a deconvolution process. Computation times were measured by running either MATLAB or compiled executable programs distributed by the authors. For comparison, blur kernel sizes are fixed for each case. The properties that are used for comparing computation times are shown in Table 3.

The results show that the proposed method is fastest among all the methods. This is because the PSF is estimated from the embedded motion controller simultaneously. The proposed method estimated latent images 4.8 times faster than the 2nd fastest algorithm and 51 times faster than the other algorithms on average. The computation time of the proposed method is not highly dependent on the size of the image and the blur kernel. On the contrary, the computation time is highly dependent on those for

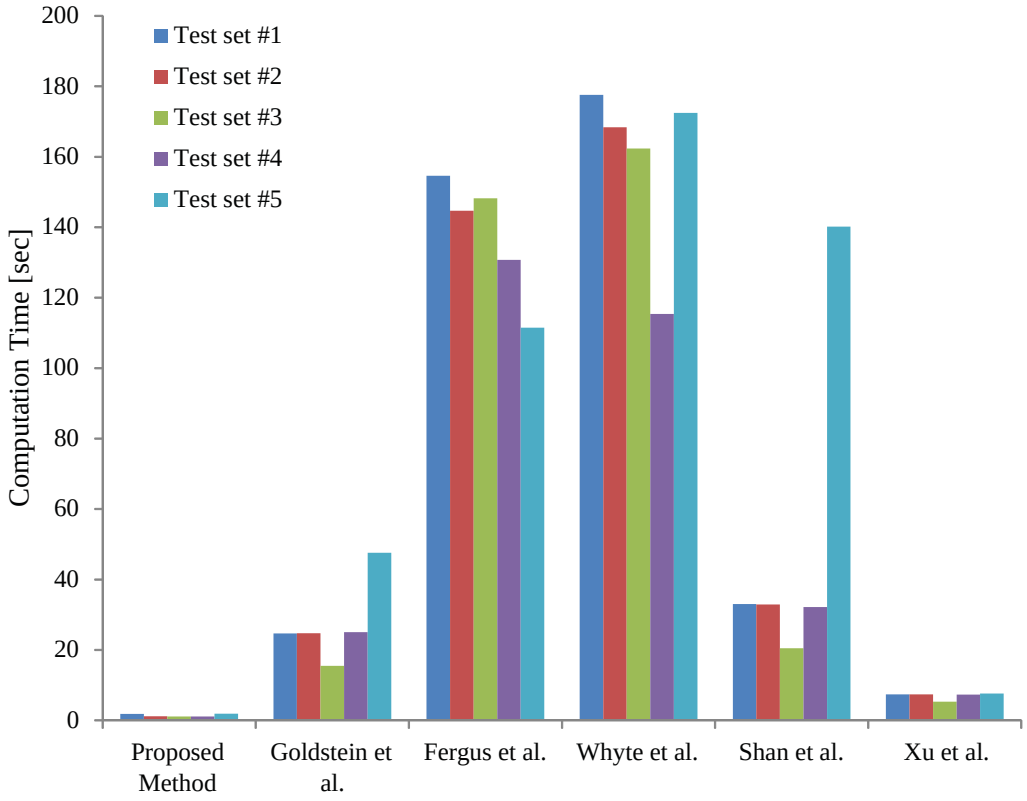


Figure 26: Comparisons of computation times

the methods proposed by Goldstein *et al.*, Fergus *et al.*, Whyte *et al.*, and Shan *et al.* This is because those are based on a statistical model of the image or a series of optimization techniques. Most of the computation efforts of the proposed method result from the iterative deconvolution process. Since the proposed method was executed in MATLAB, the computation times can be reduced further by executing a compiled program in GPU or FPGA, which will be discussed in Chapter 4. In conclusion, the proposed method is the least computationally expensive.

Table 2: Comparisons of Computation Times [sec]

Methods	Proposed	Goldstein *et al.*	Fergus *et al.*	Whyte *et al.*	Shan *et al.*	Xu *et al.*
Software	MATLAB	MATLAB	MATLAB	MATLAB	Compiled Executable	Compiled Executable
Set #1	1.84	24.68	154.63	177.58	33.06	7.41
Set #2	1.17	24.76	144.66	168.39	32.88	7.41
Set #3	1.12	15.47	148.19	162.37	20.45	5.31
Set #4	1.12	25.05	130.71	115.36	32.20	7.35
Set #5	1.88	47.59	111.50	172.45	140.16	7.61

Table 3: Image sizes that were used to estimate computation times

Properties	Image Size	PSF Size
Set #1	640x480	21x21
Set #2	640x480	21x21
Set #3	640x480	11x11
Set #4	640x480	21x21
Set #5	640x480	35x35

3.8 Experiments with Different Scenes

This section presents experiment results when the system was capturing different scenes and oriented by vibration suppression commands. For test sets #6 - #9, the results were compared with the algorithms proposed Xu *et al.* [121], Goldstein *et al.* [33], and Cho *et al* [18].

Figure 28 shows a comparison of the de-blurring results for various cases. The first two rows show results when UVGA-size images were used and the last two rows show results when VGA-size images were used. The first column shows the input images, the second column shows the results obtained by the proposed methods, and the other columns are results obtained by the compared methods. The degree of blurriness was dependent on the velocity during the exposure window and the velocities of the test sets #6, #7, #8, and #9 were 16.25, 30.47, 49.99, and 37.49 deg/sec, respectively.

For the test set #6, all the methods successfully recovered the blurry input image due to a low degree of blurriness. However, the results obtained by Xu *et al.* and Goldstein *et al.* show minor artifacts around the edges of the 'Lab logo (the name of the authors' laboratory with a mascot character)'. For the test set #7, the proposed method exhibits the least blurriness in particular around the 'Lab logo' and 'First Aid' regions, although it has a relatively low contrast around sharp edges. The result obtained by Cho *et al.* shows a fine recovery and contrast overall, although ringing artifacts exist all over the image. Different parameter values were tested for Cho's method to avoid ringing artifacts. However, the final results were deteriorated. For

Table 4: Comparisons of Computation Times [sec]

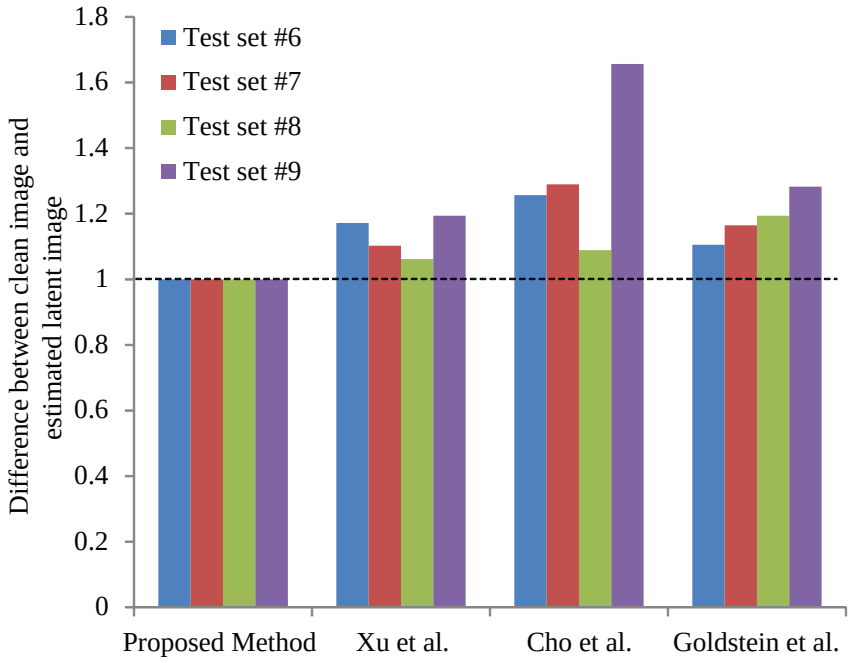
Set	Image Size	Kernel Size	Proposed Method	Xu *et al.*	Cho *et al.*	Goldstein *et al.*
#6	UVGA	35×35	2.32	13.85	15.81	29.71†
#7	UVGA	75×75	3.32	14.57	18.13	80.35†
#8	VGA	41×41	1.20	8.17	4.99	30.66
#9	VGA	31×31	0.93	7.87	4.96	49.64

† SVGA image was used due to a software limitation

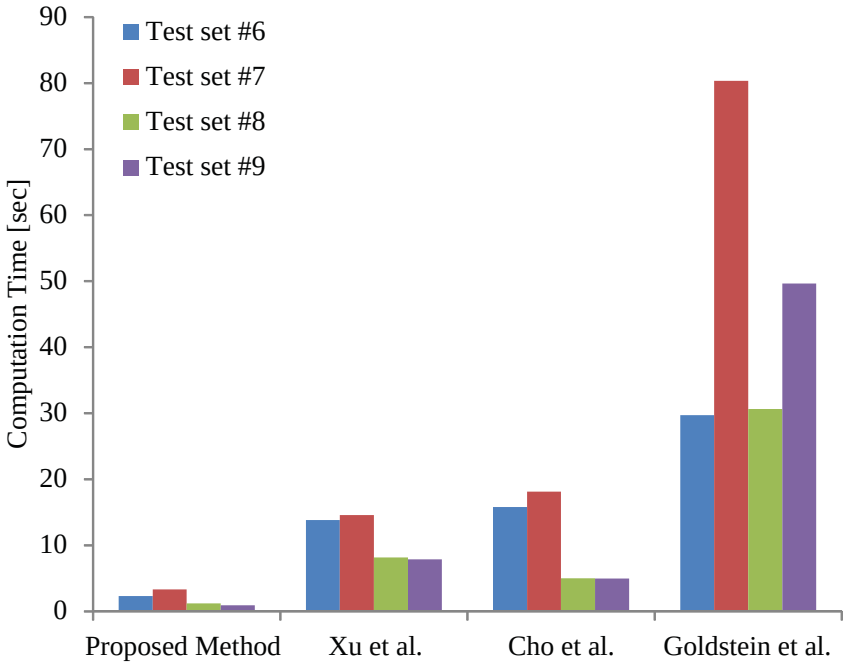
the test set #8, all the methods were not able to perfectly recover the blurry image possibly due to high velocity. Overall, it can be concluded that the proposed method obtained the best result by observing the 'Lab logo' and 'First Aid' regions. For the last test set, all the methods showed yielded improved results, with the proposed method providing the best result, based on comparison of the 'Lab logo' regions.

Overall, the proposed method obtained the best results in terms of geometrical recovery while the method created a relatively low contrast around some sharp edges. This issue can be improved by applying a sharpening filter to the final outcome by allowing a slight increase in the processing time. The total process time would increase less than 1% when using a sharpening filter depending on the dimension of the blur kernel. In this study, a sharpening filter was not introduced for a fair comparison with other methods.

Figure 27(a) shows quantitative evaluations for test sets $\#6 - \#9$. It can be found again in the graph that the proposed method provides the lowest scores for all the test cases, which means that the closest result to the clean static image can be obtained. The experiment results show that the performance metric of the proposed method is 13.23% better than the 2nd best algorithm and 21.39% better than the other algorithms on average. As can be observed in Table 4 and Figure 27(b), the proposed method provides the fastest computation time. The proposed method estimated latent images 5.7 times faster than the 2nd fastest algorithm and 12 times faster than the other algorithms on average. The experiment results in regard to



(a)



(b)

Figure 27: Experimental results of the test sets #6 − #9. (a) Quantitative Evaluation. (b) Comparisons of computation times.

both quantitative evaluations by the performance metric of the proposed method and computation times agree with the results presented in the previous section.

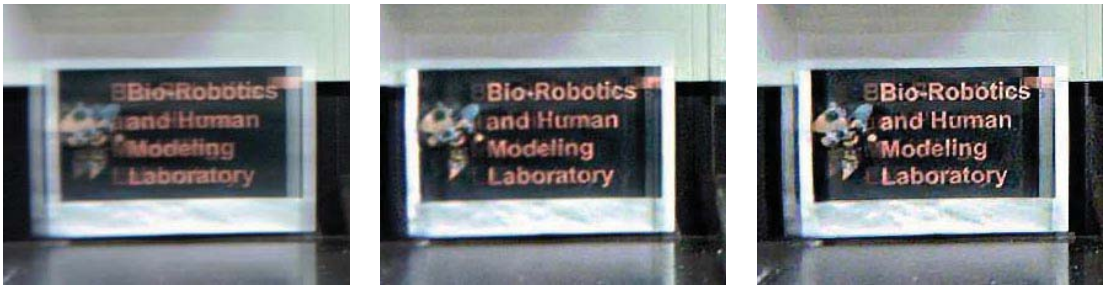
(a) Blurry input (UVGA) (b) Proposed method (c) Xu *et al.* (d) Cho *et al.* (e) Goldstein *et al.*

(f) Blurry input (UVGA) (g) Proposed method (h) Xu *et al.* (i) Cho *et al.* (j) Goldstein *et al.*

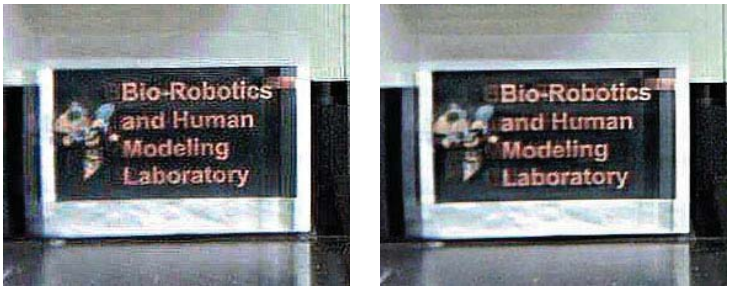
(k) Blurry input (VGA) (l) Proposed method (m) Xu *et al.* (n) Cho *et al.* (o) Goldstein *et al.*

(p) Blurry input (VGA) (q) Proposed method (r) Xu *et al.* (s) Cho *et al.* (t) Goldstein *et al.*

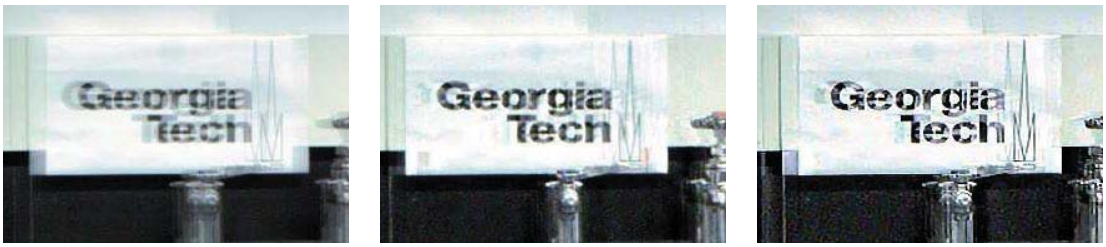
Figure 28: Comparisons of the de-blurring results. Test set #6: Figures 28(a) - 28(e). Test set #7: Figures 28(f) - 28(j). Test set #8: Figures 28(k) - 28(o). Test set #9: Figures 28(p) - 28(t).



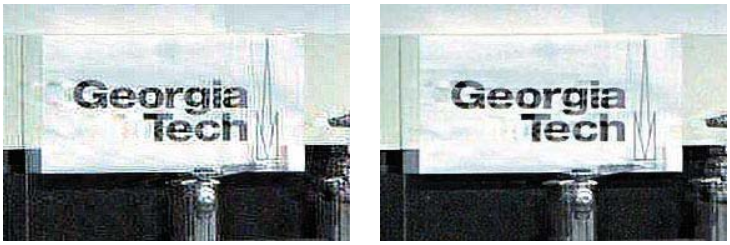
(a) Blurry input (b) Proposed method (c) Xu *et al.*



(d) Cho *et al.* (e) Goldstein *et al.*



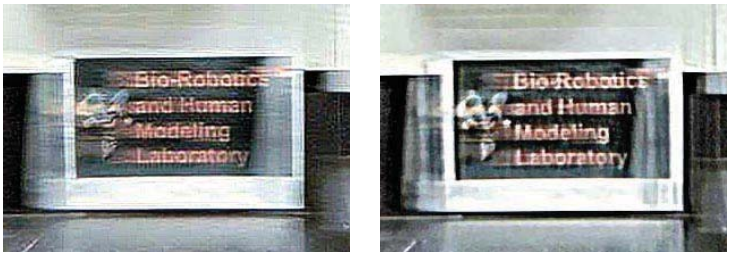
(f) Blurry input (g) Proposed method (h) Xu *et al.*



(i) Cho *et al.* (j) Goldstein *et al.*

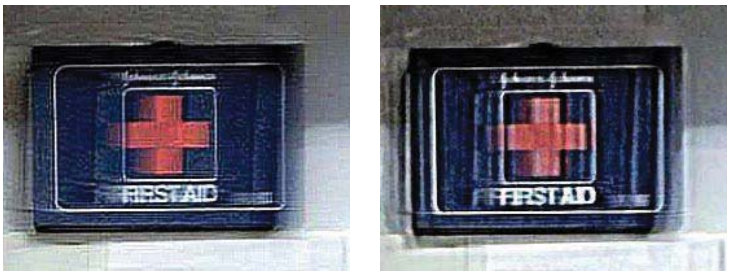
Figure 29: Test set #6. Cropped local images from Figures 28(a) - 28(e).

(a) Blurry input (b) Proposed method (c) Xu *et al.*



(d) Cho *et al.* (e) Goldstein *et al.*

(f) Blurry input (g) Proposed method (h) Xu *et al.*



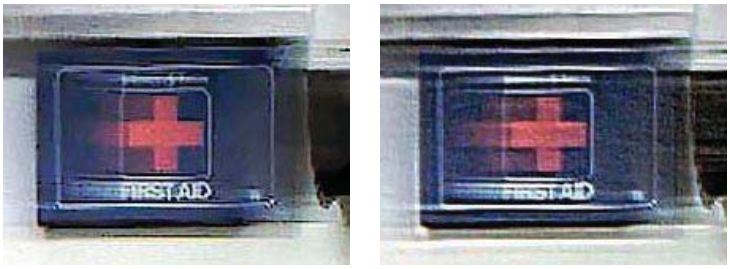
(i) Cho *et al.* (j) Goldstein *et al.*

Figure 30: Test set #7. Cropped local images from Figures 28(f) - 28(j).

(a) Blurry input (b) Proposed method (c) Xu *et al.*

(d) Cho *et al.* (e) Goldstein *et al.*

(f) Blurry input (g) Proposed method (h) Xu *et al.*



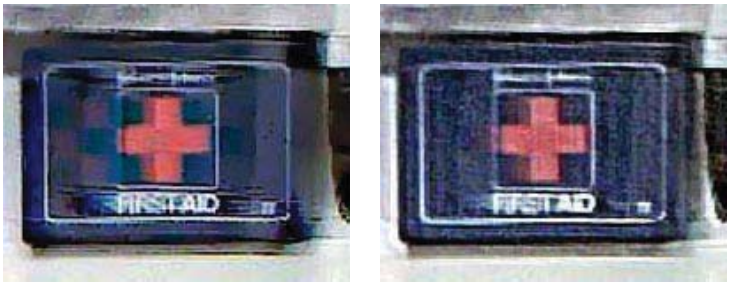
(i) Cho *et al.* (j) Goldstein *et al.*

Figure 31: Test set #8. Cropped local images from Figures 28(k) - 28(o).

(a) Blurry input (b) Proposed method (c) Xu *et al.*

(d) Cho *et al.* (e) Goldstein *et al.*

(f) Blurry input (g) Proposed method (h) Xu *et al.*



(i) Cho *et al.* (j) Goldstein *et al.*

Figure 32: Test set #9. Cropped local images from Figures 28(p) - 28(t).

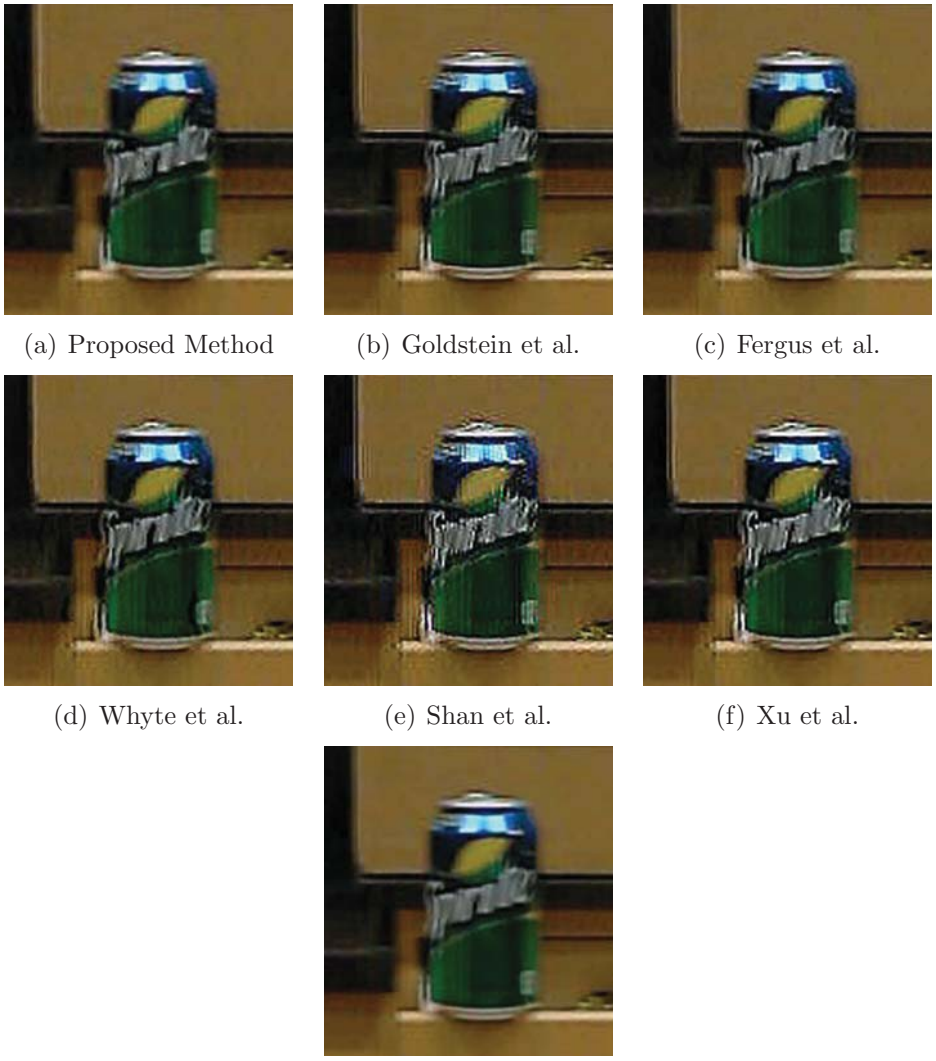
Figure 33 shows another experiment result when the system was capturing a difference scene. Figure 35 shows the estimated PSFs. Similarly to the first case, the intensity profile has a left-shifted center since the motion of the selected image frame was settled to the desired position. The results of Goldstein *etal*., Shan *etal*., and Xu *etal*. again present ringing artifacts and the results are slightly degraded. The methods proposed by Fergus *etal*. and Whyte *etal*. show an improvement but also have a small amount of ringing artifacts. The proposed method shows the best estimation with the least ringing artifacts among all the methods.

(a) Proposed Method (b) Goldstein et al. (c) Fergus et al.

(d) Whyte et al. (e) Shan et al. (f) Xu et al.

(g) Blurry input (h) Obtained blurry image

Figure 33: Test set #10. The obtained blurry image and estimated latent images by various methods. Images are taken at 29fps with Vibration Suppresion input command. (a) - (g) are cropped local images.

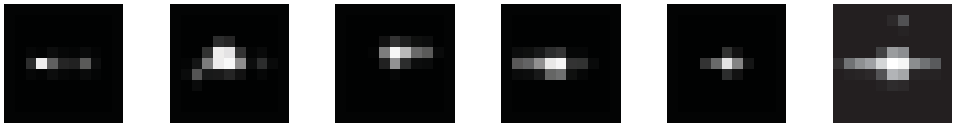


(a) Proposed Method (b) Goldstein et al. (c) Fergus et al.

(d) Whyte et al. (e) Shan et al. (f) Xu et al.

(g) Blurry input

Figure 34: Test set #10. The obtained blurry image and estimated latent images by various methods. Images are taken at 29fps with Vibration Suppresion input command. (a) - (g) are cropped local images.



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

Figure 35: Test set #10. Estimated blur kernels. The size of the PSFs is 21 × 21. (a) Proposed method (b) Goldstein *etal*. (c) Fergus *etal*. (d) Whyte *etal*. (e) Shan *etal*. (f) Xu *etal*.

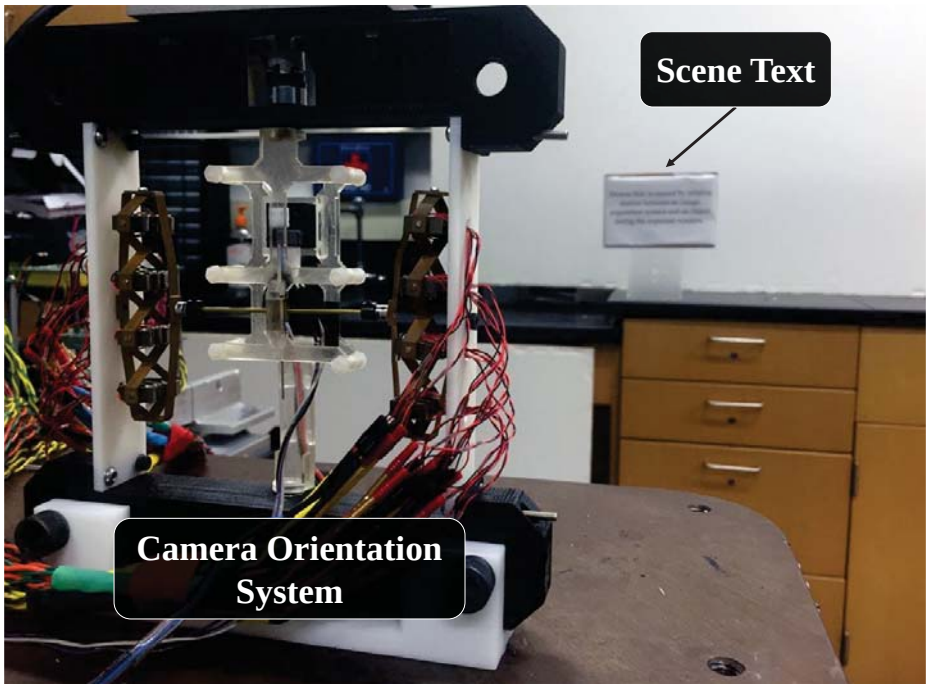
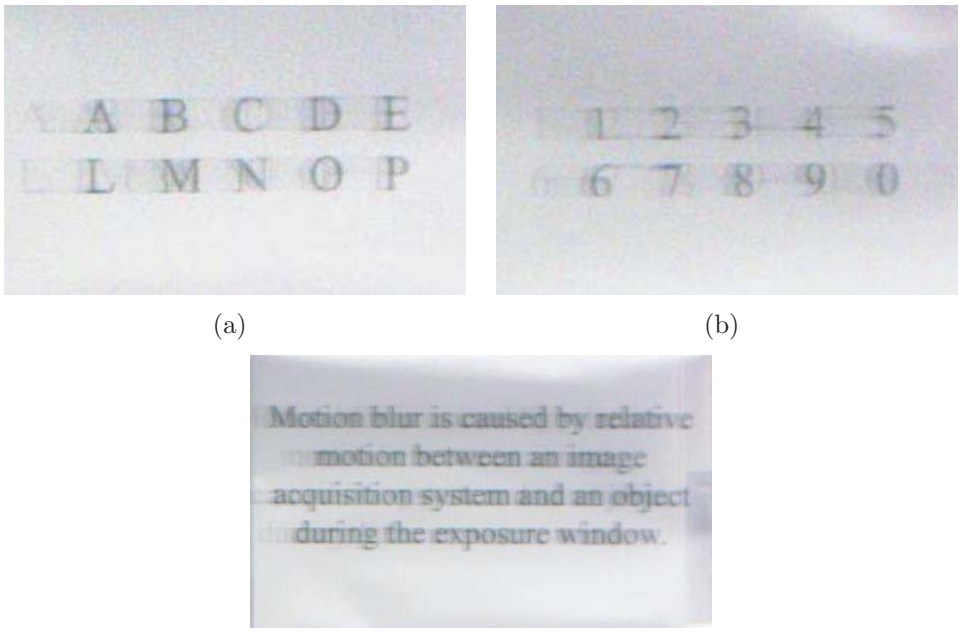


Figure 36: Experimental setup for evaluation by OCR.

3.9 Evaluation by OCR

In the previous sections, the de-blurring methods are quantitatively evaluated by taking the difference between the resultant image and the reference image. In this section, the proposed and compared de-blurring methods are evaluated by optical character recognition (OCR) technology. OCR is a technology that converts text on images into an ASCII code, machined-encoded text.

Figure 36 shows the experimental setup to evaluate the de-blurring methods by OCR. A paper, on which texts are printed, is placed in the scene. A VGA-size image is obtained at 30 fps and the mean angular velocity of the camera orientation system during the exposure is 32.5 deg/sec. The obtained image is de-blurred by the proposed and other methods. Then, OCR is used to evaluate the de-blurring methods by checking text recognition results. In this study, OCR software that is built-in Microsoft OneNote 2010 is used. For evaluation, three different printed texts are used: (1) 10 alphabet characters, (2) 10 numerical digits, and (3) A sentence



(a) (b)

(c)

Figure 37: Input blurry images for evaluating the de-blurring methods by OCR. (a) 10 alphabet characters. (b) 10 numerical digits. (c) A sentence (100 characters).

containing 100 characters. The obtained blurry images are shown in Figure 37.

Figure 38 shows the recognized texts by MS OneNote 2010 OCR from the de-blurred images when Figure 37(a) is used as an input image. Figures 39(a) and 39(b) show numbers of accurate and inaccurate text recognition of each method. The text recognition from the original input image is poor due to blurriness. It can be observed that recognition results from both the proposed and Xu's methods are successful that all characters are captured. However, the recognition result by Xu's method includes one inaccurate dot. Although the de-blurring methods improve the OCR results, texts are partially recognized from the de-blurred images by the other compared methods.

Figure 40 shows the recognized numerical digits from the de-blurred images when Figure 37(b) is used as an input image. Figures 39(a) and 39(b) show numbers of accurate and inaccurate numerical digit recognition of each method. It can be observed that all 10 numerical digits are recognized only from the de-blurred image by the proposed method. Although a recognition rate from the de-blurred image by

Xu's method is high, the OCR result includes many inaccurate machined-encoded texts. No numerical digits are recognized from the de-blurred images by Goldstein, Fergus, and Whyte methods. The OCR result from the image obtained by Shan's method show 3 numerical digits and unwanted ASCII texts.

Figure 42 shows the de-blurred images for a blurry input shown in Figure 37(c) by various methods. Figure 43 shows the recognized texts from Figure 42 by OCR. The recognition rate from a blurry input image is only 5% as it can be observed in Figure 44. It can be observed that the recognition rates are improved using the de-blurring technique. Among various de-blurring methods, the proposed method shows the best text recognition rate. However, there are large discrepancies in the OCR results between the reference image and de-blurred images by the other methods. The experimental results from all cases show that the proposed method exhibits better text recognition results by OCR than the conventional methods.

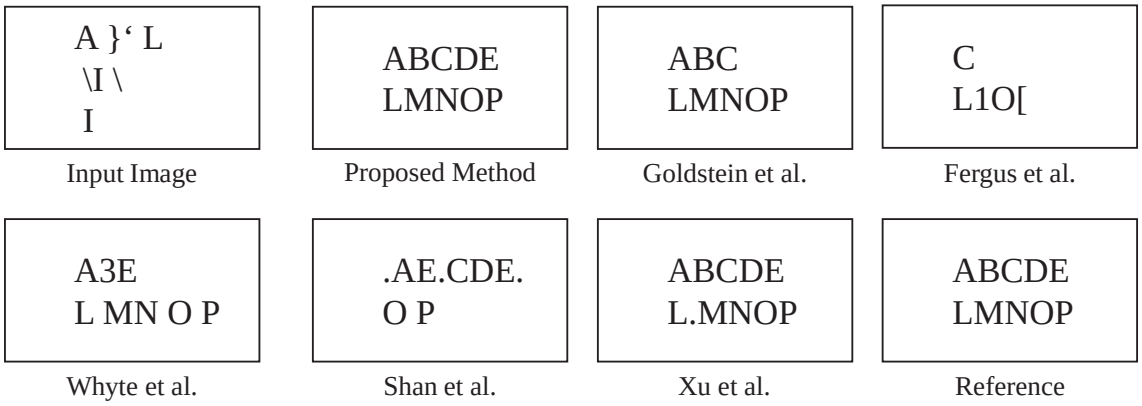
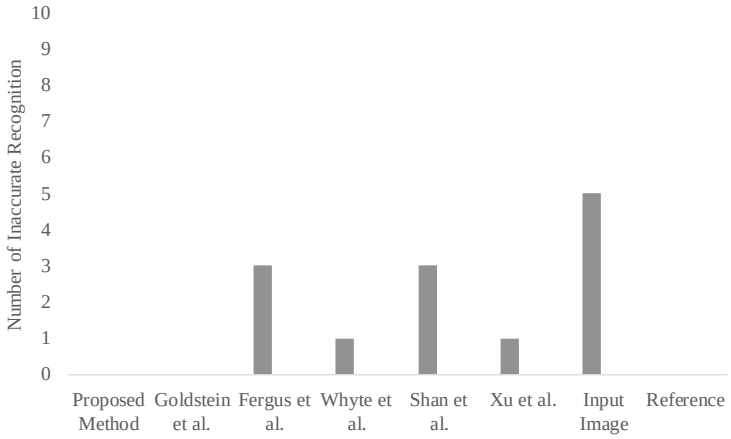


Figure 38: Text recognition results from the de-blurred image. The input blurry image is Figure 37(a).

(a)



(b)

Figure 39: Evaluation of the de-blurring methods by OCR. (a) Number of recognized text. (b) Number of inaccurately recognized text.

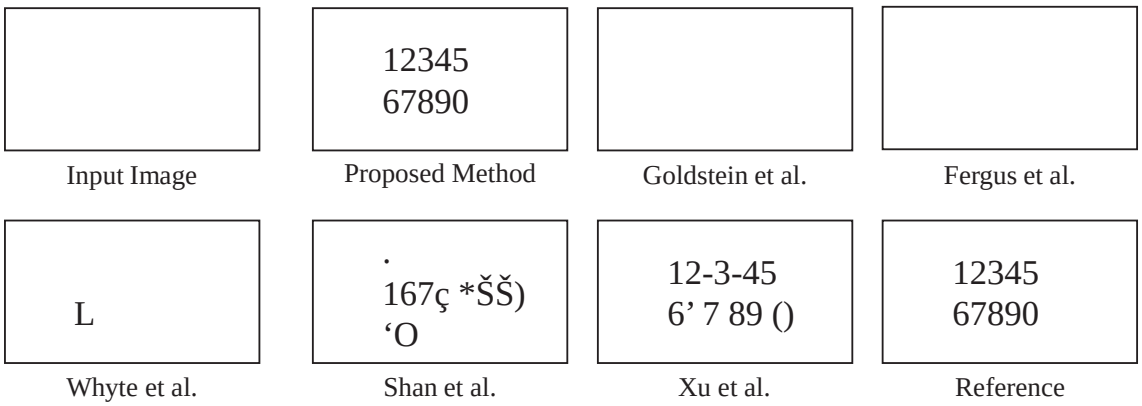
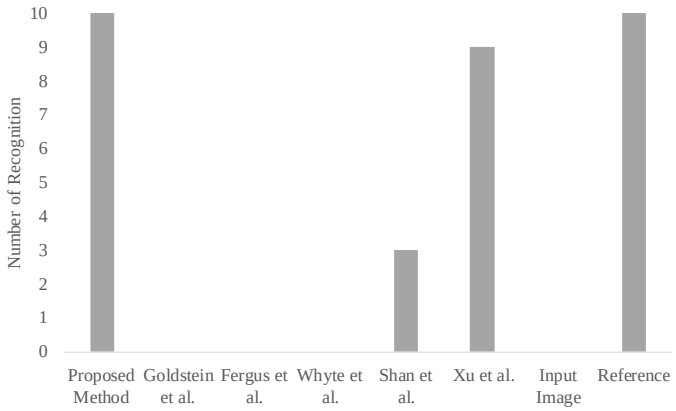


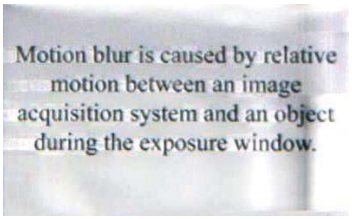
Figure 40: Text recognition results from the de-blurred image. The input blurry image is Figure 37(b).



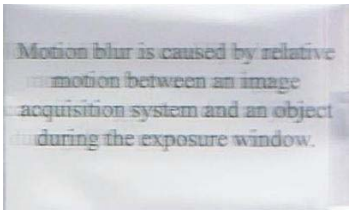
(a)

(b)

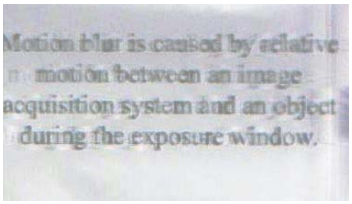
Figure 41: Evaluation of the de-blurring methods by OCR. (a) Number of recognized numerical digit. (b) Number of inaccurately recognized numerical digit.



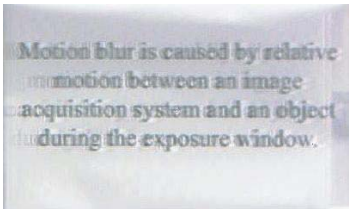
(a) Proposed method



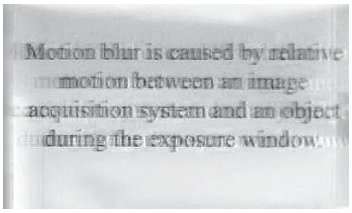
(b) Goldstein *et al.*



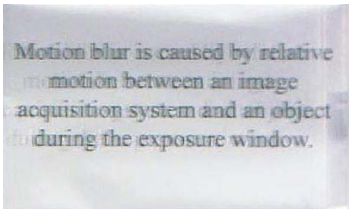
(c) Fergus *et al.*



(d) Whyte *et al.*



(e) Shan *et al.*



(f) Xu *et al.*

Figure 42: De-blurred images of 37(c) by various methods.

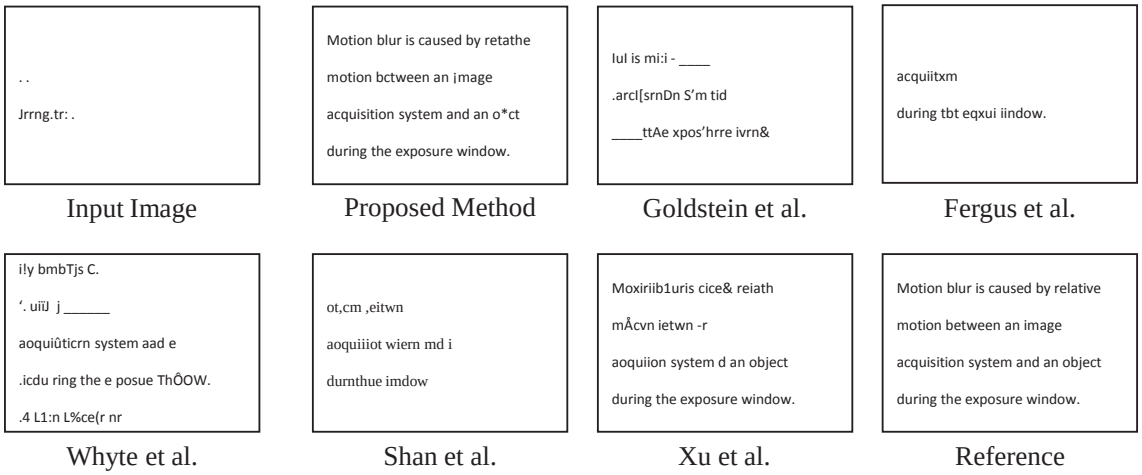


Figure 43: Recognized texts by OCR from images shown in Figure 42.

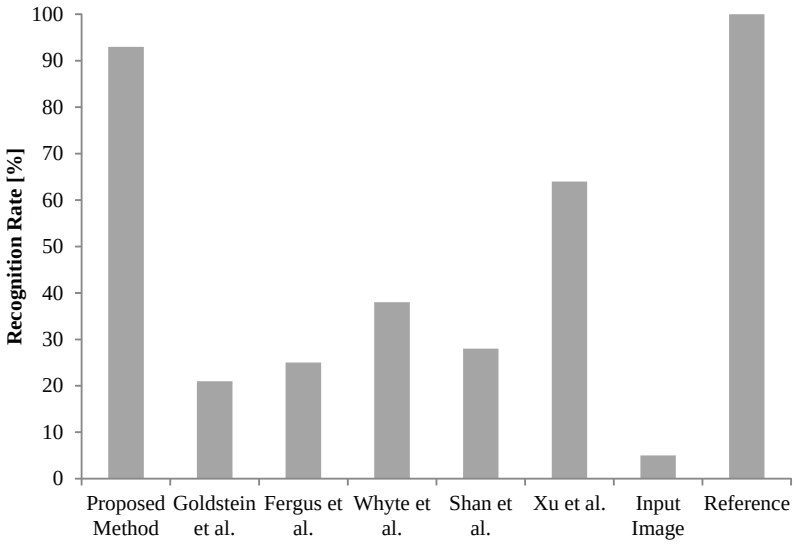


Figure 44: Recognition rates by OCR from the de-blurred images containing a sentence.

3.10 Summary

This chapter presented a method to estimate the spatial-invariant PSF for the robotic camera orientation system [55, 51]. The robotic camera system was given discrete switching commands in an open-loop manner to effectively suppress the vibration. The camera orientation system captured images with relatively small motion blur as a result of the vibration suppression technique. The PSF was estimated in an open-loop manner with the aid of known system dynamics, requiring no external sensors. It was assumed that the system dynamics was well calibrated and the system had no unpredictable disturbances. The latent image was estimated by using the deconvolution process in knowledge of the estimated PSF.

The proposed method is investigated under various motion conditions and scenes to verify the effectiveness and was compared to well-known other approaches. The proposed method shows the best estimation of the PSF overall from the experiment results. The experiment results show that overall the performance metric of the proposed method is 31.29% better than that of conventional methods. The proposed method is also evaluated by OCR technology and the text recognition rates of the proposed method is 57.8% point better than that of conventional methods. In addition, the computation time of the proposed method is 45.1 times faster than the conventional methods. Although the proposed method was tested in the single DOF mechanism, it is also applicable to a multi DOF camera orientation system. Nonlinear dynamics can also be handled since the dynamics calculation is feedforward. The proposed method, however, is limited to the application that the camera is fully controlled by actuators.

CHAPTER IV

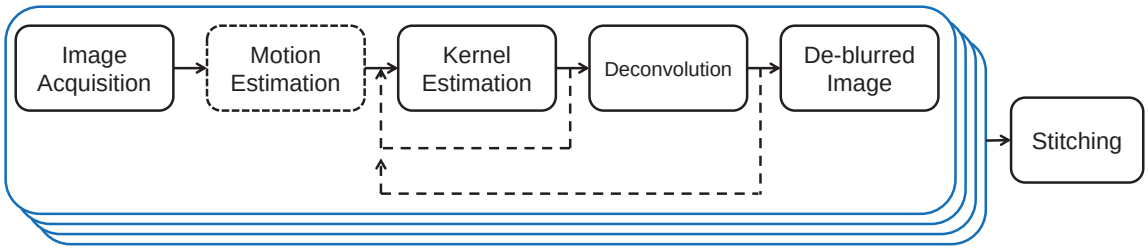
SYSTEM ARCHITECTURE FOR VISUAL-MOTOR COORDINATION AND REAL-TIME PERFORMANCE

This chapter presents a system architecture to achieve real-time performance by visual-motor coordination.

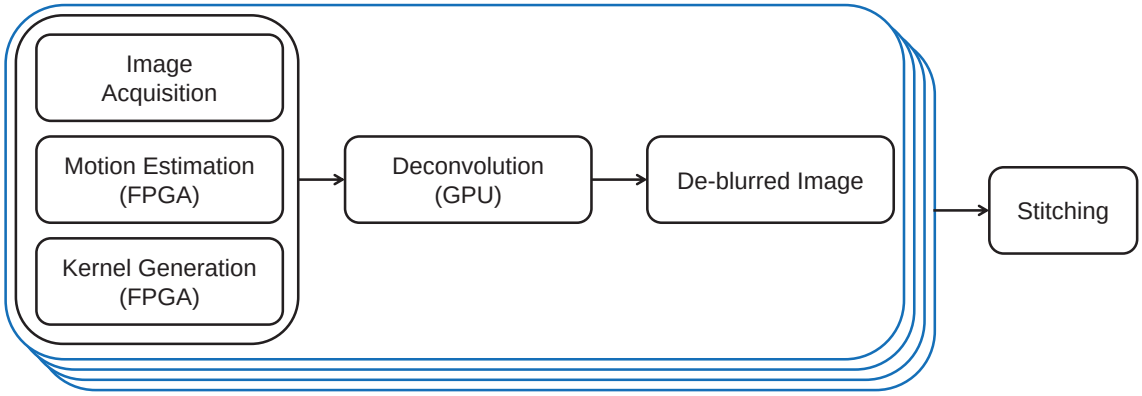
The field of view of a generic camera is limited [84]. This limitation motivates research in panoramic image generation. In classical methods, multiple images are taken at rest with different angles of view to generate a panoramic image. Existing methods are not robust to blurry images due to a lack of information on high frequencies [65]. Due to this reason, images are not obtained in the course of motion.

A panoramic image may be generated while quickly sweeping an environment by visual-motor coordination using the robotic vision system. Although images are obtained in the course of rapid motion, the dynamics-based de-blurring method discussed in the previous chapter can be incorporated to enhance the quality. Recovering high-frequency information of raw images could benefit the algorithm for panorama generation. However, if the processing time is longer than the summation of image acquisition time and positioning time, then incorporating the de-blurring technique has no advantages.

The dynamics-based de-blurring method improved computation times but the implantation was non real-time and limited to a single image. Therefore, the proposed architecture aims to achieve: (1) real-time image de-blurring within 30 fps, and (2) parallel system architecture for visual-motor coordination and accelerating computation time.



(a)



(b)

Figure 45: Comparison of pipelines. (a) Pipeline of conventional de-blurring methods. Image acquisition, motion estimation, and kernel generation are processed in a serial manner. (b) Pipeline of the proposed de-blurring method. Image acquisition, motion estimation, and kernel generation are processed in a parallel manner.

4.1 Methodology and System Architecture

Figure 45(a) shows a general pipeline of the existing de-blurring methods. The processes are performed essentially in a serial manner including iterative loops, which can be an obstacle for achieving the real-time performance. A blurry image is obtained and its blur kernel or PSF is estimated by analyzing the obtained image. Then, non-blind deconvolution is performed to restore the blurry image [62]. Most of the studies update the blur kernel iteratively and some studies even feedback the deconvolution results to refine the final outcome [18, 33]. Sensor-based approaches include the motion estimation process and may update the kernel in an iterative manner [9, 3, 45].

This research proposes a pipeline in which image acquisition, motion estimation, and kernel estimation are performed in a parallel manner using the FPGA as shown in Figure 45(b). The dynamics-based approach has been adopted to estimate the blur kernel [55]. Since it relies only on dynamics and control information, the blur kernel can be estimated from the motion controller on the FPGA in a parallel manner during image acquisition as shown in Figure 45(b). Although this study does not introduce an iterative process, a feedback loop may be included to further improve estimation of the blur kernel.

The previous section demonstrated that the dynamics-based approach can improve computation times; however, the implementation was non real-time and limited to a single image [55] . To further improve the dynamics-based concept, this section proposes a system architecture that exploits parallel computing to achieve real-time image de-blurring and make use of resultant images for panoramic image generation as shown in Figure 46. In order for real-time image processing and control of multiple actuators, the proposed system architecture embodies various processors to increase the parallelism.

4.1.1 Vision Sensor

In order for the dynamics-based approach to estimate the blur kernel in a sensorless fashion, it is critical to accurately locate an exposure window of each image frame. Otherwise, the motion causing the blurry image may not be properly estimated, resulting in a deteriorated de-blurred image. This study deals with motion blur due to rapid motion with regular frame rates as opposed to existing studies that focused on motion blur caused by handshake during a long exposure. Thus, it is recommended to employ a camera that supports GPIO (General Purpose Input/Output). Synchronization is guaranteed by sharing the clock between the camera and motion system or digital signals to communicate exposure window information if GPIO is supported.

Although cameras such as webcams do not support IO capabilities, synchronization could be achieved by software clocks. However, it may not be robust to a communication delay or jitter in sampling clock. It is also recommended to operate a camera in a global shutter mode to avoid the wobble effect. Otherwise, artifacts exist in a rolling shutter mode where all pixels of an image sensor are not registered at once. The wobble effect can be removed by introducing a homographic model [35]. However, minor artifacts may still exist and the computational effort may increase.

4.1.2 FPGA

The FPGA is a 2D integrated circuit that is programmable, reconfigurable, and scalable [71]. Unlike a controller run by an operating system, FPGA programs run at the hardware level. A number of programs can be operated concurrently in a parallel manner because programs are compiled into physical hardware and thus the FPGA allows for parallel operation at a deterministic and fast rate [76]. By taking advantage of this parallelism, the proposed system intensively exploits the FPGA for motion control and partial image processing as shown in Figure 46.

Xilinx Virtex-5 LX 85 having 12960 slices was used to operate required functions at a sampling frequency of 100 kHz. The vision sensor was operated by sharing a clock with the FPGA and thus exposure windows could be monitored. For vibration suppression of the camera positioning mechanism having a total of 32 PZT actuators, activations of the distributed actuators were switched every 0.9 ms at the fastest.

Given a desired motion profile, low-level commands were generated and given to the motor drivers. While controlling the actuator system, the trajectory was estimated by tracking generated low-level commands with mechanical information for each exposure window and thus the PSF was generated in a parallel and sensorless fashion. Each FPGA program adopted a pipelining method to increase the throughput. Since the FPGA is naturally asynchronous with other controllers, the Direct

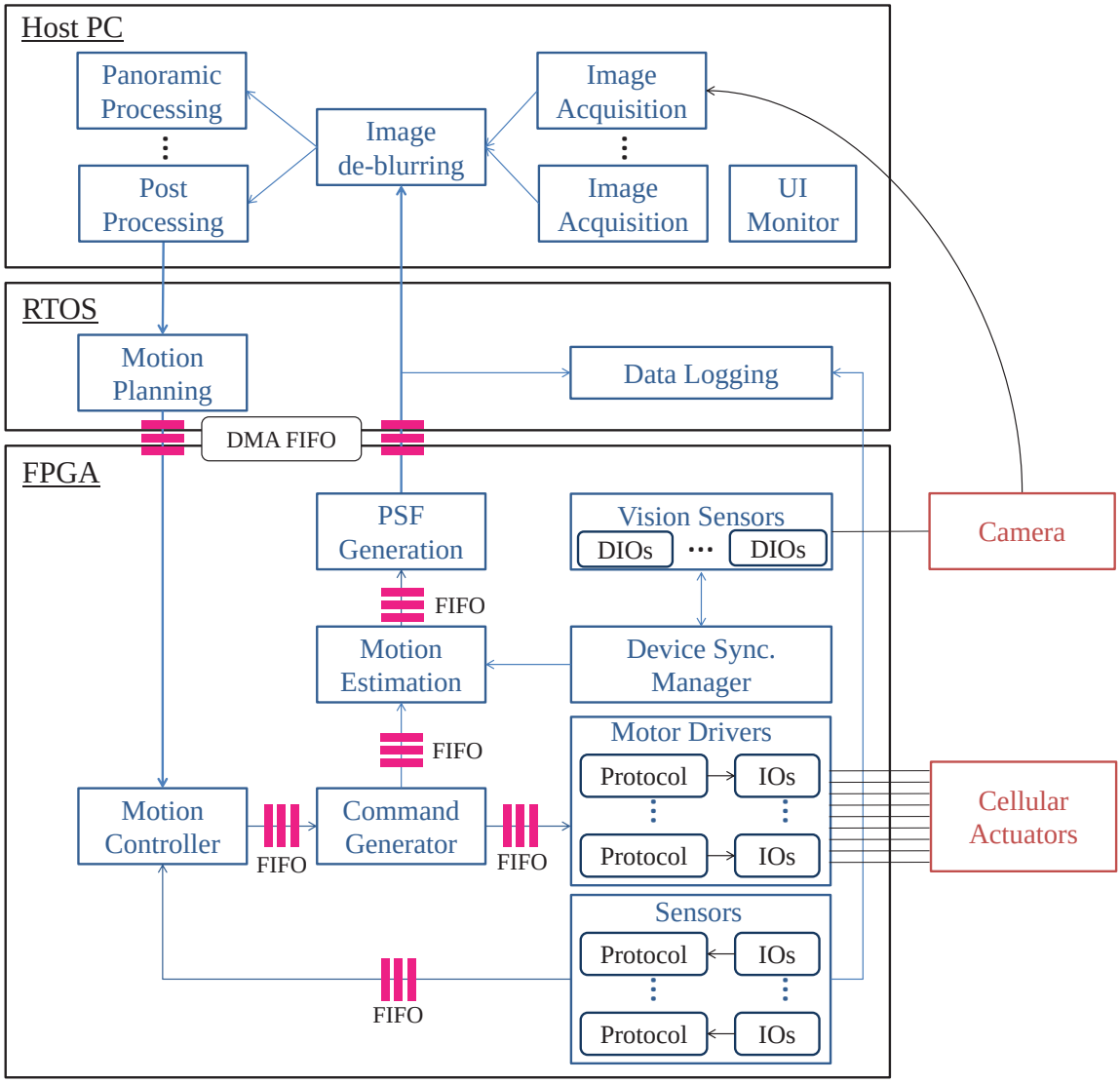


Figure 46: Function blocks and data flows of the proposed system architecture for a general robotic system with the real-time image de-blurring capability. The FPGA is used to control a multi-actuator system and execute various functions in a parallel manner. The GPU is used to enhance process times of matrix computations.

Memory Access (DMA) and First-in First-out (FIFO) methods were used to enable loss-less data transfer at high rates.

4.1.3 GPU

Once the blur kernel was determined, the non-blind deconvolution process was performed by parallel computing to achieve real-time performance. The GPU was dedicated to accelerate the speed of image de-blurring, panoramic generation, and other post-processing. The process speed can be benefited by using a large number of threads with a high memory bandwidth of the GPU. The GPU programs were coded in C/C++ by using NVIDIA's CUDA (Compute Unified Device Architecture). A NVIDIA GeForce GTX 770 graphics card that features 1536 CUDA cores operating at 1046 MHz with 2 GB of memory performed the computations. A desktop computer running Microsoft Windows 7, 64 bit version, with Intel i7-2600K 3.40 GHz processor and 12.0 GB RAM was used.

Since the FPGA was intensively used in this architecture, one might argue that the use of a FPGA may be a better option to accelerate image processing [79] than the use of a GPU. However, previous research reported that GPUs outperformed FPGAs in term of matrix computation [94, 83]. In addition, the use of GPU has the advantage of scalability and compatibility. Also, the CUDA-enabled libraries have been widely available in various platforms [67, 78].

In the proposed architecture, the use of GPU was limited to the deconvolution process since the kernel could be estimated in a parallel manner as shown in Figure 45(b). Otherwise, image acquisition, motion estimation, and kernel generation would need to be executed in a serial manner on the GPU.

4.1.4 Real-Time Controller and Host PC

In a number of robotics applications, a real-time operating system (RTOS) has been used for timing critical tasks such as motion control because of the reliability of deterministic sampling rates [34]. In this architecture, since the majority of computational tasks were executed on either FPGA or GPU, the role of the RTOS was limited to

high-level motion planning and data logging. The role of the host PC except for GPU was for the user interface (UI) and information display.

4.2 Real-Time Image de-blurring

4.2.1 Spatially invariant and deterministic kernel from pure rotational system

Existing studies attempted to estimate a spatially-invariant kernel assuming the uniform motion blur by investigating local regions or by averaging the kernel over the entire image [27]. However, this approach may not be applicable to the case where a camera experiences certain motion with a non-constant depth in scene. Therefore, some studies attempted to estimate a spatially-variant kernel by other means such as using the Microsoft Kinect sensor to obtain the depth map, measuring local blur by image segmentation or introducing a homography model to the camera motion [122, 120]. These existing approaches are effective when motion of a camera involves translational movements. The model of a blur kernel must be carefully chosen based on the condition. In this study, since the camera positioner only generates pure rotational movements like the human ocular system, the spatially-invariant kernel model is adopted.

Figure 47 shows the model of a pinhole camera experiencing rotational motion in the 2D space. FOV is the field of view in the initial camera position and FOV$'$ is the field of view of the camera rotated for θ_x degrees. An arbitrary point on the 2D plane appears on the image plane at I_x, which is a horizontal distance from the image center. The position of the pixel of an arbitrary point for the rotated camera $I_{x'}$ is equivalent to $I_{x1'} + I_{x2'}$. P_x is the horizontal distance of an arbitrary point from the pinhole on the 2D plane. Similarly, the horizontal distance of an arbitrary point from the pinhole for the rotated camera $P_{x'}$ can be expressed as $P_{x1'} + P_{x2'}$. The depths of an arbitrary point at the initial and rotated positions are d and d', respectively.

From the geometry shown in Figure 47,

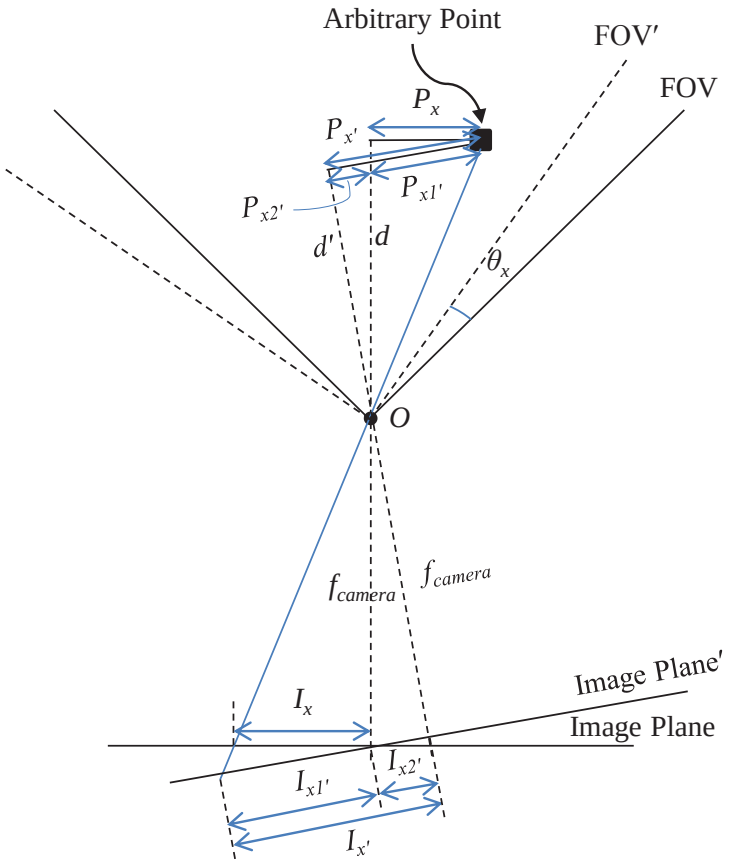


Figure 47: Pinhole camera model for rotational motion

$$I_{x1'} = \frac{f_{camera}}{d'} P_{x1'}, \tag{16}$$

where f_{camera} is the focal length, and

$$P_{x1'} = \frac{P_x}{\cos(\theta_x)}. \tag{17}$$

Also,

$$I_{x2'} = f_{camera} \tan(\theta_x). \tag{18}$$

Therefore,

$$\Delta I_x = f_{camera} P_x \left(\frac{1}{d' \cos(\theta_x)} - \frac{1}{d}\right) + f_{camera} \tan(\theta_x). \tag{19}$$

If a scene or object is not positioned close enough to the pinhole, it may be assumed that $d \approx d'$. Since the motion blur in this study is mainly caused by rapid motion within a short exposure time, as opposed to existing studies, it may also be assumed that $\cos(\theta_x) \approx 1$. If the camera is rotated for 10 degrees, approximately a half stroke of the camera orientation system, an error of 1.52% error would be induced. This error is not significant in this study for VGA and UVGA images since the corresponding error on the blur kernel is even smaller than the size of a single pixel. As a consequence, a constant depth can be assumed even if the scene or object is close to the camera. From these assumptions, (19) can be simplified as:

$$\Delta I_x = f_{camera} \tan(\theta_x). \tag{20}$$

Therefore, a spatially invariant kernel model was adopted for the robotic camera positioner that generates pure rotational motions. In addition, unlike existing studies require user inputs, the dimension of the blur kernel in the horizontal direction k_x can be determined by

$$k_x = f_{camera} \tan(\theta_x) s_x \tag{21}$$

where s_x is the pixel size of the image sensor in the horizontal direction.

4.2.2 Instant dynamics-based blur kernel estimation

In the FPGA, the angular displacement of the camera positioner, $\boldsymbol{\theta} = \theta_x \hat{x} + \theta_y \hat{y}$, at each sample was estimated by using the dynamics-based method reported in [55]. By sampling the estimated motion, the trajectory of the blur kernel can be represented as

$$\overrightarrow{\boldsymbol{k}} = k_x\hat{x} + k_y\hat{y}, \tag{22}$$

where $\hat{x}$ is the unit vector in the horizontal direction and $\hat{y}$ is the unit vector in the vertical direction. The intensity along with the profile at each sample is determined as

$$K_{(k_x,k_y)} = n_{k_x k_y}, \tag{23}$$

where $K_{(k_x,k_y)}$ is the pixel value of the blur kernel $\boldsymbol{K}$ at the location (k_x, k_y), $n_{k_x k_y}$ is the number of samples remaining at the pixel (k_x, k_y) within a single exposure window. The resultant blur kernel $\boldsymbol{K}$ is a $o \times p$ matrix, where o is the height of the blur kernel, and p is the width of the blur kernel. After the exposure window for a single frame was closed, to preserve information of the initial position, this blur kernel $\boldsymbol{K}$ was enlarged to a $(2q+1) \times (2q+1)$ matrix by locating the starting point at the center $(q+1, q+1)$ where $q = \max(o, p)$. The newly introduced pixels were padded by zeros.

Then, the summation of elements of the blur kernel was normalized to one for intensity conservation:

$$\sum_{u=0}^{2q+1} \sum_{v=0}^{2q+1} K_{(u,v)} = 1. \tag{24}$$

Since the programs on the FPGA used for kernel estimation run at 100 kHz, the PSF was generated in $30\mu s$ after an image was obtained. This is the fastest kernel estimation reported and was not feasible by using existing image-based methods or sensor-based approaches [9, 3, 45]. These existing methods employ an additional vision sensor and therefore require additional image processing.

The normalization of the blur kernel was performed on the GPU to decrease the data transfer time. The data type of integer can be used for the blur kernel if not

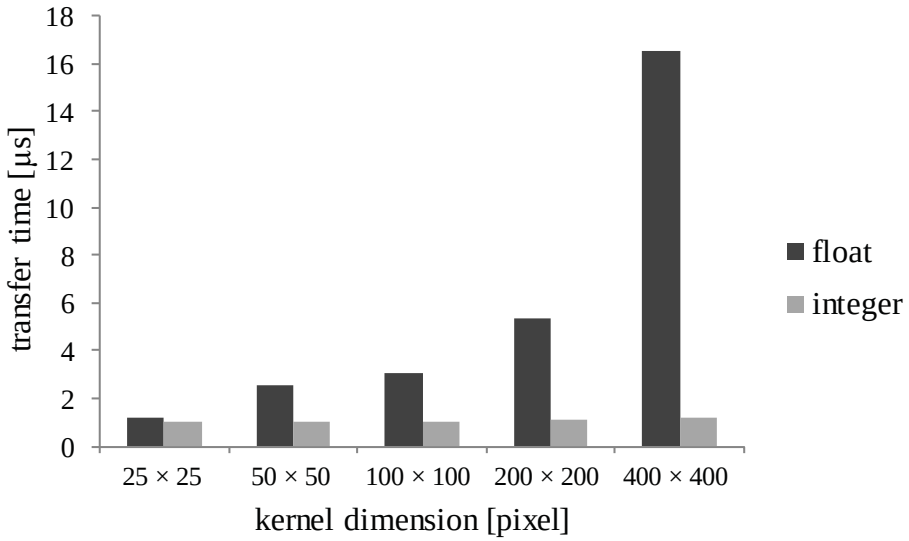


Figure 48: Comparison of data transfer times per pixel from CPU to GPU for various kernel dimensions

normalized by (24) while the data type of float must be used when normalized. Figure 48 shows a comparison of transfer times of a blur kernel per pixel from the CPU to GPU for various kernel dimensions. The use of integer data type was always faster than the float data type because of a smaller memory size. In addition, the transfer times using the data type of float significantly increased as the kernel dimension increased while the times were almost constant for the integer data type. If the maximum value of the blur kernel elements exceeded 255, the kernel was equalized ranging from 0 to 255 and floating points were rounded to the nearest integer to minimize the memory size.

4.2.3 Real-Time Deconvolution

The Richardson-Lucy approach is known to be robust in the presence of high noise levels. However, it is an iterative method that increases the computation time. Therefore, the real-time performance may not be feasible for large-size images. On the other hand, the Wiener method is known to be faster than the Richardson-Lucy method. It estimates a latent image in the frequency domain in a non-iterative fashion. This

method is less robust than the Richardson-Lucy method to the existence of noise as shown in Figure 49 [101]. In this research, both Richardson-Lucy and Wiener deconvolution methods were used and selection of the method was dependent on the input image size and light conditions. [89, 81].

The objective of the Wiener deconvolution is to estimate $\hat{\boldsymbol{I}}(t)$ that minimizes the mean square error of $\boldsymbol{I}$(t) [81]:

$$\hat{\boldsymbol{I}} = \boldsymbol{g} \otimes \boldsymbol{B} \tag{25}$$

where $\boldsymbol{g}$ is defined by the Wiener filter in the frequency domain as

$$\boldsymbol{G}(f) = (\boldsymbol{H}^*(f) \odot \boldsymbol{S}(f)) \oslash (|\boldsymbol{H}(f) \odot \boldsymbol{H}(f)| \odot \boldsymbol{S}(f) + \boldsymbol{N}(f)) \tag{26}$$

$$\boldsymbol{G}, \boldsymbol{H}, \boldsymbol{S}, \boldsymbol{N} \in \mathbb{R}^{m \times n}$$

where $\boldsymbol{G}(f)$ is the Fourier transform of $\boldsymbol{g}$ at the frequency of f, $\boldsymbol{H}(f)$ is the Fourier transform of $\boldsymbol{K}$ at the frequency of f, $\boldsymbol{H}^*(f)$ is the complex conjugate of $\boldsymbol{H}(f)$, $\boldsymbol{S}(f)$ is the mean power spectral density of $\boldsymbol{I}(t)$, and $\boldsymbol{N}(f)$ is the mean power spectral density of $\boldsymbol{N}$.

From (25),

$$\hat{\boldsymbol{I}}(f) = \boldsymbol{G}(f) \odot \boldsymbol{B}(f) \tag{27}$$

where $\hat{\boldsymbol{I}}(f)$ is the Fourier transform of $\hat{\boldsymbol{I}}(t)$. Therefore, the latent image can be found by taking the inverse Fourier transform of $\hat{\boldsymbol{I}}(f)$. Although the dimension of the blur kernel in the frequency domain is the same as that of the obtained image, it is a faster method than the Richardson-Lucy method.

Since a spatially-invariant kernel was used, a color image consisting of three channels was treated as a single channel image as shown in Figure 50 when performing the Richardson-Lucy deconvolution process on the GPU. This approach reduced the

(a) (b) (c)

Figure 49: Comparison between Richardson-Lucy and Wiener deconvolution methods. The image was taken under low light condition to introduce noise and to test robustness. The results shows that the Richardson-Lucy method is robust to noise while the Wiener method is not. (a) Original blurry input (b) Result obtained by the Richardson-Lucy method (c) Result obtained by the Wiener method

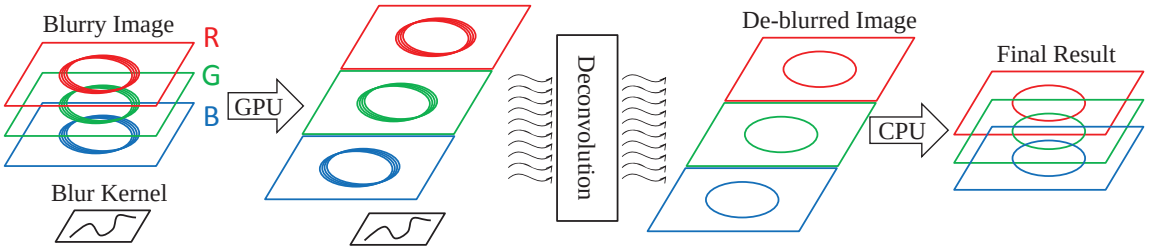


Figure 50: Single channel approach when performing the deconvolution process on the GPU to improve computation time.

computation time by 5.06% and 6.57% for UVGA and VGA size images respectively as shown in Table 5. In contrast, no significant change was observed for the Wiener deconvolution method.

Table 5: Improvement in computation time by single channel approach

Image size	three channels [msec]	single channel [msec]
VGA	92.9	88.2
UVGA	27.72	25.9

4.3 Experimental Results

Tables 7 and 8 show comparisons of computation times of image de-blurring in conditions given in Table 6. An existing de-blurring algorithm proposed by Xu *et al.* [121] was chosen as it is known to be efficient, widely used for benchmarking purposes, and

supports GPU computation. A code distributed by the author was used. The conventional algorithm include free parameters that need to be tuned by a user. These parameters were adjusted manually until the best possible results were obtained.

In addition, UVGA-size images were decreased to the SVGA size when running Xu's method due to software limitations. Therefore, the computation time of these methods would be longer for UVGA size images.

Table 6: Conditions of the test sets

Set	Image	Kernel	velocity (deg/sec)
#1	1280 × 960	35 × 35	16.25
#2	1280 × 960	75 × 75	30.47
#3	640 × 480	41 × 41	49.99
#4	640 × 480	31 × 31	37.49

Table 7: Comparisons of Computation Times using GPU for Color Images [ms]

Set	Proposed (R-L)	Proposed (Wiener)	Xu *et al.*
#1	52.1	46.8	2519†
#2	88.2	47.7	2214†
#3	26.7	16.2	1710
#4	25.9	16.1	1986

† SVGA image was used due to software limitation

Table 8: Comparisons of Computation Times using GPU for Grayscale Images [ms]

Set	Proposed (R-L)	Proposed (Wiener)	Xu *et al.*
#1	19.3	19.8	2687†
#2	30.9	19.9	2233†
#3	9.8	9.9	1713
#4	9.4	9.9	1725

† SVGA image was used due to software limitation

As can be observed in Tables 7 and 8, the proposed method provides the fastest computation time. Real-time performance has been achieved for VGA-size color images regardless of the deconvolution method. Although real-time performance for

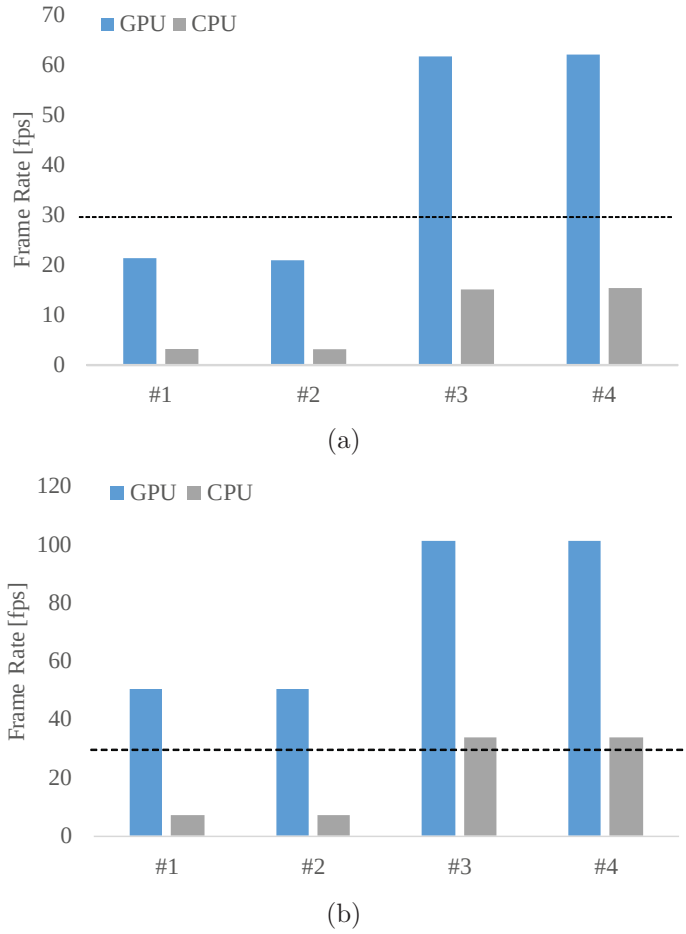


Figure 51: Improvement in the computation times. (a) Color images (b) Grayscale images

UVGA-size color images could not be achieved by using the Richardson-Lucy method, it was close to real-time using the Wiener method. It can also be observed that the kernel size was not the major factor affecting the computational time in the Wiener method. On the contrary, the computational time of the Richardson-Lucy method is highly dependent not only on the image size but also the kernel size. Although all results are far from the real-time performance, it can be confirmed that the dynamics-based approach provides the fastest results.

In general, the main part of computation time in de-blurring is occupied by kernel estimation. Although the method by Xu *et al*. has shown notable results among image-based de-blurring algorithms as shown in Figure 26, the computation time

was significantly longer than the proposed method as shown in Table 8. It includes nested iterative loops for kernel estimation resulting in an extensive computation. The method by Cho *et al*. has also shown notable results by using the CPU. However, the real-time performance may not be feasible using GPU-based acceleration according to their paper [18].

4.4 Summary

This chapter presented a system to control distributed PZT actuators and achieve real-time performance within 30 fps for image de-blurring. The experimental results show that the real-time performance for image stitching within 30 fps was achieved when VGA-size images and UVGA-size grayscale images were used. Although the frame rate was slightly below 30 fps for UVGA-size color images, real-time performance may be achieved by using an advanced GPU or adopting multi-GPU system.

This chapter validated the hypothesis that the key enabler for real-time visual processing in a fast moving vision system is utilizing dynamics information and parallel computing; a neural mechanism utilizes the dynamic model of the eye to predict self-induced eye movements and compensate motion smear. The preliminary research in this chapter provided the basis of rapid panorama generation by visual-motor coordination that will be explained in Chapter 5.

CHAPTER V

DYNAMICS-BASED PANORAMA GENERATION

This chapter presents a fast panoramic generation method using a rapid-moving robotic vision system. It is realized by developing a dynamics-based method for homography estimation using a robotic vision system. For panoramic image stitching, most of existing algorithms [28, 31, 39, 66] are designed to process clean images, resulting in limited robustness to motion blur. The speed of motion in traditional dynamic camera platforms with servomotors is often bounded to avoid blurry images. The use of a high framerate camera for robotic vision is an alternative practical approach. Most of existing high-speed vision systems employ a camera with a high frame rate, e.g., 200-1000 fps, mounted on a standard positioning mechanism [16, 125]. However, applications may be limited by the requirement of a bright light condition as well as the cost and size of high framerate cameras.

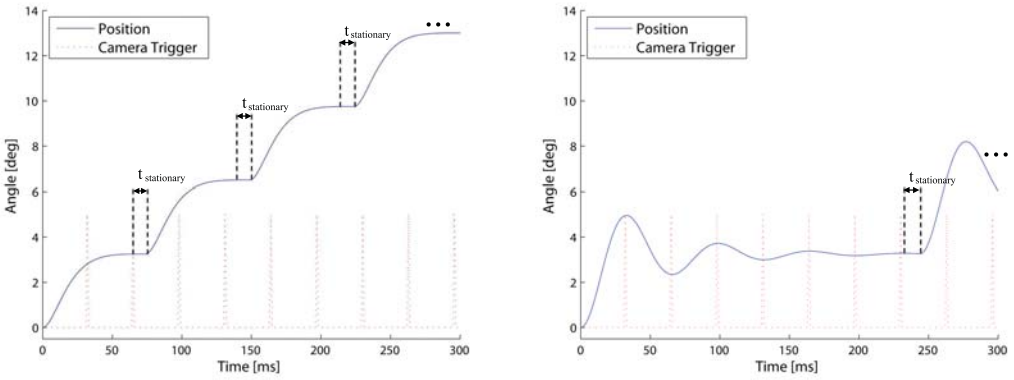
In this research, a camera with a standard frame rate is used and images are acquired from different angles of view while the robotic camera positioner quickly scans the environment. The obtained images may be blurry due to the fast motion of the camera positioner. An image with an enlarged FOV will be generated based on dynamics-based homography estimation, which is robust to degraded images. The resultant image may be improved in a parallel fashion by incorporating the real-time de-blurring method discussed.

5.1 Visual-Motor Coordination

This section demonstrates the importance of the coordination between motion control and image processing for quick scanning of the field of view. While the proposed dynamics-based image processing technique is solely useful, the method is even more

effective when combined with the vibration suppression technique in terms of scanning time. If the system needs to scan multiple positions, the total acquisition time is dependent on both mechanical settling time and $t_{stationary}$. The time $t_{stationary}$ is the difference between the time that the new commands are given and the time the motion generated by previous commands was settled in a single exposure window as shown in Figure 52. $t_{stationary}$ is minimally the exposure time of the camera if no motion was involved. The mechanical settling time is dependent on the natural frequency and can be considered as a constant. Therefore, the total acquisition time can be adjusted by $t_{stationary}$. A longer $t_{stationary}$ would result in less motion blur and vice versa. Since the degree of motion blur is dependent on the total acquisition time, there is a trade-off between a scanning time and an image quality.

Figure 53 shows the experimental results when the camera orientation device scans four different equally spaced positions. The proposed method was only used to estimate the latent image because its advantage has already been verified in Sec. 3.6. A total of four trials were conducted for each case and the total travel distance was 16 PZT steps. The results show that the completion times are faster overall when the vibration suppression commands are given. In comparison, the same quality of



(a) (b)

Figure 52: Visual-motor coordination (a) Vibration suppression input (b) Step input

estimated latent images cannot be achieved in the same time by the step input, or vibration uncompensated input. When the vibration suppression commands are given, standard deviation of the estimated image quality decreases as the completion time increases. However, this tendency is not clear when the vibration uncompensated command is given. Therefore, the results show that the vibration suppression technique is essential for the scanning time, even if the proposed de-blurring method is applicable for any type of commands. Also, quantitative evaluation supports the notion that the vibration suppression technique contributes to the image quality.

By means of visual-motor coordination, incorporating the dynamics-based motion de-blurring method with vibration suppression technique allows environmental scanning at a fast rate while achieving acceptable image quality.

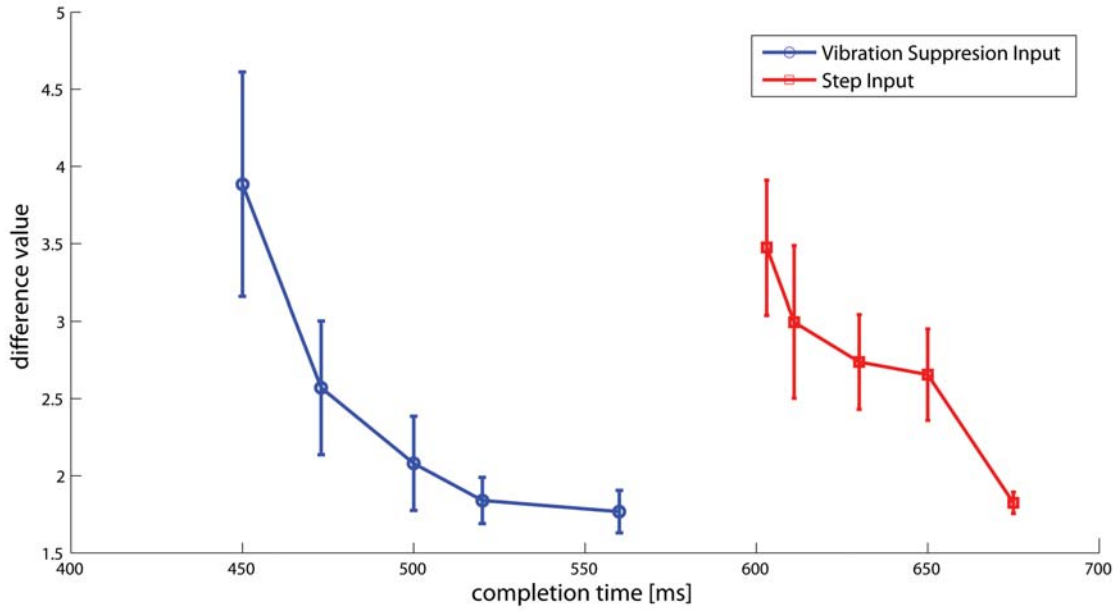


Figure 53: Results of visual-motor coordination. The camera orientation device scans four different positions. The total travel distance is 16 PZT inputs.

5.2 Dynamics-based Stitching

The effectiveness of the vibration suppression commands to generate saccade-like movements for visual-motor coordination was presented in the previous section. A

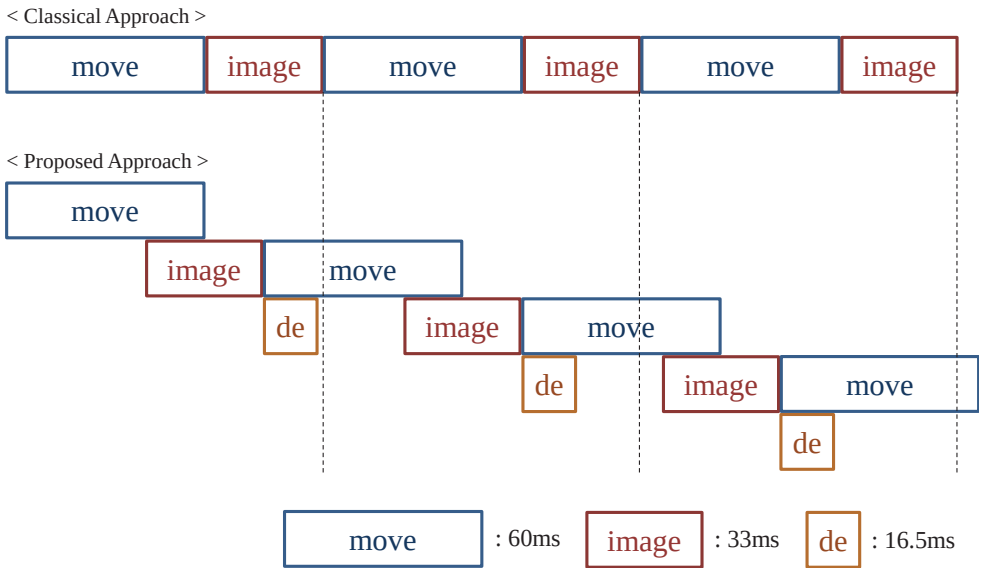


Figure 54: Timeline to create a panoramic image from VGA images. The 'move' block represents the settling time for a point-to-point motion using the robotic vision system. The 'image' block represent the time of image acquisition. The 'de' block represents the computation time of image de-blurring that set to 16.5 ms based on the results from section 4.3.

panoramic image can be generated by acquiring multiple images while the robotic vision system quickly sweeps and scans the environment as shown in Figure 54. To generate an enlarged FOV, the camera may acquire images with respect to the position of the camera orientation system rather than acquiring images at the full frame rate for feeding video. The total scanning time can be decreased by adjusting time lengths at rest. It should be noted that there is a trade-off in quality as shown in Figure 53. To relate neighbor images to each other, a homography matrix should be obtained to warp images on the same plane.

5.2.1 Homography Estimation

This section the dynamics-based approach for the estimation of the homography matrix. The camera positioner presented in this study produces pure rotational motion and thus the homography matrix can be given as [111]:

$$\boldsymbol{H} = \boldsymbol{C}_{camera}\boldsymbol{R}\boldsymbol{C}_{camera}^{-1} \tag{28}$$

$$\boldsymbol{H} \in \mathbb{R}^{3\times 3}$$

where $\boldsymbol{C}_{camera}$ is the camera intrinsic matrix and $\boldsymbol{R}$ is the rotation matrix. The rotation matrix is a function of a static angle $\bar{\boldsymbol{\theta}}$:

$$\boldsymbol{R} = \boldsymbol{R}(\bar{\boldsymbol{\theta}}). \tag{29}$$

The static angle $\bar{\boldsymbol{\theta}}$ is equivalent to the center of gravity of the blur kernel that was estimated on the FPGA. The center of gravity of the i-th blur kernel $\boldsymbol{K}_{cog,i} = \begin{bmatrix} K_{cog,x,i} & K_{cog,y,i} \end{bmatrix}^T$ is located at

$$\boldsymbol{K}_{cog,i} = \sum_{u=0}^{2q+1}\sum_{v=0}^{2q+1} K_{(u,v),i} \begin{bmatrix} u \\ v \end{bmatrix}, \tag{30}$$

where $K_{(u,v),i}$ is the value of the i-th blur kernel at the pixel (u, v).

Therefore, the static angle of the i-th frame for transformation $\bar{\boldsymbol{\theta}}_i = \begin{bmatrix} \bar{\theta}_{x,i} & \bar{\theta}_{y,i} \end{bmatrix}^T$ is

$$\bar{\theta}_{x,i} = \arctan(\frac{K_{cog,x,i}}{f_{camera}s_x}) + \theta_{x,i0} \tag{31}$$

$$\bar{\theta}_{y,i} = \arctan(\frac{K_{cog,y,i}}{f_{camera}s_y}) + \theta_{y,i0} \tag{32}$$

where s_y is the pixel size of the image sensor in the vertical direction and $i0$ denotes the initial position of the i-th frame.

5.2.2 Modified Deconvolution

The real-time de-blurring method was presented in the previous chapter but the results tend to have ripple artifacts around edges as shown in Figure 55(b) due to zero padding when computing convolutions. This artifact may not be significant when de-blurring a single image. However, these ripples could exist in multiple regions in

(a) (b) (c)

Figure 55: Deconvolution results. The original method has artifacts around the edges due to zero padding. (a) Original blurry input. (b) Obtained by general deconvolution. (c) Obtained by modified deconvolution.

a panoramic image or confuse a conventional image-based panoramic algorithm. The artifacts could be neglected by simply trimming the them after the deconvolution process but it conflicts with the purpose of a panoramic image.

To avoid this issue, the input image was padded with flipped images as shown in Figure 56. The dimension of the flipped image is:

$$m_{pad} = \begin{cases} k_x \cdot n_{RL}, & \text{Richardson-Lucy Method} \\ k_x, & \text{Wiener Method} \end{cases} \tag{33}$$

$$n_{pad} = \begin{cases} k_y \cdot n_{RL}, & \text{Richardson-Lucy Method} \\ k_y, & \text{Wiener Method} \end{cases} \tag{34}$$

where m_{pad} and n_{pad} are the width and height of the flipped image, respectively, and n_{RL} is the total number of iteration in the Richardson-Lucy method.

The artifact can be avoided by padding each side of the input image with a flipped image as shown in Figure 55(c) [50]. This padding approach results in a longer computation time due to the increased dimension of the input image.

5.3 Experimental Setup

Figure 57 shows the experimental setup for generating a panoramic image by visual-motor coordination. The times of 'move', 'image', and 'de' blocks in Figure 54 were set to 60 ms, 33 ms, and 16.5 ms, respectively, based on the results from Section 4.3.

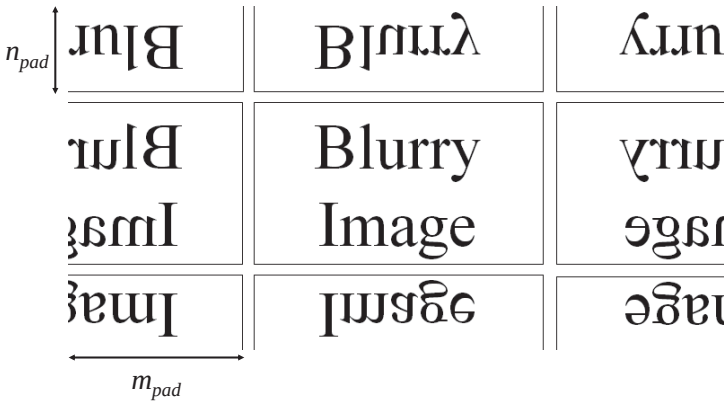


Figure 56: An input image padded with flipped images to avoid ripple artifacts.

The proposed method generates a panoramic image solely based on the dynamics of the camera positioner. Note that a slight non-linearity was observed when the camera passed the neutral position where both the cellular actuators were activated. Since the proposed method is purely based on the dynamics, encoder data was used for adjusting only when the camera passed the neutral position. In other motion ranges, encoder information was not used when only a single cellular actuator was used to position the camera and no significant non-linearity was observed. Since the method is independent on image features, it should be robust to degraded blurry images and does not require overlapped regions between neighboring images.

5.4 Experimental Results

5.4.1 Evaluation

Figures 58, 59 show the panoramic images generated by the proposed method and existing methods. For comparisons, Microsoft (MS) Image Composite Editor (ICE) [73], PTGui [77], and Panoweaver [25] programs were chosen. The robotic camera system scanned the environment to the right and left facing to the object and a total of 17 images were captured in the motion. The size of image was VGA and the mean angular velocity during the single exposure window was 36.1 deg/sec. For the proposed method, a newly obtained image was stitched on the top of the existing panoramic

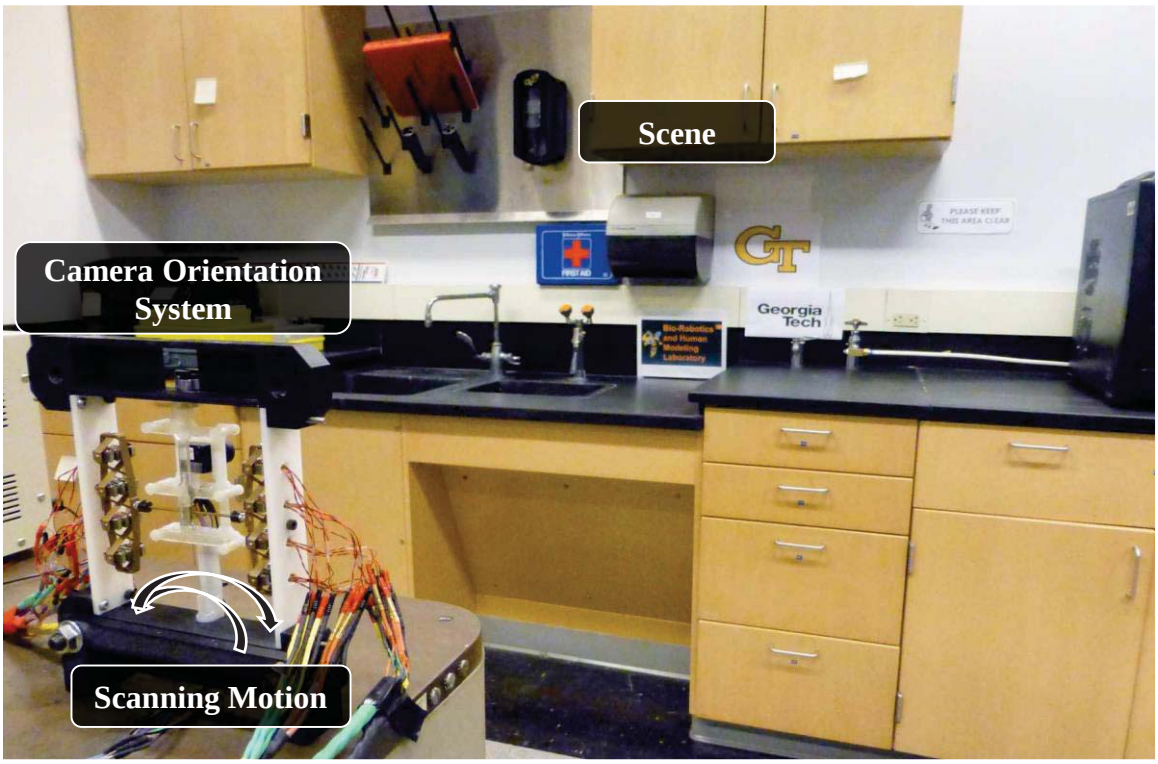


Figure 57: Experimental setup for generating a panoramic image while quickly scanning the environment by visual-motor coordination.

image. For the conventional methods, the entire image set was given to the program at once. Otherwise, errors would be propagated when stitching the latest image on the existing panoramic image, resulting in incompletion of the algorithms. In addition, two additional *clean* images obtained at rest were given for the other methods due to failure. A panoramic image generated from clean images at rest is also shown in Figure 58(c) for reference.

It can be seen that the proposed method generates stitching results with no geometric distortion or misalignment as shown in Figure 58(b). In addition, the output image is enhanced by incorporating real-time image de-blurring as shown in Figure 58(a). The result exhibits minor seams since no additional image processing was applied such as blending.

The panoramic image generated by MS ICE also shows notable results and the degree of the blurriness is improved due to blending the additional *clean* images.

(a)

(b)

(c)

Figure 58: Panoramic results of the proposed method. (a) Dynamics-based method incorporating with real-time image de-blurring. (b) Dynamics-based method incorporating without image de-blurring. (c) Clean static image for reference.

(a)

(b)

(c)

Figure 59: Panoramic results of the compared methods. (a) MS ICE (b) PTGui (c) Panoweaver

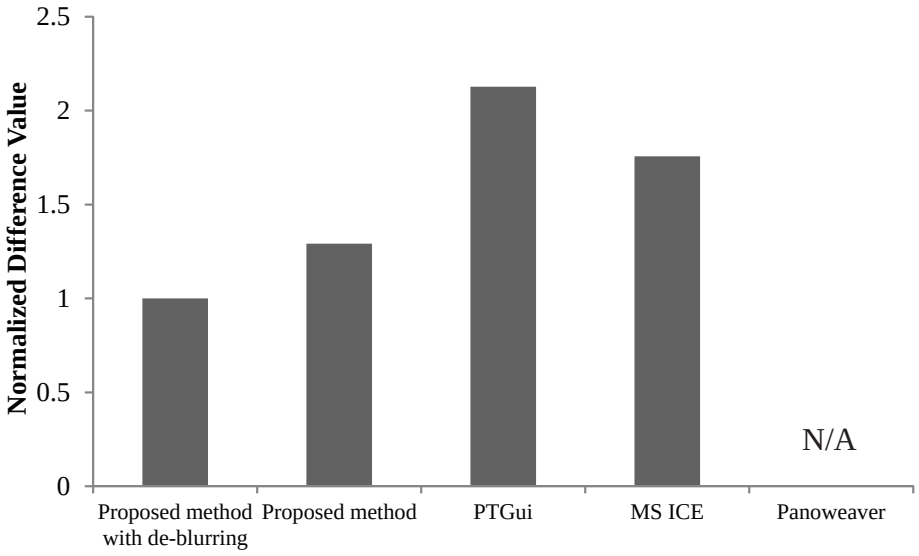


Figure 60: Quantitative evaluation of the panorama results.

However, geometrical misalignments are observed in particular on the right side. In addition, the entire image is slightly curved or distorted. The panoramic image generated by PTGui exhibits substantial geometrical misalignments in the bottom left area. The panoramic image generated by Panoweaver exhibits substantial geometrical misalignments and the left area is identical to the additional *clean* images.

Figure 60 shows a quantitative evaluation of the image stitching results. The difference value was obtained by averaging the difference of pixel values between the resultant panoramic image and the panoramic image generated from clean images. For a fair comparison, the reference image was generated by using the same method and then warped. Therefore, lower scores indicate better results. The results show that the performance metric of the proposed method is 41% better than the conventional algorithms on average. It should be noted that the current proposed method is limited to a robotic vision system that produces pure rotation motion. Also, the dynamics-based approach requires an accurate system model and it is not completely robust to external disturbances.

5.4.2 Computation Time

Tables 9 and 10 show that real-time image stitching can be achieved when VGA-size image are given. The real-time performance was achieved for both GPU and CPU computations. The proposed dynamics-based method provides a faster computation time than other methods.

For UVGA-size images, the performance was close to real-time with 30 fps by taking advantage of the GPU-based acceleration. Similarly, the computation times of the proposed method were faster than those of the conventional methods. Contrary to image deconvolution, the computation time was not significantly improved by massive parallelism because the processes did not involve a number of matrix computations.

Table 9: Comparisons of Stitching Times per Image (GPU Computation)

	VGA		UVGA	
Method	Time [ms]	fps	Time [ms]	fps
Proposed	11.1	90.1	36.8	27.17
PTGui	168.1	5.95	260.9	3.83

Table 10: Comparisons of Stitching Times per Image (CPU Computation)

	VGA		UVGA	
Method	Time [ms]	fps	Time [ms]	fps
Proposed	28.9	34.6	72.1	13.87
MS ICE	1189	0.84	1303	0.77
Panoweaver	N/A	N/A	N/A	N/A

5.5 Response to External Disturbance

The developed dynamics-based image processing methods are not robust to external disturbances. Since the developed methods do not rely on external sensors or analyze image domains, artifacts may exist in de-blurring or stitching process due to the external disturbances. Although it has been assumed that no unpredictable

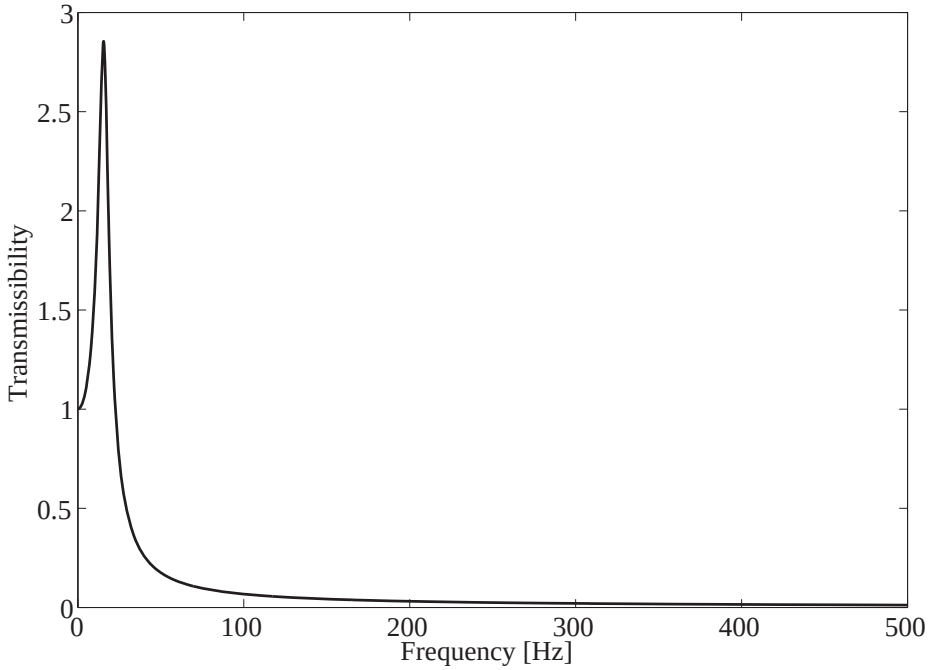


Figure 61: Transmissibility analysis of the camera orientation system.

disturbances exist in this study, simulation and experiments are conducted to validate the mechanical design of the camera orientation system in regard to the external disturbances.

Figure 61 shows the transmissibility analysis by simulation for the camera orientation system, where the transmissibility is the ratio of the angular displacement of the camera to the external disturbances. It can be observed that the high-frequency disturbances are decayed out, and thus the effect of high-frequency noises on image acquisition may be negligible.

$$\text{Transmissibility} = \frac{\sqrt{1 + (2\zeta\gamma)^2}}{\sqrt{(1-\gamma^2)^2 + (2\zeta\gamma)^2}} \tag{35}$$

where $\gamma = \omega/\omega_n$.

Figure 62 shows the experimental setup to test the transmissibility of the camera orientation system. The camera orientation system was assembled to a frame, which is rotatable. An impact force was manually applied on the indicated point by a

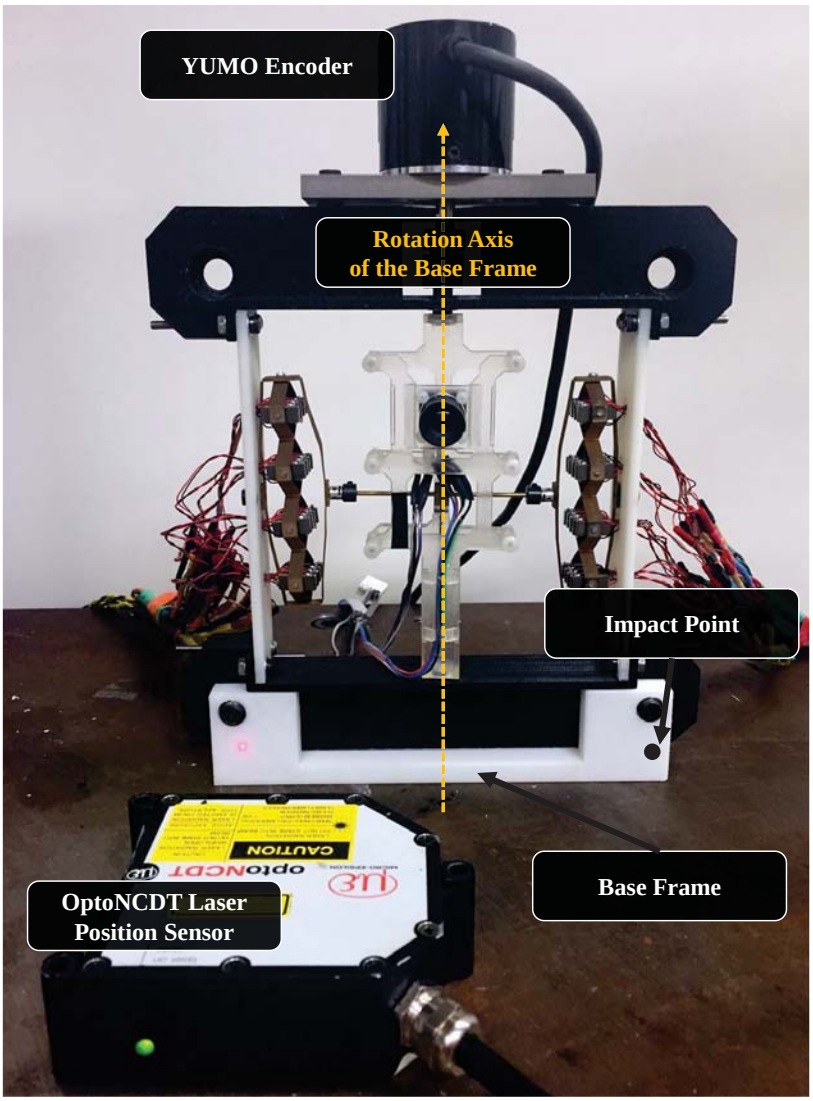


Figure 62: Transmissibility analysis of the camera orientation system.

rubber hammer. The angular displacement of the entire camera orientation system can be derived from the measurement by a Micro-Epsilon optoNCDT 2200 ILD 2200-20 laser position sensor. The angular displacement of the camera within the frame was measured by a 10-bit YUMO E6B2-CWZ3E encoder. Therefore, the angular displacement of the camera in the global coordinate can be determined by comparing the signals captured by the laser sensor and encoder.

Figure 63 shows the responses of the entire camera orientation system and camera inside the frame due to the impact force. The total angular displacement of the entire

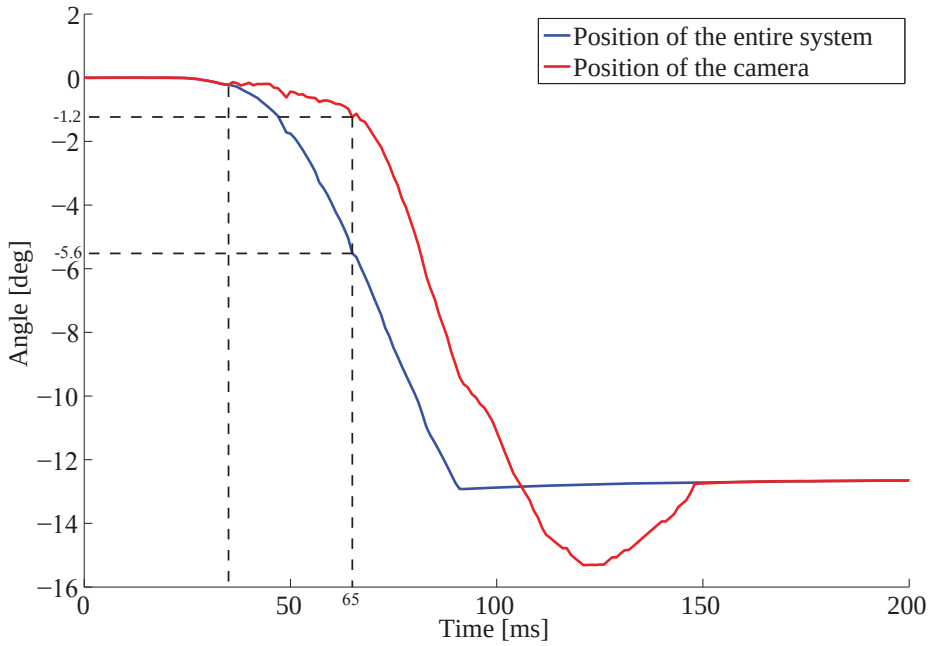


Figure 63: Responses of the entire system and camera due to the impact force.

system due to the impact was 12.9 degrees. In can be observed that the motion of the entire system was not fully transmitted to the camera in the early stage of response. When the angular displacement of the entire system was -5.6 degrees, the angular displacement of the camera was -1.2 degrees. The external disturbance was compensated by flexibility in the actuation system.

Additional experiments were conducted when the camera orientation system was firmly fixed. Figure 64 shows the experimental setup to observe responses to the external disturbances.An impact force was manually applied on the frame by using a rubber hammer. The magnitude of the applied force was 100 $\pm$ 5 N, which was measured by a Omegadyne LCM703-50 load cell. The external force was applied on the system with a total of 6 different conditions as demonstrated in Table 11. For each load condition, a total of 10 trails were conducted.

The camera experienced impulse-like displacements in response to the impact force and the peak values are shown in Figure 65. When the force was applied on the bottom part of the frame, the angular displacements were not significant. However,

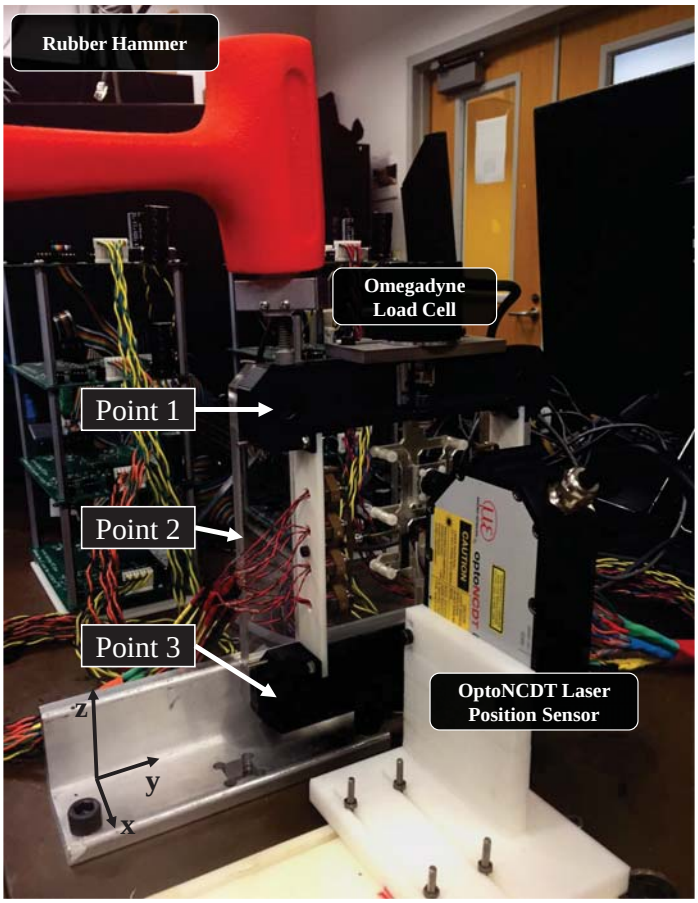


Figure 64: Experimental setup. The camera orientation system is supported in a vertical manner. A rubber hammer is used to apply external disturbances to the camera orientation system.

the angular displacements were not negligible when the force was applied on the top part of the frame. This is because the camera orientation system was supported in a vertical manner and only the bottom part of the frame was firmly fixed. The displacement was significant when the force was applied normal to the lens surface due to bending moment. When the impact force was vertically applied on the Point 1, the white plastic frame was bended resulting in a angular displacement.

The same experiment was conducted when the entire frame of the camera orientation system was firmly fixed to a rigid surface as shown in Figure 66. The load conditions are tabulated in Table 12 and the peak values of the angular displacements in response to the impact load are shown in Figure 67. It can be observed that the

Table 11: Conditions of the external force applied on the system

Condition #	Impact Point	Direction of the force
1	1	y
2	2	y
3	3	y
4	1	-x
5	3	-x
6	1	-z

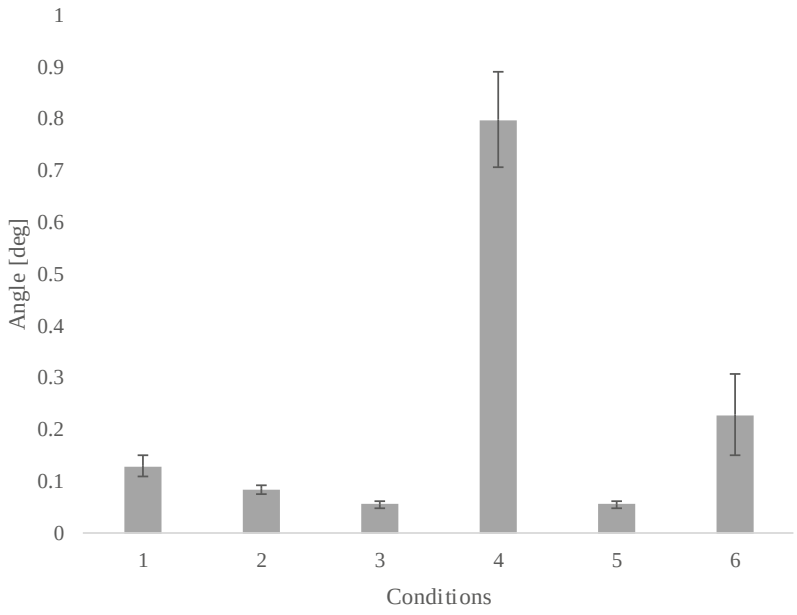


Figure 65: Peak values of the displacements due to the impact force applied on the camera orientation system.

external disturbances are negligible when the entire frame of the camera orientation system was firmly fixed.

Table 12: Conditions of the external force applied on the system

Condition #	Direction of the force
7	-x
8	-y
9	-z

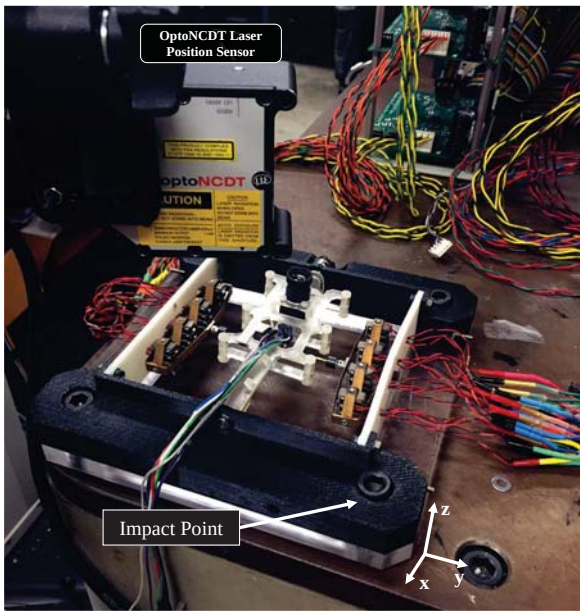


Figure 66: Experimental setup. The entire frame of the camera orientation system is firmly fixed.

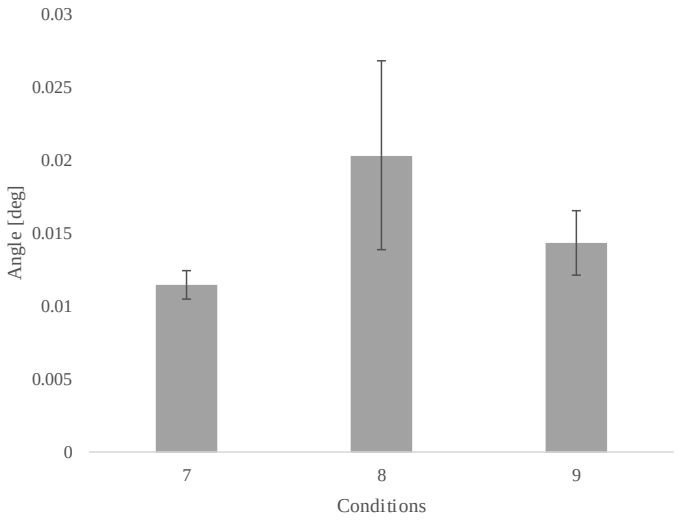


Figure 67: Peak values of the displacements due to the impact force applied on the camera orientation system.

5.6 Summary

This chapter presented a method to generate a panoramic image in real-time using a robotic vision system that can move as fast as human eyes [54, 52]. The dynamics-based image stitching method generated a panoramic image from images acquired in the course of rapid motion. The velocity of the robotic vision system was limited for the sake of comparison. The proposed method was robust to degraded blurry images while the conventional methods were not robust. The experimental results showed that the real-time performance for image stitching within 30 fps was achieved when VGA-size images were used. The performance metric of the proposed method was 41% better than the conventional methods.

The analysis on external disturbances was conducted, although it has been assumed that no external disturbances exist. The high-frequency disturbances were compensated out due to flexibility in the actuation system. Also, the disturbances were negligible on the camera when the entire system was firmly fixed.

CHAPTER VI

DISCRETE SWITCHING CONTROLLER FOR SMOOTH-PURSUIT LIKE MOVEMENTS

This chapter presents a method to generate discrete switching commands for a camera orientation system exhibiting quantization and flexibility to achieve smooth-pursuit like movements. While the human eyes have two representative movements that are saccade and smooth pursuit [24], prior works on robotic camera positioning systems have been focused only on saccade-like motion [55, 54, 96, 119]. Existing vibration suppression techniques also have focused on point-to-point movements.

An arbitrary velocity profile is generated by combining and optimizing a series of pulse-like commands at each primitive. The proposed controller generates pulse-like switching commands to match the velocity of an object of interest for image acquisition. Unlike other visual tracking methods for feeding video, the proposed method is designed to match the velocity of the object of interest only during the exposure window. Therefore, discrete switching commands for vibration suppression are generated after the exposure window to reduce the number of switching further.

A desired motion profile may be generated using a classical feedback controller with PWM quantization but each unit actuator is required to produce high-frequency switching that can potentially damage the complaint mechanism. For this reason, a discrete control method with reduced numbers of switching must be developed.

6.1 Cellular Actuator: Compliance and Quantization

The cellular actuator used in this study for generating motions is shown in Figure 68. A total of 16 lead zirconate titanate (PZT) actuators are nested in a multi-layer

fashion [98]. Each PZT actuator is independently operated in a binary manner (i.e., either 'ON' or 'OFF'), as if each motor unit in the biological system switches between contraction and relaxation. In this way, the cellular actuator exhibits quantization.

A single PZT actuator, APA50XS by Cedrat Technologies, can generate a blocking force of 18 N and a maximum displacement of 80 μm [14]. The multi-layer mechanism is made of brass and has a thickness of 170 μm, which introduces compliance in the mechanism. By combining these features, the multi-layered structure is capable of creating increased displacements and strains at the output point, while displacements of a single PZT are limited [116].

Figure 69 shows the single DOF camera positioner used in this study [97]. Two cellular actuators are connected to a moving platen via a rod and placed in an antagonistic manner. When the PZT actuators are activated, the output point of the cellular actuator moves inward, i.e., contraction. By contracting or activating one cellular actuator while relaxing or deactivating the other cellular actuator, a tilting motion of the camera is generated. In this study, only one of the cellular actuators was activated to generate motion.

6.2 Quantization Effects

Considering the quantization effect due to independent binary operations in the individual units, the output displacements of the cellular actuator are discrete as shown in Figure 70. Consequently, the angular displacements of the camera orientation system with respect to the number of PZT actuators activated are discrete as shown in Figure 71. It is assumed that the cellular actuator is a linear system, since no significant nonlinearity was observed in the discrete angular displacements during overall activation; however, minor nonlinearity was observed when a total of 12 PZTs were activated. The camera positioner created 16 discrete angles that were measured by a 10-bit YUMO E6B2-CWZ3E encoder. The resolution of 0.6407° with a σ value of

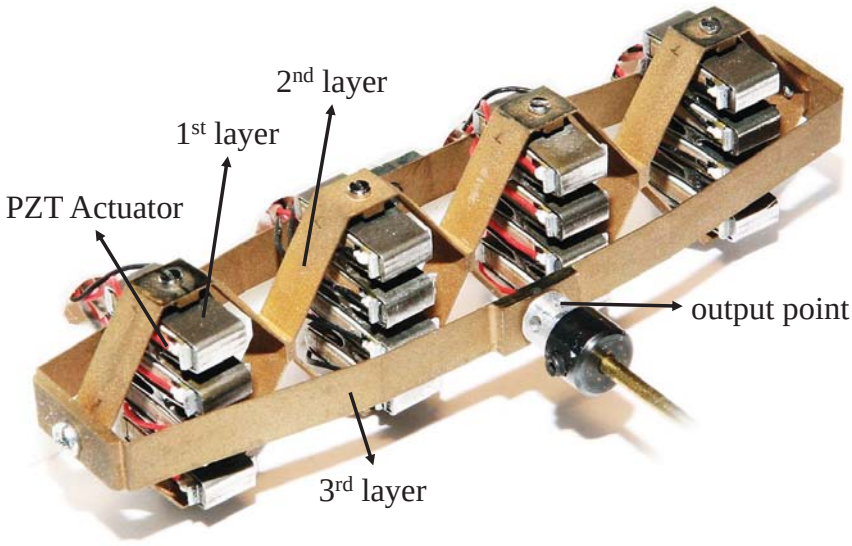


Figure 68: The cellular actuator exhibits quantization and flexibility. A total of 16 PZT actuators is nested in a multi-layer fashion and each PZT is operated in an ON-OFF manner.

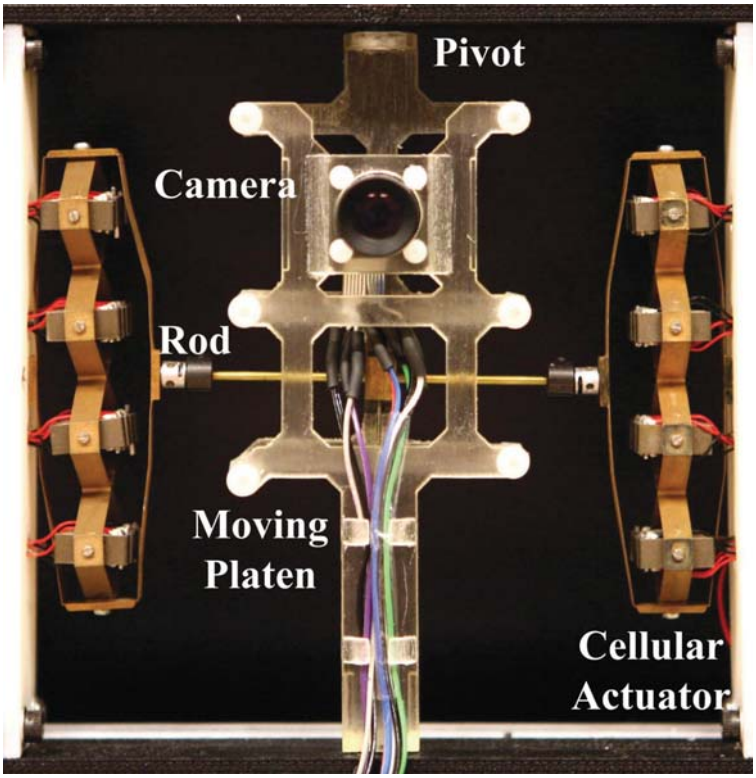


Figure 69: The camera orientation system. It has a 1 DOF motion driven by an antagonistic pair of cellular actuators.

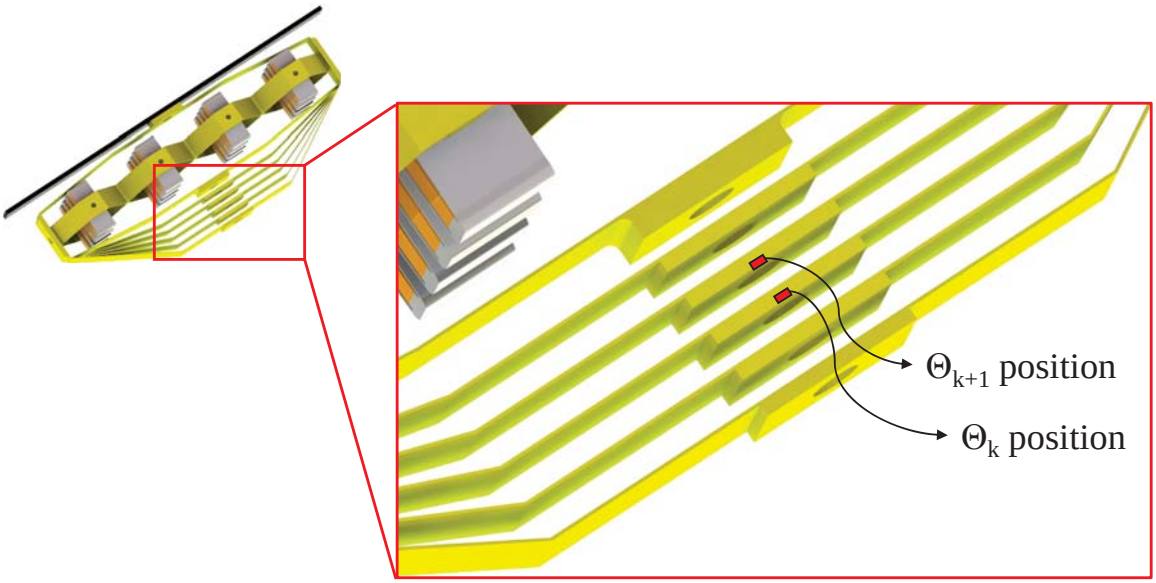


Figure 70: The cellular actuator produces discrete output positions due to the binary operation in the PZT actuators.

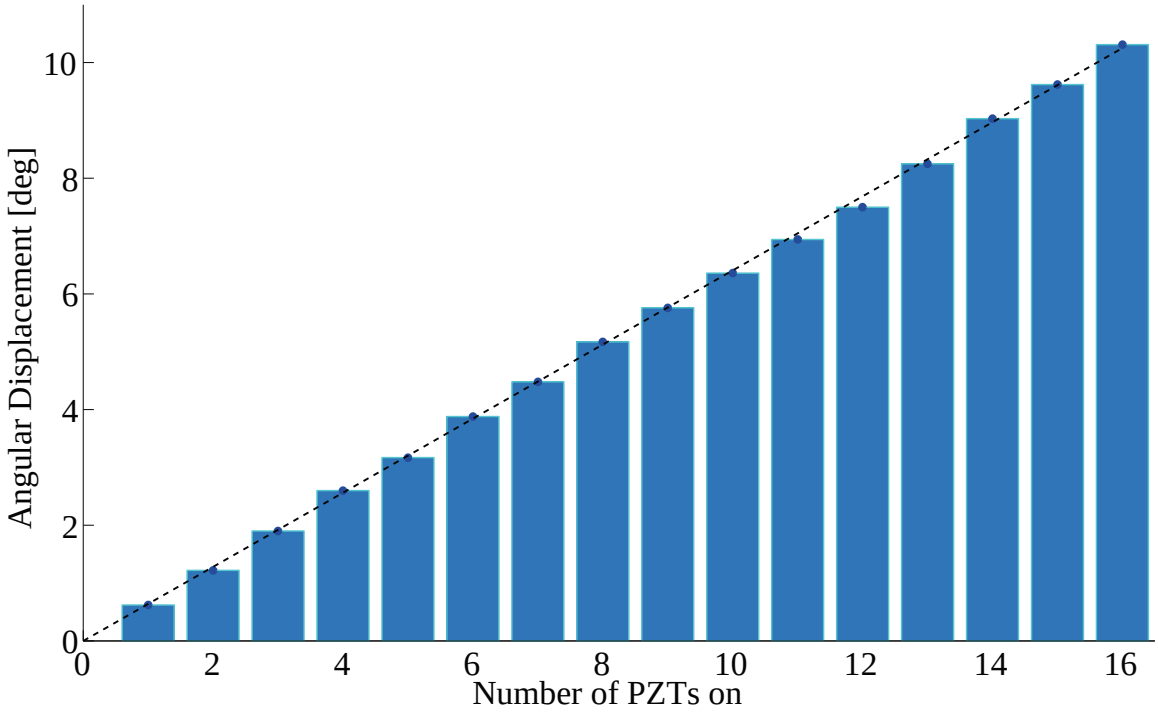


Figure 71: Angular displacements of the camera orientation system with respect to the number of individual PZTs activated.

0.0032 and an R^2 value of 0.9995 was obtained in the discrete actuation of the camera orientation system.

Let p_k be the activation level of the cellular actuator, and a total of k PZT actuators are activated. Then, the corresponding discrete output position Θ_k is

$$\Theta_k = \Psi \cdot p_k \tag{36}$$

where Ψ is a linear mapping observed in Figure 71.

To reproduce smooth pursuit-like movements, this study aims to generate an arbitrary velocity profile from one discrete position Θ_k to the next one Θ_{k+1}. The motion y can be achieved by

$$y(t) = f(x(t),\ u(t),\ t) \tag{37}$$

$$y(0) = \Theta_k$$

where t is the time, f is the dynamics of the camera orientation system, x is the state, and u is the control input.

In order to achieve an arbitrary velocity profile, an appropriate $u(t)$ value needs to be determined. Owing to the quantization effects in the actuation system, the control input must be discretized as opposed to conventional control inputs, which are continuous variable. Therefore, the following constraint must be satisfied:

$$u(t) \in \mathbb{Z}. \tag{38}$$

Figure 72 shows a response of the camera orientation system when a single PZT actuator is activated. Let the maximum velocity in the course of the transient response be simply equivalent to $\frac{x_p}{t_p}$. For tracking a slowly moving object, this study focuses on velocity profiles that are slower than vel_{max}. It was considered that velocity profiles comparable to vel_{max} are saccade-like movements.

Assume the desired motion is a constant velocity that is 10 times slower than vel_{max} as shown in Figure 72. The transient response is dependent on mechanical

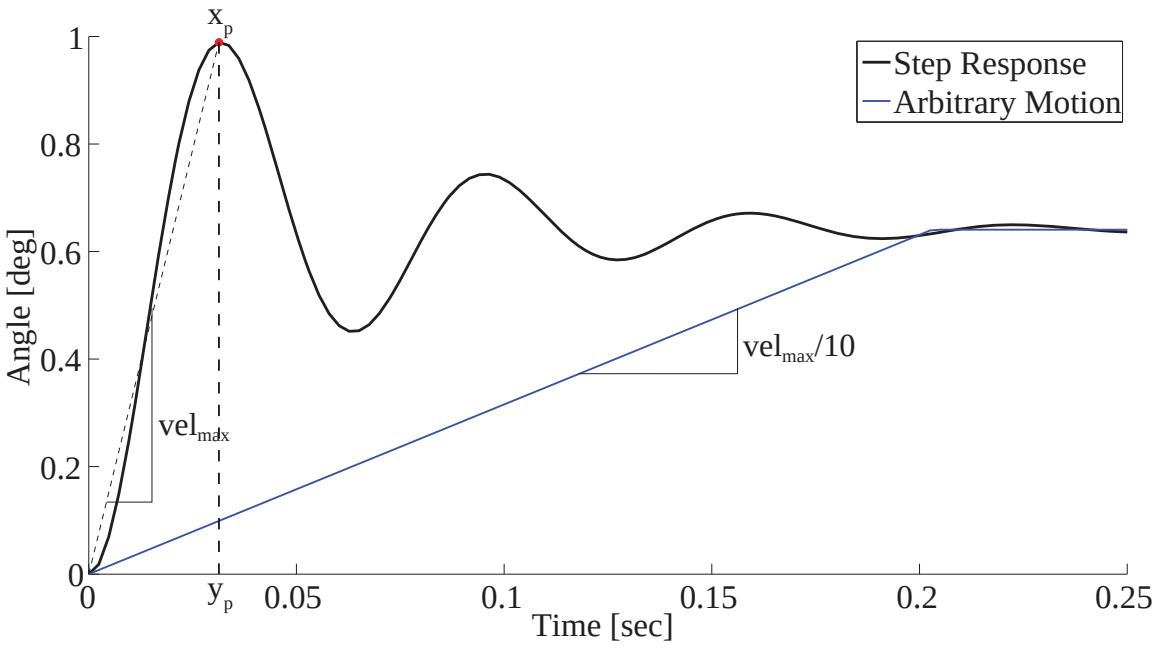


Figure 72: Step Response of the camera orientation system. This study is interested in generation of motion with velocity slower than the early stage of step response.

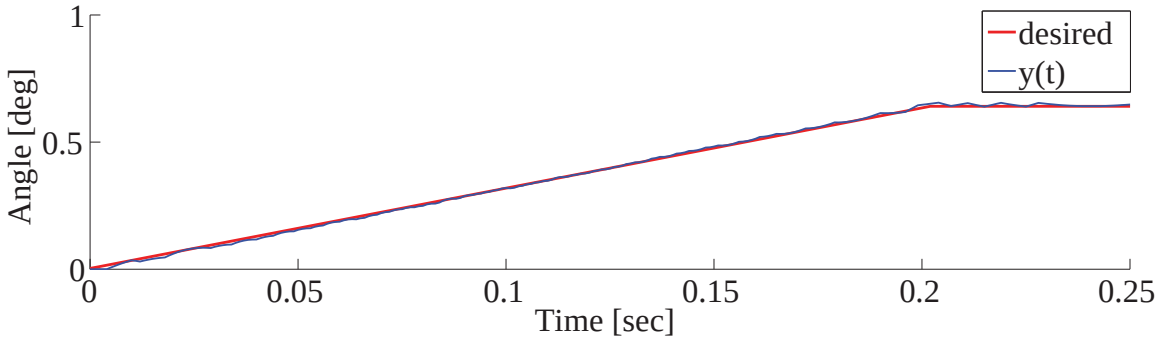
properties such as the damping coefficient ζ and dominant natural frequency ω_n. Each individual unit may need to be activated and deactivated consistently to generate slowly moving profiles [53]. One conventional approach is to use a feedback controller with PWM quantization [112].

Figure 73 shows the simulation result of a PWM feedback controller to achieve this motion. The cost function was

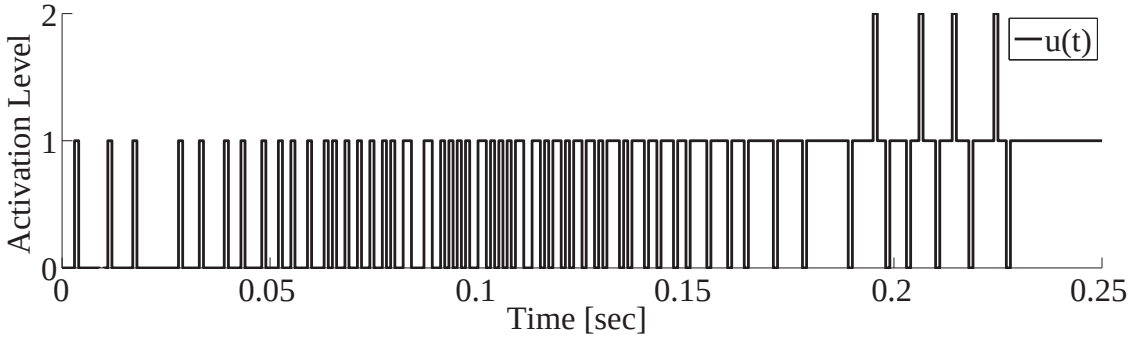
$$J = \int_0^{t_{end}} (x^T Q_{opt} x + u^T R_{opt} u) dt + x^T(t_{end}) S_{opt} x(t_{end}) \tag{39}$$

where Q_{opt}, R_{opt}, and S_{opt} are positive definite weighting matrices. The sampling frequency was 1 kHz. The continuously variable control input $\tilde{u}$ was quantized into u by PWM modulation through iterative optimization processes to satisfy (38).

It can be verified with simulation that the feedback controller with PWM quantization successfully tracks the desired motion. However, the controller causes high frequency switching in the individual PZT actuators as shown in Figure 73(b). In



(a) Time response



(b) Activation level

Figure 73: Using a conventional feedback controller with PWM quantization, the camera orientation system can track a ramp signal that is 10 times slower than the maximum velocity in response to the step input. However, it causes high-frequency switching in the individual PZTs.

practice, this is not desirable because it may potentially damage the compliant mechanism, which has a thickness of 170 μm. In addition, this approach has little in common with the actuation strategy in the biological system [95]. Therefore, a discrete switching controller should be developed.

6.3 Discrete Switching Commands

The proposed discrete switching controller is designed to track an object of interest only during the exposure windows. It mainly consists of two parts: (1) Generating velocity profiles for tracking, and (2) Suppressing vibration. The discrete switching commands for tracking are generated to match the velocity of the object of interest, similar to smooth pursuit in the biological system. The commands are given to

the system during the exposure windows, and thus images can be obtained in the course of motion while avoiding blurriness of the object of interest. After images are obtained, additional discrete switching commands are generated for suppressing vibrations and attaining the desired position. The commands for vibration suppression require less switching than in the tracking period. No additional switching is required once the vibration is completely suppressed. Therefore, the suppressing vibrations limits switching in individual PZT units, and avoids fatigue in the thin compliant mechanism.

6.3.1 System Model and Modified Input

It was observed that the dynamics of the camera orientation system was slightly changed and exhibited a linear third-order system. The transfer function of a third-order system can be represented as

$$G(s) = \frac{K}{s^2 + 2\zeta\omega_n s + \omega_n^2} \cdot \frac{1}{\tau s + 1} \tag{40}$$

where τ is equivalent to a time constant in the first order system. The time response of a third-order system to step inputs can be given as

$$\theta(t) = \sum_{i=1}^{n} \frac{A_i \cdot 1(t - t_i)}{\omega_n^2} \Big(1 - \frac{e^{(t-t_i)/\tau}}{C_1} - \frac{e^{-\zeta\omega_n(t-t_i)}}{C_1} \cdot [C_2 \cos(\omega_d(t - t_i)) + C_3 \frac{\zeta}{\sqrt{1-\zeta^2}} \sin(\omega_d(t - t_i))]\Big) \tag{41}$$

where

$$C_1 = \frac{1}{\eta(\eta - 2)\zeta^2 + 1}$$

$$C_2 = \eta(\eta - 2)\zeta^2$$

$$C_3 = \eta((\eta - 2)\zeta^2 + 1)$$

$$\eta = \frac{1}{\tau\zeta\omega_n}$$

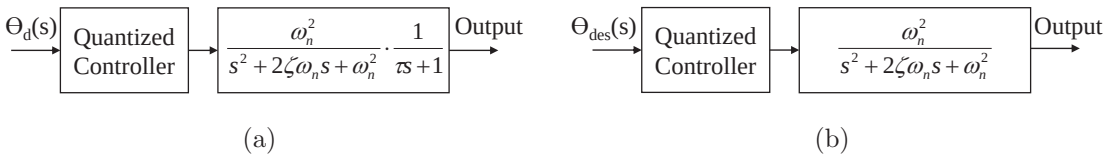


Figure 74: Block diagrams. (a) Open-loop controller for a third-order system. (b) Open-loop controller for a second-order system with modified desired profile.

System parameters were determined experimentally by the observation of responses to discrete switching commands. The time constant τ was 0.005, the natural frequency was slightly changed to 11.4 Hz, and the damping ratio was 0.19.

In order for smooth-pursuit like movements, an open-loop quantized controller needs to be designed to achieve an arbitrary velocity profile as shown in Figure 74(a). Although the mechanical system is a third-order system, this study treats the camera orientation system as a second-order system. The first-order term $\frac{1}{\tau s+1}$ is neglected in the plant which causes lag. This effect can be compensated by modifying the signal given to the controller to $\Theta_{des}(s) = \Theta_d(s)(\tau s + 1)$ as shown in Figure 74(b), while the desired profile is $\Theta_d(s)$. Therefore, the modified signal in the time domain is determined as

$$\theta_{des}(t) = \theta_d(t) + \tau\dot{\theta}_d(t). \tag{42}$$

This approach is applicable if the system is linear.

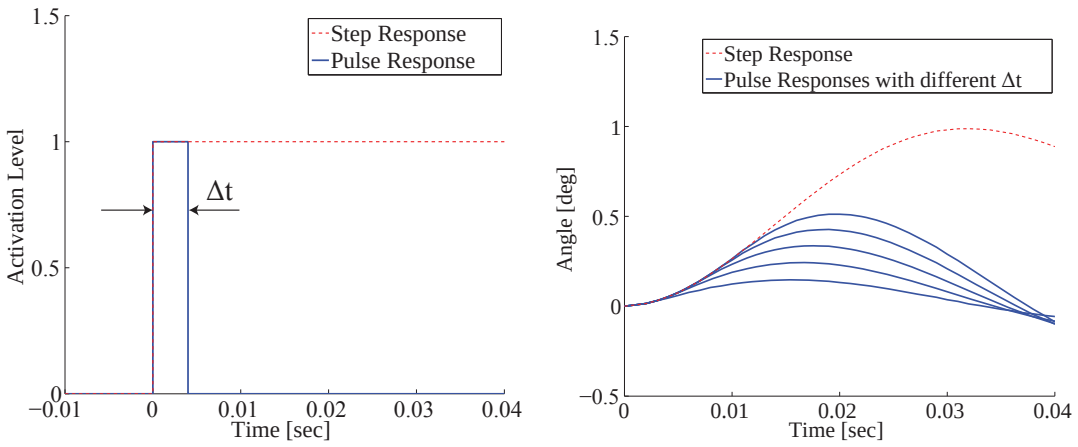
6.3.2 Tracking Commands

The time response of the camera orientation system, modeled as a linear second order system, can be represented as

$$\theta(t) = \Psi \cdot \sum_{i=1}^{n} \frac{A_i \cdot 1(t-t_i)}{\omega_n^2}[1 - \frac{e^{-\zeta\omega_n(t-t_i)}}{\sqrt{1-\zeta 2}} \sin(\omega_d(t-t_i)+\psi)] \tag{43}$$
$$A_i \in \mathbb{Z},\ \forall i$$

where

$$\psi = \text{atan}\left(\frac{\sqrt{1-\zeta^2}}{\zeta}\right), \tag{44}$$



(a) Pulse generation (b) Time response

Figure 75: Example of pulse generation. The amplitude of the response can be adjusted by changing Δt.

n is the total number of inputs given to the system, A_i is the amplitude of the i$^{\text{th}}$ step input, t_i is the time of the given i$^{\text{th}}$ step input, and ω_d is the damped natural frequency.

The objective is to determine piecewise functions to generate motion primitives. The piecewise function is a pulse-like command consisting of positive and negative step inputs. By changing the time interval, Δt, between the ON and OFF commands, the amplitude of the response can be adjusted as shown in Figure 75(b). Since the camera orientation system is modeled as an LTI system, an arbitrary velocity profile can be achieved by a summation of the responses at each primitive.

The brief procedure of the proposed discrete switching controller is shown in Figure 76. A total of m pulses is given to the system to achieve the desired motion from activation level p_k at time t_k to p_{k+1} at time t_{k+1}. There is a tradeoff between the number of pulses m and the tracking performance. The number of pulses must be chosen such that the control frequency ω_c is at least two times greater than the natural frequency ω_n to meet the Nyquist criterion.

Due to redundancies in the actuation system, a number of solutions exist. The solution space is reduced by generating the ON signal in a periodical manner.

$$t_{on,j} = t_k + \frac{t_{k+1} - t_k}{m}(j-1) \tag{45}$$

where j denotes the j$^{\text{th}}$ pulse and *on* denotes the positive step input. Since redundancy still exist, the solution space is narrowed further by introducing the following constraints:

$$|A_{on,j}| = |A_{off,j}| = 1 \tag{46}$$

$$p_k \leq \sum A_i \leq p_{k+1}, \forall i \tag{47}$$

where off denotes the negative step input. This bounds the activation level between the start and desired discrete positions at any time, limiting the amplitude of residual oscillations in the compliant actuation system. In addition, this could prevent failure in implementation. For instance, the control input may not be possible to implement when approaching to the maximum activation level p_{max} from p_{max-1} without the bounding constraints.

The desired motion profile $\theta_{des}(t)$ is divided into m target positions $\bar{\theta}_{tar,j}$ along the trajectory with a constant time interval as

$$\bar{\theta}_{tar,j} = \theta_{des}(t_{on,j+1}). \tag{48}$$

The j$^{\text{th}}$ OFF command is followed by the ON command with a time interval of Δt_j at time $t_{off,j}$ to satisfy (48) at each primitive. Due to residual vibration, the time interval Δt_j is not simply linear with $\bar{\theta}_{tar,j}$ as in PWM. Timing of the OFF commands $t_{off,j}$ are determined at each primitive in a recursive manner as follows.

Residual vibrations may exist in the system due to control inputs. Let the residual vibration at $t_{on,j}$ be introduced into the system at $t_{r,j}$ with an amplitude of $A_{r,j}$ and a phase of $\phi_{r,j}$. Unlike A_i, the amplitude of the residual vibration $A_{r,j}$ is not necessarily an integer. After a pulse-like command are generated, three different signals, the residual vibration, and vibrations due to ON and OFF commands, can be represented

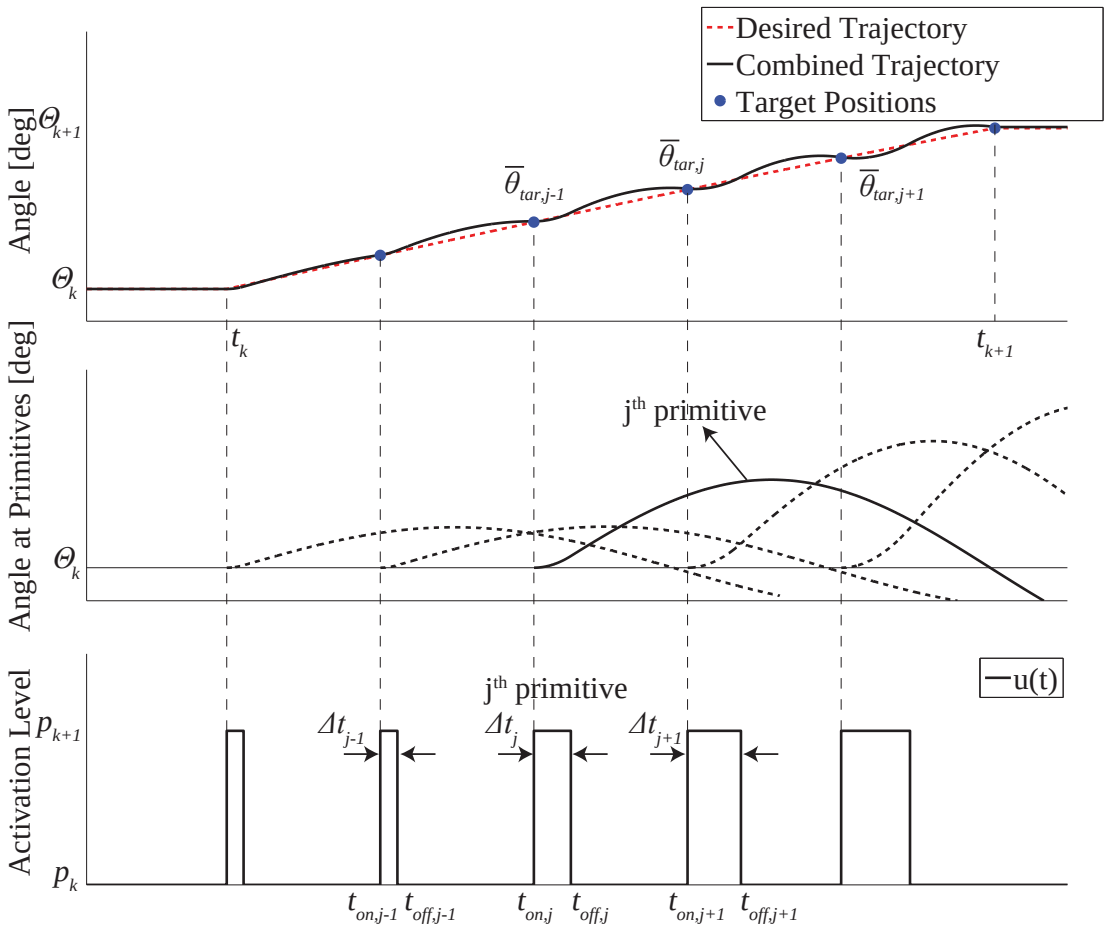


Figure 76: Generation of discrete commands for velocity matching. The timings of primitive movements are determined to achieve the target positions indicated by blue circles at the time when the next primitive is generated.

as a single impulse response. Let $B_{r,j}$, $B_{on,j}$, and $B_{off,j}$ be the decayed amplitudes of the residual vibrations at time $t_{off,j}$ due to $A_{r,j}$, $A_{on,j}$, and $A_{off,j}$, respectively.

$$B_{r,j} = A_{r,j} \cdot e^{-\zeta\omega_n(t_{off,j}-t_{r,j})} \tag{49}$$

$$B_{on,j} = -A_{on,j} \cdot e^{-\zeta\omega_n(t_{off,j}-t_{on,j})} \tag{50}$$

$$B_{off,j} = -A_{off,j} \tag{51}$$

Similarly, phases can be calculated at time $t_{off,j}$.

$$\Phi_{r,j} = \omega_d(t_{off,j} - t_{r,j}) + \phi_{r,j} \tag{52}$$

$$\Phi_{on,j} = \omega_d(t_{off,j} - t_{on,j}) + \psi \tag{53}$$

$$\Phi_{off,j} = \psi \tag{54}$$

where $\Phi_{r,j}$, $\Phi_{on,j}$, and $\Phi_{off,j}$ are the phases of $B_{r,j}$, $B_{on,j}$, and $B_{off,j}$, respectively, at $t_{off,j}$. An equivalent single impulse input with an amplitude of B_{eqv} and a phase of Φ_{eqv} given at time $t_{off,j}$ representing these three signals can be determined as follows:

$$B_{eqv} = \sqrt{(B1)^2 + (B2)^2} \tag{55}$$

$$\Phi_{eqv} = \text{atan}\left(\frac{B2}{B1}\right) \tag{56}$$

where

$$B1 = B_{r,j}\cos(\Phi_{r,j}) + B_{on,j}\cos(\Phi_{on,j}) + B_{off,j}\cos(\Phi_{off,j}) \tag{57}$$

$$B2 = B_{r,j}\sin(\Phi_{r,j}) + B_{on,j}\sin(\Phi_{on,j}) + B_{off,j}\sin(\Phi_{off,j}) \tag{58}$$

When a pulse-like command is given to the system, the response of the camera orientation system at time τ can be represented as:

$$\theta_j(\tau) = \theta_{ref} + \Psi\Big(\frac{B_{eqv}}{\omega_n^2\sqrt{1-\zeta^2}}e^{-\zeta\omega_n(\tau - t_{off,j})} \cdot \sin(\omega_d(\tau - t_{off,j}) + \Phi_{eqv})\Big), \tag{59}$$

$$\tau > t_{off,j}$$

where θ_{ref} is the reference position to the desired motion, which may be Θ_k.

Timing of primitive movements are determined to achieve target positions at the time when the next primitive is generated. Therefore, timing of the OFF command can be determined by solving (59) for $t_{off,j}$ while setting $\tau = t_{on,j+1}$ and $\theta_j = \bar{\theta}_{tar,j}$. Since the pulse-like commands are generated at each primitive, the solution must satisfy

$$t_{off,j} \in [t_{on,j},\ t_{on,j+1}]. \tag{60}$$

Therefore, $t_{off,j}$ can be numerically determined by using bracketing methods [15, 38], guaranteeing convergence. The following target positions can be achieved by

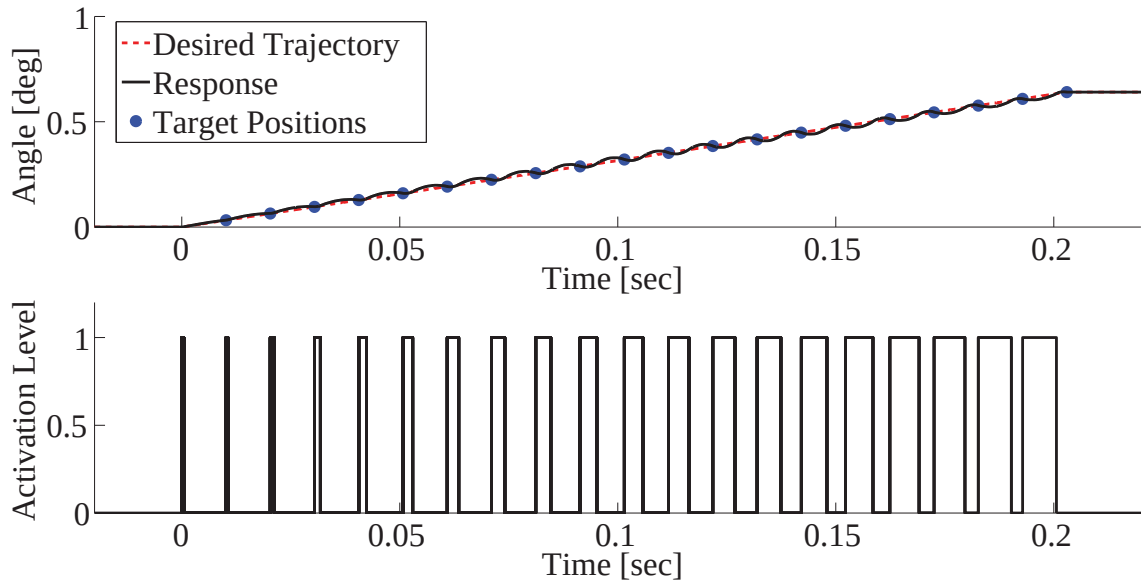


Figure 77: Discrete switching commands for generating an arbitrary movement. The desired trajectory is the same as that in Figure 73.

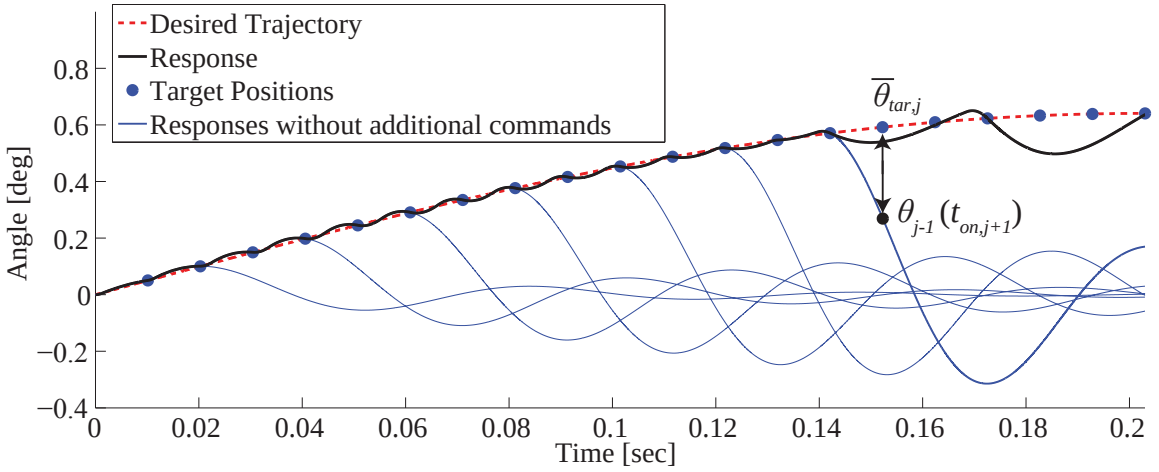


Figure 78: Failure of command generation in achieving the target position.

substituting $A_{r,j+1}$, $\phi_{r,j+1}$, and $t_{r,j+1}$ with B_{eqv}, Φ_{eqv}, and $t_{off,j}$, respectively, and repeating (49)-(60) from $j = 1$ to $j = m$.

Figure 77 shows the simulation results of the proposed controller when the desired trajectory is identical to the simulation shown in Figure 73. Although minor position errors exist, it can be observed that the proposed controller successfully achieved the desired motion with less switching.

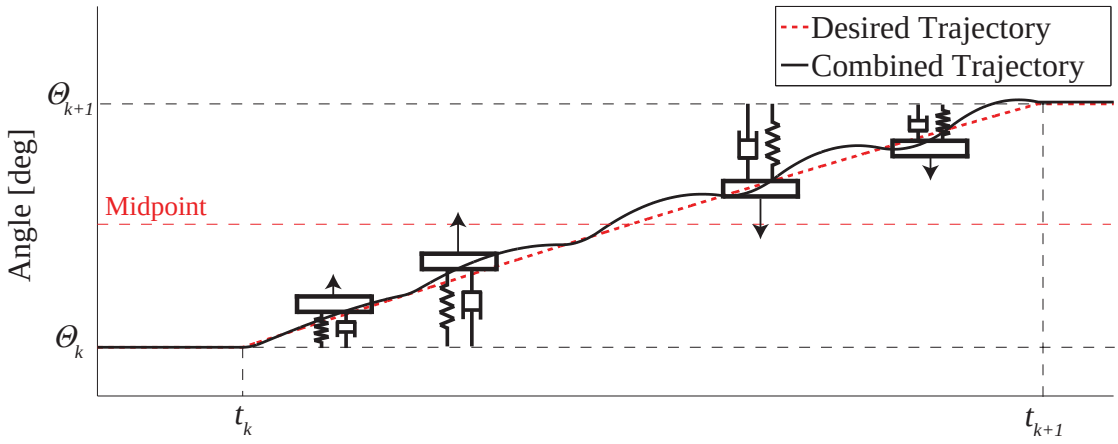
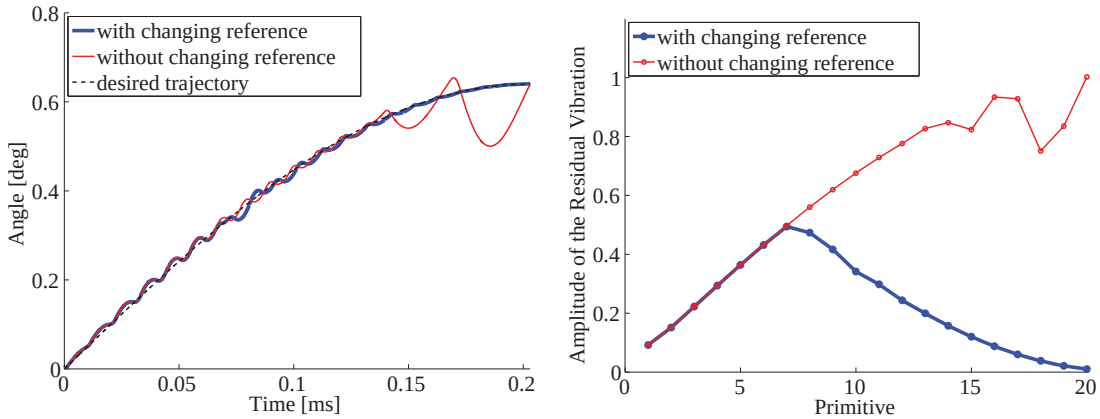


Figure 79: Conceptual depiction of the change in the reference position when the desired motion crosses the midpoint.



(a) Responses (b) Residual Amplitudes

Figure 80: Comparison between with and without changing the reference.

The resultant position due to a total of $j-1$ pulse-like commands at time $t_{on,j+1}$ is $\theta_{j-1}(t_{on,j+1})$ as indicated by a black dot in Figure 78. The j$^{\text{th}}$ pulse-like command is generated to compensate the discrepancy between $\bar{\theta}_{tar,j}$ and $\theta_{j-1}(t_{on,j+1})$. However, the discrete switching controller cannot achieve the target position if $|\bar{\theta}_{tar,j} - \theta_{j-1}(t_{on,j+1})| > \Delta\theta_{max}$, where $\Delta\theta_{max}$ is the maximum displacement that can be generated at the primitive.

$$\Delta\theta_{max} = \frac{\Psi}{\omega_n^2}[1 - \frac{e^{-\zeta\omega_n \frac{(t_{k+1}-t_k)}{m}}}{\sqrt{1-\zeta^2}} \sin(\omega_d \frac{(t_{k+1}-t_k)}{m} + \psi)] \tag{61}$$

Due to the nature of pulse-like commands, the response settles back to the reference position θ_{ref} after the target position is achieved. This settling displacement is linear to B_{eqv}. The amplitude of the residual vibration B_{eqv} increases as the displacement $|\bar{\theta}_{tar,j} - \theta_{ref}|$ increases; $B_{eqv} \approx |\bar{\theta}_{tar,j} - \theta_{ref}|$. Therefore, the controller may fail to compensate for the discrepancy in positions when $\bar{\theta}_{tar,j}$ is close to Θ_{k+1}. The value of $|\bar{\theta}_{tar,j} - \theta_{j-1}(t_{on,j+1})|$ is also dependent on the phase Φ_{eqv}, which varies at each primitive.

To avoid this issue, the reference position θ_{ref} is changed to Θ_{k+1} when $\bar{\theta}_{tar,j}$ exceeds the midpoint $\frac{\Theta_k + \Theta_{k+1}}{2}$, or vice versa. In this case, the ON command is not generated and does not obey (45). Let this takes place at the $\mathrm{j_c}^{\mathrm{th}}$ primitive and the sign of A_{off,j_c} is inverted. The same procedures (49)-(60) are used to determine t_{off,j_c}.

After the reference position is changed, the calculation of command timings is identical except that the signs of $A_{on,j_c \sim m}$ and $A_{off,j_c \sim m}$ are inverted. Figure 79 shows the concept of changing the reference position when the target position crosses the midpoint. The discrete switching controller is considered to push the actuation system toward the midpoint by inducing residual vibrations until the reference position θ_{ref} is changed. Then, the discrete switching controller gradually diminishes the residual vibrations to settle the actuation system at the desired discrete position Θ_{k+1}. Using this approach, the range of B_{sum} can be reduced to approximately $[0, \frac{|A_i|}{2}]$, which helps avoid failure as shown in Figure 80. The program containing the proposed discrete switching controller is summarized by the pseudocode in Algorithm 1.

6.3.3 Validation of the Controller for Tracking

The proposed discrete controller for tracking an arbitrary velocity profile is experimentally validated. A photo of the experimental setup is shown in Figure 81. The moving platen is measured by a Micro-Epsilon optoNCDT 2200 ILD 2200-20 laser position sensor. The position resolution, linearity, and measurement frequency of

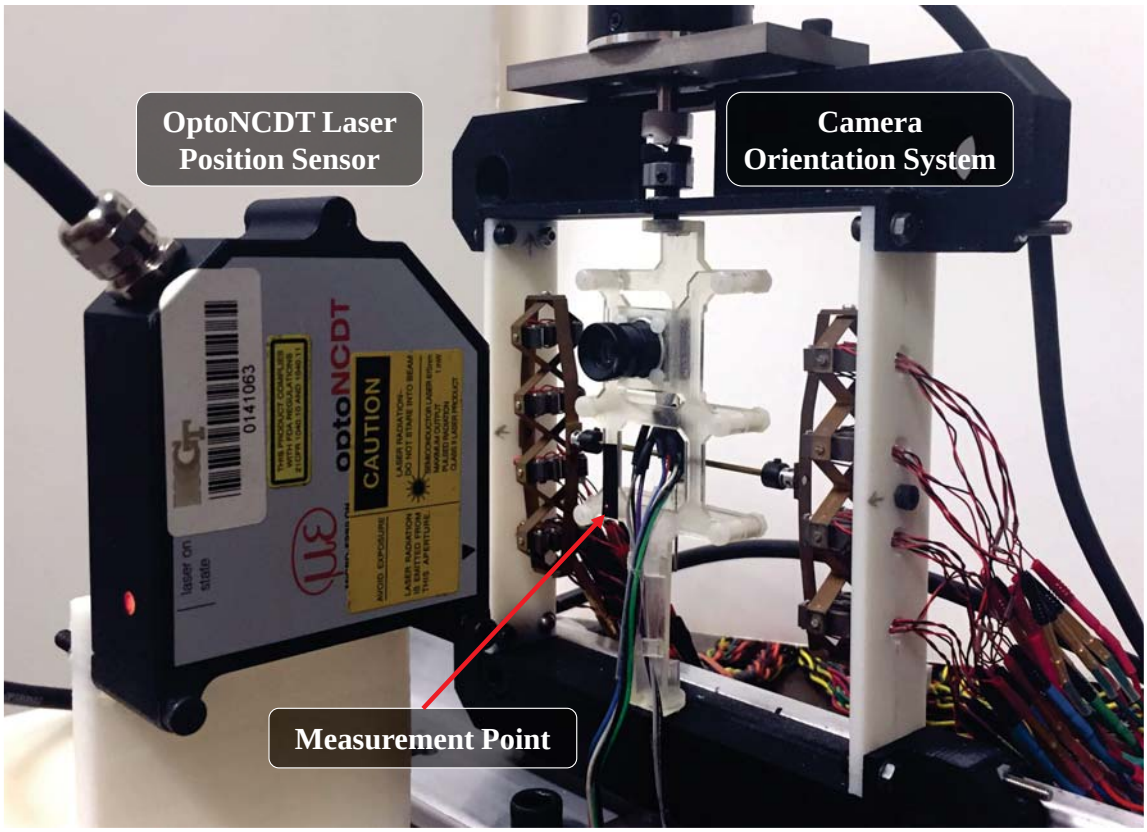


Figure 81: Experimental setup for measuring the responses.

the laser sensor are 0.3 μm, 6 μm, and 10 kHz, respectively. The signal from the laser sensor was recorded in the FPGAs with cRIO-9118 and NI 9205 platforms. The

Algorithm 1 Discrete Controller for Tracking

Input: desired motion θ_{des}; number of pulses m;
Define timings of ON commands $t_{on,j}$ by (45)
Define target positions $\bar{\theta}_{tar,j}$ by (46)
for each primitive j **do**
    if $\theta_{des,j-1} \rightarrow \theta_{des,j}$ crosses the midpoint **then**
        Invert signs of $A_{on,j\sim m}$ and $A_{off,j\sim m}$
        Ignore (45) and omit ON command
    end if
    Determine timings of OFF commands $t_{off,j}$
     by (49) - (60)
    Update residual vibrations:
    $B_{r,j+1} = B_{eqv}$, $\phi_{r,j+1} = \Phi_{eqv}$, and $t_{r,j+1} = t_{off,j}$
end for
Output: Discrete switching input $u(t)$

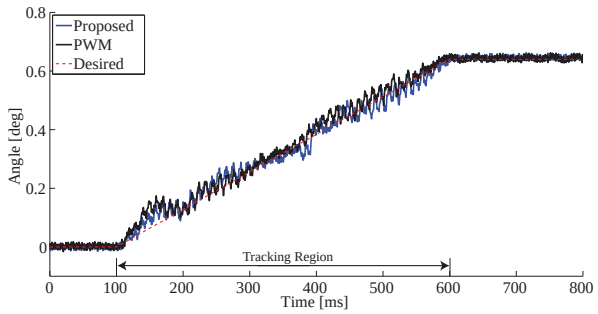
measured displacements are converted to angles by triangulation.

The proposed controller was tested for various motion profiles and compared with a conventional PWM controller. For a direct comparison, the PWM controller had the same amount of switching, and the commands were generated in an open-loop manner. The discrete switching commands for tracking were generated only in the indicated period. Thus, the response due to residual oscillations after the tracking region can be neglected in this experiment.

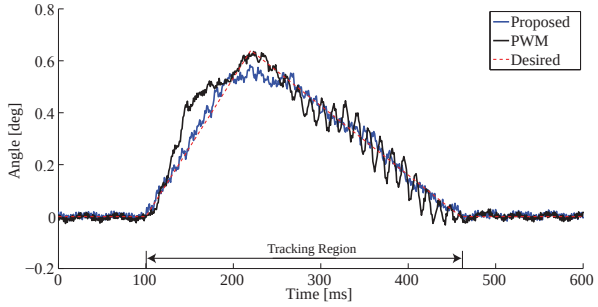
A total of four different velocity profiles were tested as shown in Figure 82. All velocity profiles were in the range of a single PZT activation level. The responses of both the proposed and PWM controllers were comparable when the desired motion was a ramp input as shown in Figure 82(a). It can be observed from Figures 82(b) - 82(d) that the proposed controller performed better than the the conventional PWM controller when the velocity of the desired motion was not constant. The proposed discrete controller exhibited better RMS (Root Mean Square) position errors than the PWM controller as demonstrated in Table 13. The proposed discrete switching controller successfully tracked the desired profiles, although minor position errors exist due to quantization effects and slow control frequency. However, the conventional PWM controller experienced significant position errors, particularly when the velocity varied consistently. This is because the PWM controller does not account for dynamics.

Table 13: RMS Position Errors [deg]

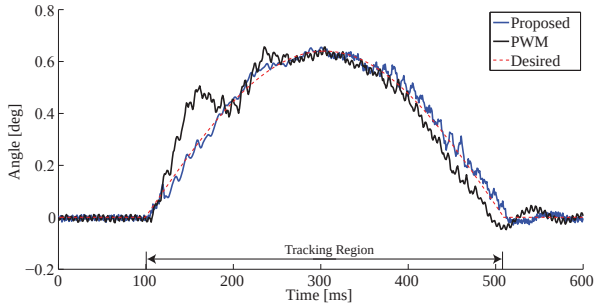
Test #	Tracking time	# of pulses	ω_c	Proposed	PWM
1	500 ms	18	36.0 Hz	0.0259	0.0309
2	360 ms	20	55.5 Hz	0.0319	0.0644
3	407 ms	20	49.1 Hz	0.0210	0.0744
4	350 ms	20	57.1 Hz	0.0368	0.1862



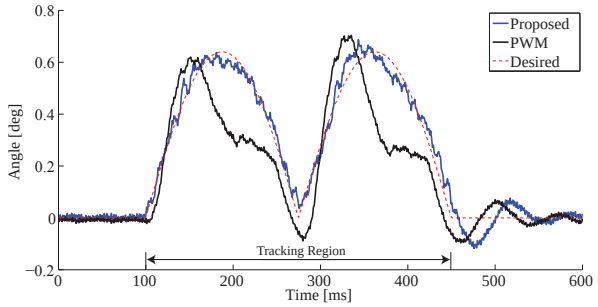
(a)



(b)



(c)



(d)

Figure 82: Comparisons between the proposed discrete controller and the conventional PWM controller. (a) Test #1. Ramp signal. (b) Test #2. Half period of a triangle wave. (c) Test #3. Half period of a sinusoidal signal. (d) Test #4. Absolute period of a sinusoidal signal.

6.3.4 Vibration Suppression Commands

The proposed method is designed to match the velocity of the object of interest only during the exposure window, unlike other visual tracking methods for feeding video. It was considered that acquisition of few images in one PZT increment is sufficient, since the FOV of the camera is not significantly changed. After the robotic vision system obtains images, the tracking commands are no longer necessary. Therefore, while the camera orientation system reaches the desired quantized position, residual vibrations are suppressed in the flexible system. With this approach, the number of switching can be further reduced, which contributes to the prevention of fatigue in the thin-layered mechanism. To achieve vibration suppression, the real and imaginary components of the residual vibrations must be zero. Therefore, a set of vibration suppression commands can be determined as follows:

$$\sum_{i=1}^{n} A_i \cdot e^{-\zeta\omega_n(t-t_i)} \cdot \cos\phi_i = 0 \tag{62}$$

$$\sum_{i=1}^{n} A_i \cdot e^{-\zeta\omega_n(t-t_i)} \cdot \sin\phi_i = 0 \tag{63}$$

The solutions satisfying both (62) and (63) are not unique due to nonlinearity in the equations. Schultz and Ueda investigated vibration suppression commands using a quantized system for point-to-point movements in [96]. They used a nested-loop algorithm to first find a pattern of A_is, and then used a numerical solver to determine a set of t_is. One of their conditions was relaxed that the system is undamped ($\zeta = 0$); Thus, (62) and (63) can be expressed as $\sum_{i=1}^{n} A_i \cos\phi_i = 0$ and $\sum_{i=1}^{n} A_i \sin\phi_i = 0$, respectively. However, their algorithm may return a different command set when given the same desired point-to-point movement because of the iterative nested loops and the numerical solver. For this reason, they concluded in [96] that no statements in regard to existence, nonexistence, or uniqueness of the solutions can be made using classical mathematical methods. Solutions to systems of nonlinear equations are still

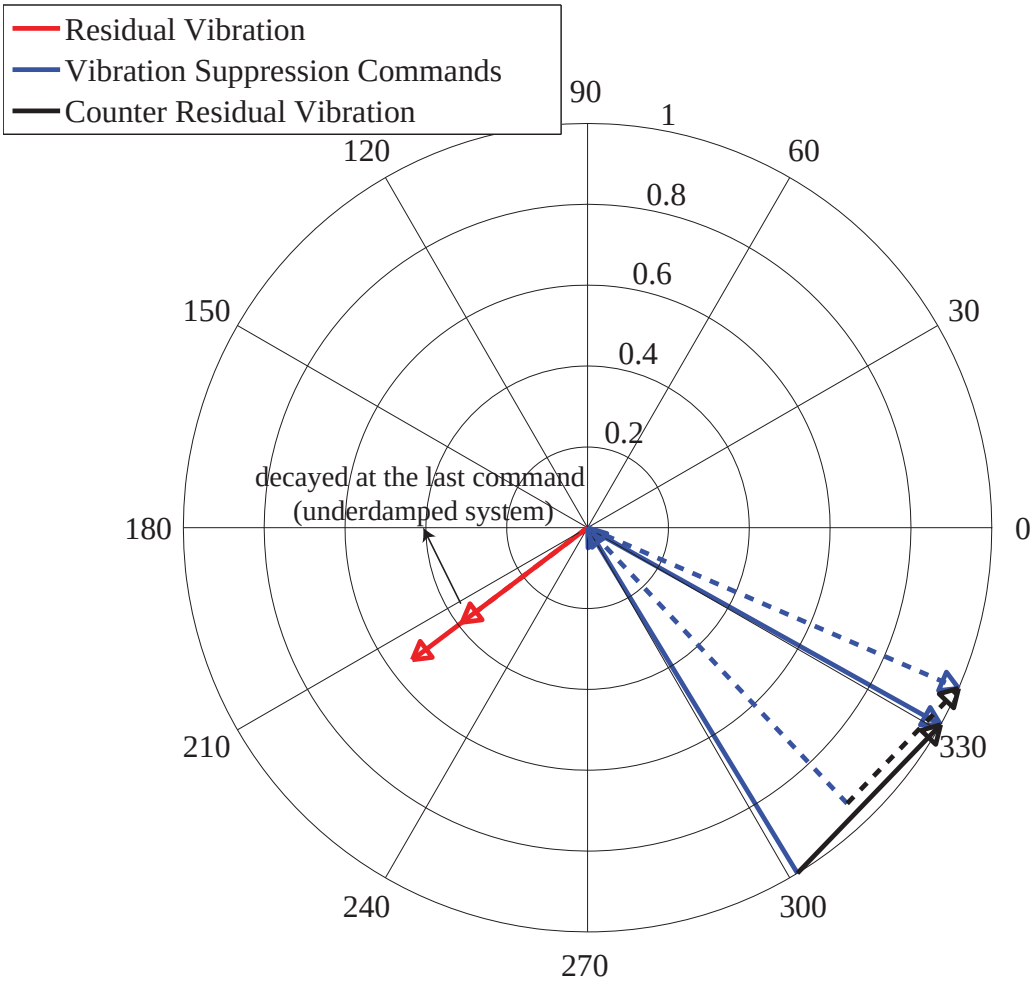


Figure 83: Representation of residual vibrations in the complex plane. The solid lines represent an example for an undamped system. The dashed lines represent an example for an underdamped system.

a topic of mathematics research.

In this study, the vibration suppression commands are determined in a minimum switching manner after the tracking region. This introduces additional constraints:

$$\sum_{i=1}^{m_1+m_2} A_i = 1 \text{ such that } (m_1 + m_2) \text{ is minimized,} \tag{64}$$

where m_1 and m_2 are the number of switching in tracking and vibration suppression commands, respectively.

If the last point of the tracking region is above the midpoint $\frac{\Theta_{k+1}+\Theta_k}{2}$, an additional

even number of switches is required because $\sum_{i=1}^{m_1} A_i = 1$. If the last point of the tracking region is below the midpoint $\frac{\Theta_{k+1}+\Theta_k}{2}$, an additional odd number of switches is required because $\sum_{i=1}^{m_1} A_i = 0$. In this case, one additional switch is not the minimum number that can satisfy (62) and (63), as B_{sum} is not an integer and decays with time. The first ON switch after the exposure window in the tracking commands is included to make $\sum_{i=1}^{m_1} A_i = 1$.

Let B_{track} and Φ_{track} be the amplitude and phase of the residual vibrations due to m_1 switches during the exposure window. Since B_{track} is approximately $[0, \frac{|A_i|}{2}]$, introducing two additional switches is sufficient to suppress the residual vibration. This can be explained using the complex plane representation.

Figure 83 shows the complex plane and the red arrow represents the residual vibration due to tracking commands. The solid lines represent an example for an undamped system and the dashed lines represent an example for an underdamped system. The residual vibration can be suppressed in the next quadrant where Φ_{track} exists. For vibration suppression, the sum of vectors in the complex plane must be equal to zero. A counter residual vibration with an amplitude and phase of B_{track} and $\pi + \Phi_{track}$, respectively, can be generated in the following quadrant by having two additional switches. It can be observed from the complex plane that the discrete switching commands for vibration suppression also satisfy (46) and (47).

The two additional phases ϕ_{vs1} and ϕ_{vs2} to suppress the residual vibration for an undamped system ($\zeta = 0$) can be given as

$$\phi_{vs1} = \Phi_{track} + \frac{\pi}{2} - \frac{\alpha}{2} \tag{65}$$

$$\phi_{vs2} = \phi_{vs1} + \alpha \tag{66}$$

where

$$\alpha = \text{acos}\Big(1 - \frac{(B_{track})^2}{2}\Big) \tag{67}$$

Timing of the two commands, t_{vs1} and t_{vs2}, to suppress the residual vibration can

be determined using the following relationship:

$$t_i = \phi_i/\omega_n \tag{68}$$

For an underdamped system, the amplitudes of the residual vibration and the first switch for vibration suppression are exponentially decaying, which introduces additional nonlinear terms compared to the undamped system. As a result, the phases of each switch for an underdamped system exist slightly after that for undamped system. The solutions to timing of the vibration suppression commands can be determined using numerical methods that guarantee convergence. The phases of an undamped system can be used as a reference to find a local minimum.

The entire set of discrete switching commands for both tracking and vibration suppression can be determined as

$$u(t) = \sum_{i=1}^{m_1+2} A_i \cdot 1(t - t_i), \tag{69}$$

where

$$\boldsymbol{A} = [A_1 \ \cdots \ A_{m_1} \ -1 \ 1], \tag{70}$$

$$\boldsymbol{t} = [t_1 \ \cdots \ t_{m_1} \ t_{vs1} \ t_{vs2}]. \tag{71}$$

Figure 84 shows one possible visual-motor application due to the suppression of vibrations after the exposure window. Tracking switching commands run first for object tracking, and vibration suppression switching commands follow for background image capture. The repetition of this switching in a short period of time allows images of both the object of interest and the background to be updated. In addition, the generated commands in a single quantized increment are reusable if the velocity of the tracking profile is constant in multiple quantized increments. This is because no residual vibrations exist when the tracking controller is initiated.

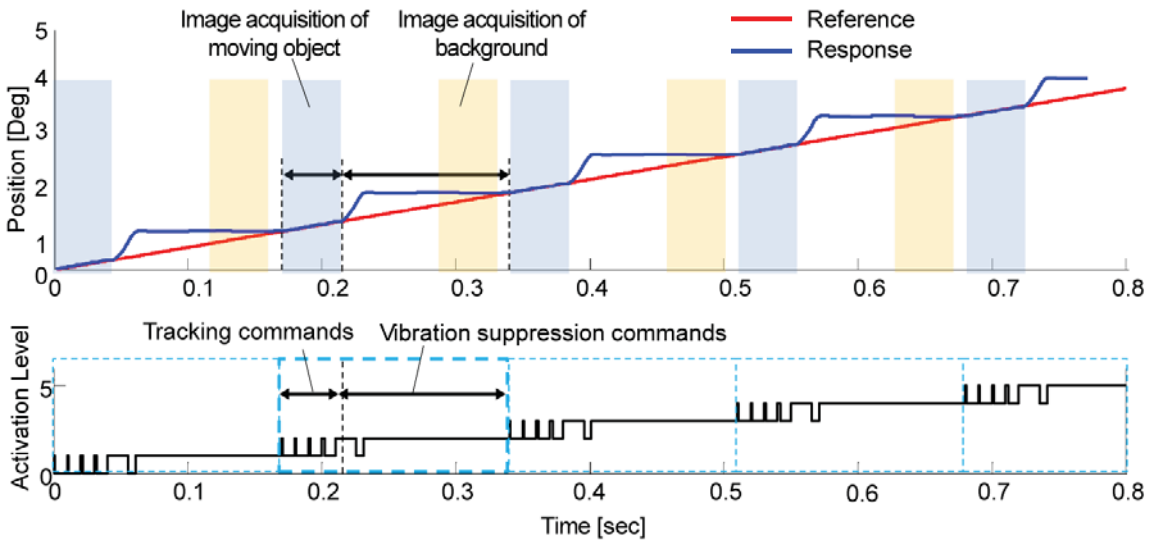


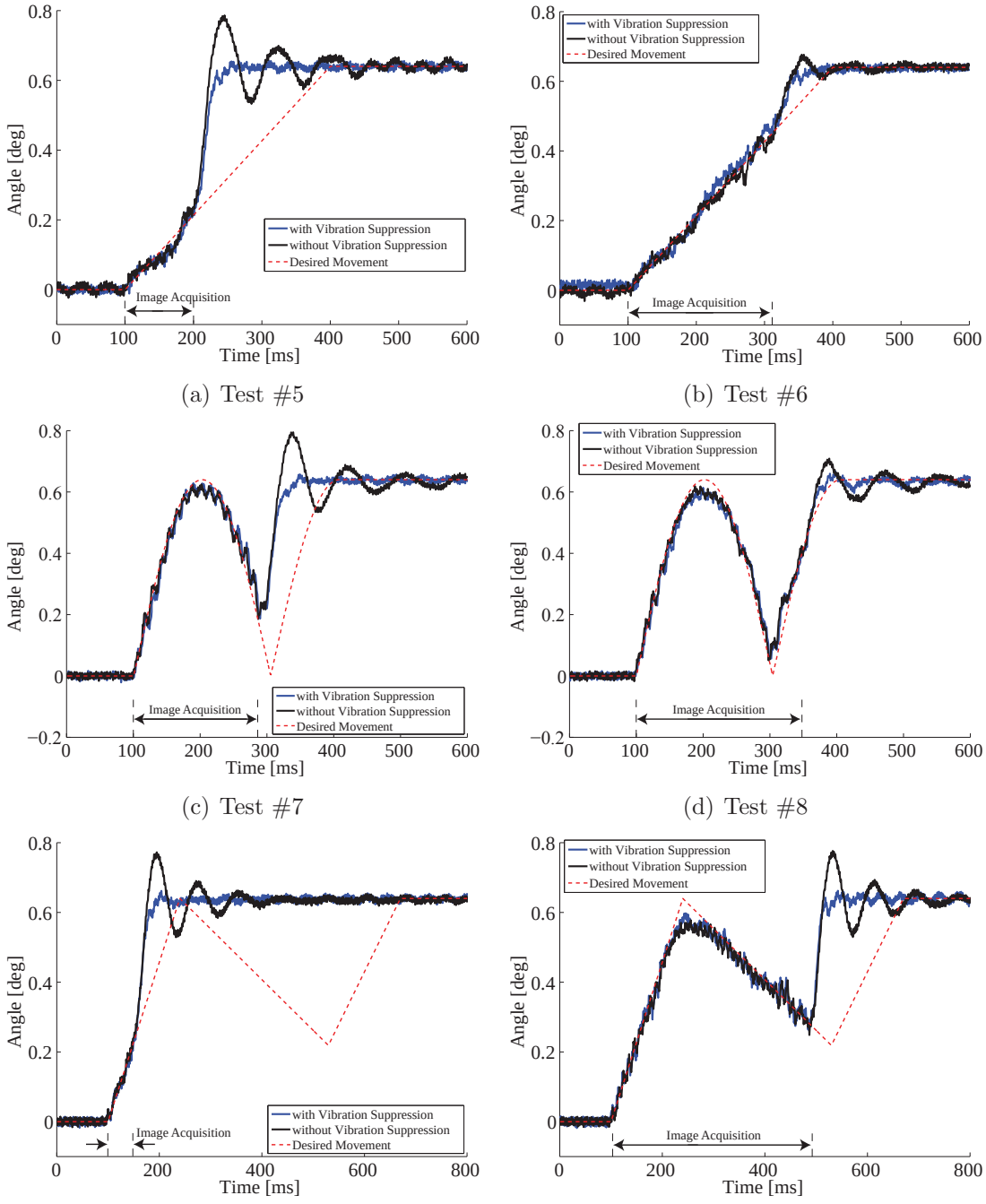
Figure 84: Possible combination of saccade-like movements and smooth pursuit when the desired velocity is constant.

6.3.5 Validation of the Controller for Vibration Suppression

To validate the proposed method, the responses of the system were measured using the same setup shown in Figure 81. The proposed method consisting of both tracking and vibration suppression controllers was compared with the method involving the tracking only controller discussed in the previous section. For the tracking only controller, no further switches were generated after the image acquisition period. The experimental conditions are listed in Table 14.

Three different velocity profiles were tested and two different image acquisition times were used for each velocity profile. Tests #5 and #6 had the same desired motion, while the image acquisition times were 100 ms and 210 ms, respectively. Tests #7 and #8 had the same desired motion, while the image acquisition times were 180 ms and 240 ms, respectively. Tests #9 and #10 had the same desired motion, while the image acquisition times were 90 ms and 390 ms, respectively.

It can be observed in Figure 85 that the proposed controller successfully suppressed the residual vibration after the image acquisition period. For all cases, the



(a) Test #5 (b) Test #6

(c) Test #7 (d) Test #8

(e) Test #9 (f) Test #10

Figure 85: Comparisons between the proposed method, consisting of both the tracking and vibration suppression controllers, and the tracking only controller. Tests #5 and #6 have the same desired motion with different image acquisition times. Tests #7 and #8 have the same desired motion with different image acquisition times. Tests #9 and #10 have the same desired motion with different image acquisition times.

Table 14: Test Conditions

Test #	Desired motion time	Image acquisition time	ω_c
5	300 ms	100 ms	60 Hz
6	300 ms	210 ms	60 Hz
7	310 ms	180 ms	97 Hz
8	310 ms	240 ms	97 Hz
9	510 ms	90 ms	59 Hz
10	510 ms	390 ms	59 Hz

proposed controller has an improved settling time. It also reduced the switching indicated in Table 15. Although the residual vibrations were not significant using the tracking only controller in test #6 as shown in Figure 85(b), the proposed controller exhibited a better performance. The responses after the image acquisition period without the vibration suppression commands vary because they are dependent on the tracking commands. It can be observed again that the proposed tracking controller was effective during the image acquisition period with low control frequency.

Table 15: Experimental Results of the Proposed Method

Test #	Improvement in settling time	Reduction in number of switching
5	209	23
6	43	8
7	393	16
8	97	8
9	208	28
10	239	5

6.4 Experimental Setup for Tracking

Figure 86 shows the experimental setup to validate the study. A red checkerboard is used as an object of interest. The object of interest is positioned in the horizontal direction by a linear stage from Eshed Robotek. The device contains a Pittman GM8712G759 24V DC motor equipped with a 19.5:1 gearbox and a 0.435 diameter pulley. An embedded controller is used to control the DC motor to generate motion.

The robotic vision system employs a Ximea subminiature camera MU9PC-MH, which obtains images at 12.5 fps with UVGA or 28 fps with VGA. The subminiature camera is operated in a trigger mode and the camera provides active exposure information to external devices. Images are obtained using a global shutter mode. The image data is sent via USB 2.0 to the host PC running MS Windows 7 64 bit version with Intel i7-2600K 3.40 GHz CPU and 12.0 GB Ram.

The moving objects can be tracked and their motion profiles can be estimated by various means [21, 103]. However, motion estimation of moving objects is not in the scope of this study. For this reason, it was assumed that motion of the moving object of interest is known.

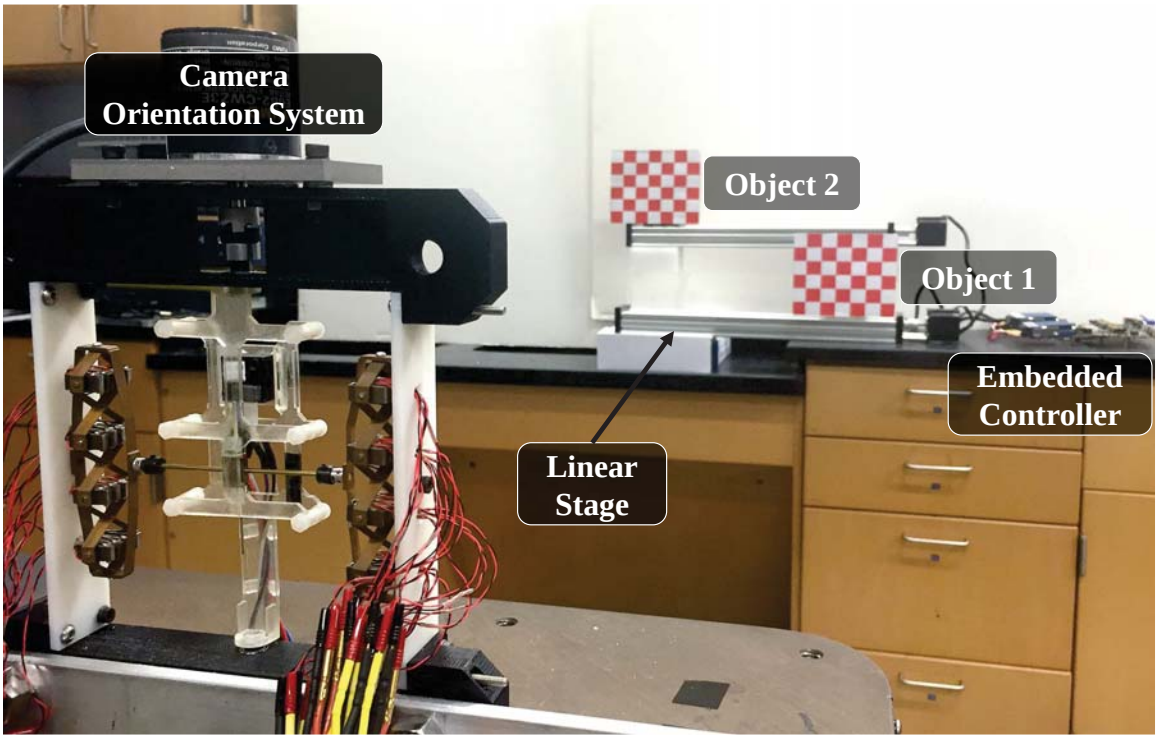


Figure 86: Experimental setup. The proposed discrete switching commands are given to the system for tracking the object of interest positioned by the linear stage.

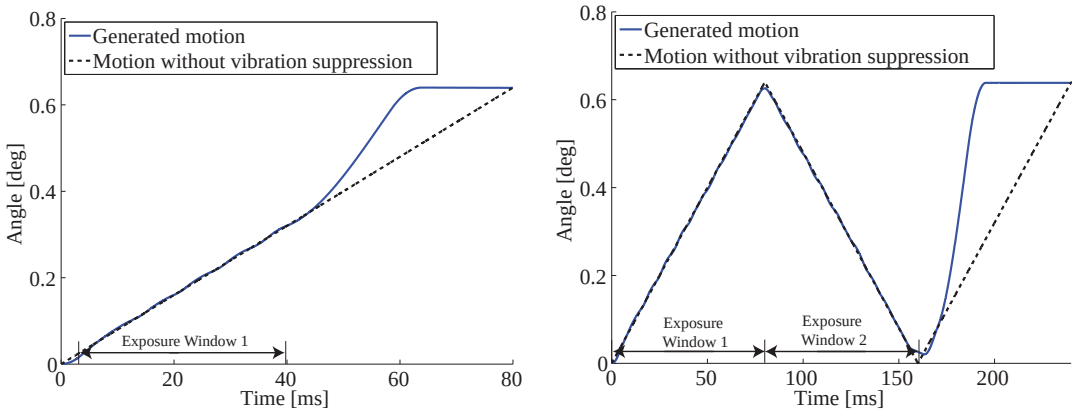


Figure 87: Generated movements to track objects of interest. The discrete switching commands for vibration suppression were generated after the exposure window. (a) Tracking Test #1. One image is obtained. (b) Tracking Test #2. Two images are obtained

6.5 Results and Discussions

The discrete switching commands were given to the system to match the velocity of the object during the exposure window and vibration suppression commands followed. If the velocity of the camera orientation system matches that of the object during the exposure window, no blurriness would be observed on the object in the obtained images. Two different experiments for tracking objects were conducted and the experimental conditions are listed in Table 16 and shown in Figure 87.

Table 16: Experimental Conditions for Tracking Objects

Tracking Test #	Exposure time	Image size	Number of obtained images	Mean Velocity
1	35 ms	VGA	1	8 deg/sec
2	80 ms	UVGA	2	8 deg/sec

6.5.1 Tracking a Single Object

For tracking test #1, the camera orientation system obtained a single image while tracking a single moving object. The generated motion profile is shown in Figure 87(a) and the image was obtained during the indicated exposure window. The exposure

(a) (b) (c)

Figure 88: Tracking test #1. The exposure time was 35 ms. The images were obtained under various conditions for comparison. (a) Obtained image while tracking. (b) Object in motion with camera at rest. (c) Reference image.

time of the camera was 35 ms and the image resolution was VGA. The images are obtained under various conditions for comparison and the cropped images are shown in Figure 88.

Figure 88(b) shows the image obtained when the object was in motion while the camera was at rest. The motion of the object is not negligible, since blurriness is observed. Figure 88(c) shows the reference image obtained when both the object and camera were at rest. Figure 88(a) is the image obtained by using the proposed method, and its quality is comparable with the reference. Since no significant blurriness is observed in the image obtained by the proposed method, it can be concluded that the controller successfully matched the motion of the object of interest during the exposure window.

The results can also be evaluated in a quantitative manner. Since a checkerboard image was used as an object, the gradients of the images in the horizontal direction were computed for evaluation. If the checkerboard image is blurry, the gradient map would show weak peaks, because high frequency information is degraded. In contrast, the gradient map of a clean image would show strong peaks.

Figure 89 shows the absolute values of gradients in the middle line of the images. It can be observed that the image obtained by the proposed method exhibits comparable peak intensities with the reference. Although some peaks of the proposed method are

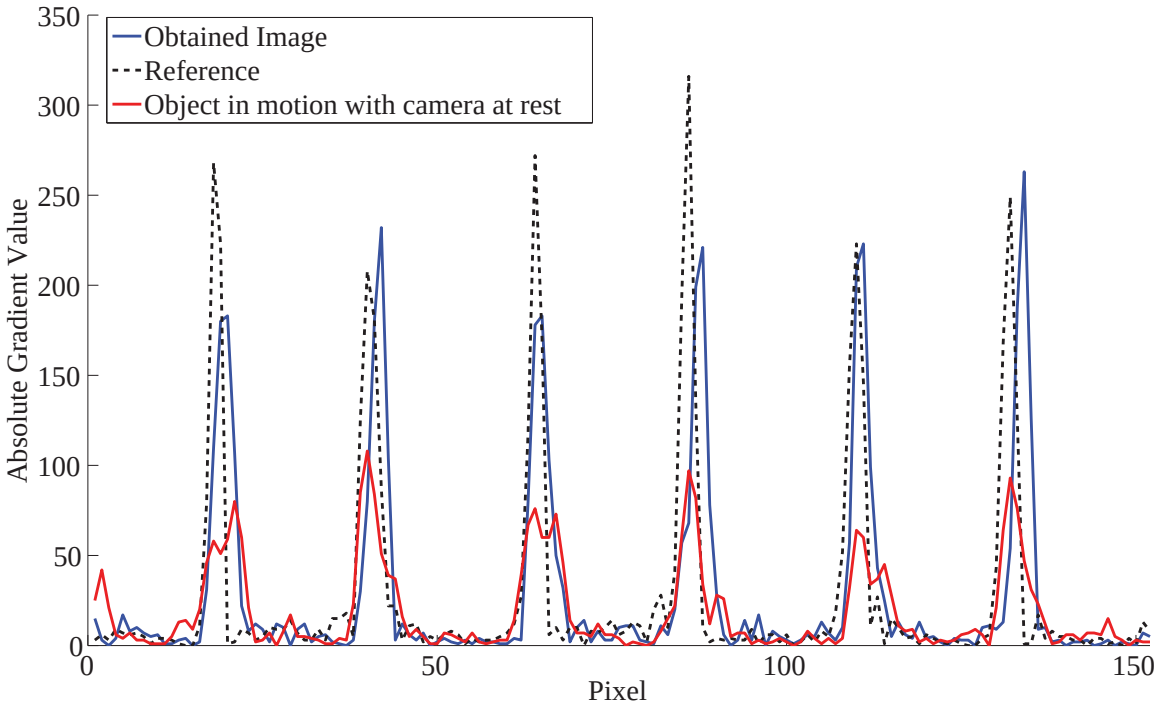


Figure 89: Comparisons of the absolute of the gradient values. Gradients of the middle line of the images are depicted.

slightly weaker than those of the reference image, half of the peaks exhibit similar intensities. The image obtained when only the object was in motion exhibits low peak values due to blurriness. The peak values were computed over the entire domain and the mean values are listed in Table 17. The proposed method was effective, since the mean peak value in the image obtained by the proposed method was comparable with that in the reference image.

Table 17: Mean Peak Values of Image Gradients (Tracking a Single Object)

Proposed Method	Reference Image	Object in motion & camera at rest
213.79	231.08	115.79

6.5.2 Tracking Two Objects

For tracking test #2, the camera orientation system obtained two images during the course of motion. Two objects, 'Object 1' and 'Object 2', traveled in opposite

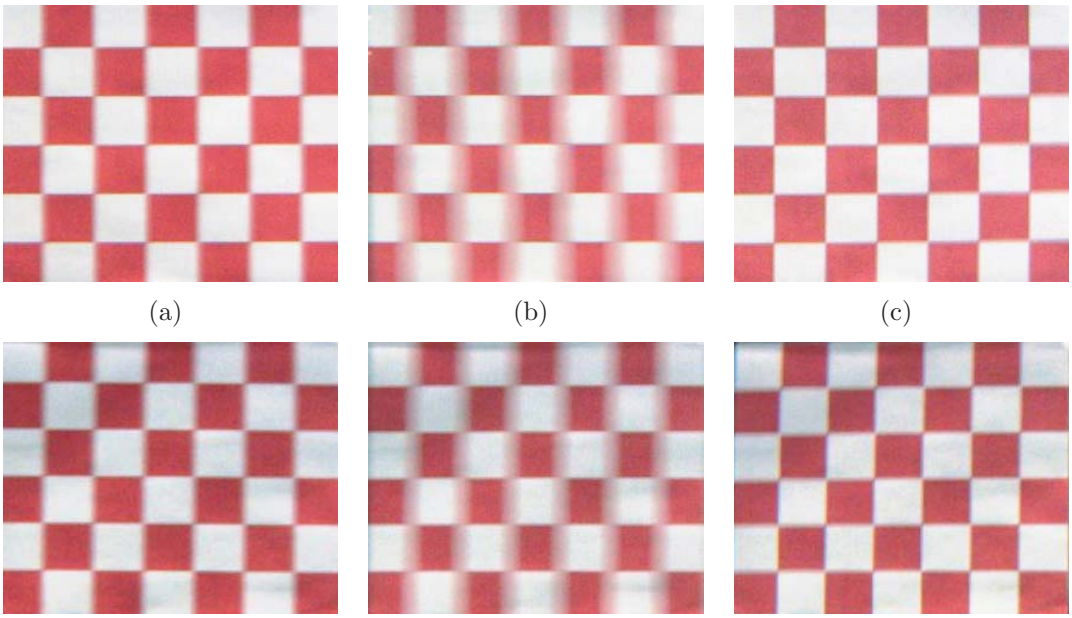


Figure 90: Tracking test #2. The exposure time was 80 ms. The system tracked two moving objects in a short period of time and two images were obtained. (a) 'Object 1' captured in 'Exposure Window 1'. (b) 'Object 1' captured when the camera was at rest. (c) Reference image of 'Object 1'. (d) 'Object 2' captured in 'Exposure Window 1'. (e) 'Object 2' captured when the camera was at rest. (f) Reference image of 'Object 2'.

directions. The generated motion is shown in Figure 87(b) and shows the proposed discrete controller can possibly be used for tracking multiple objects in a short period of time. The camera orientation system attempted to match the velocity of 'Object 1' in 'Exposure Window 1' and the velocity of 'Object 2' in 'Exposure Window 2'. In order to test the proposed method rigorously, the exposure time was kept longer than that in tracking test #1 as indicated in Table 16.

The obtained images in tracking test #2 are shown in Figure 90. The top row shows the cropped images of 'Object 1' and the bottom row is the cropped images of 'Object 2' obtained under various conditions. The first column shows the images obtained by the proposed method and the last column shows the reference images obtained at rest. The middle column shows the images when the object was in motion while the camera was at rest.

It can be observed that the images obtained by the proposed method exhibit comparable quality with the reference images; however, minor blurriness exists. Although the camera orientation system successfully generated the desired movements, minor blurriness cannot be avoided. This is because the camera orientation system experiences rotational motion while the object of interest experiences linear motion. This effect became significant in tracking test #2 due to the long exposure time. Figures 90(b) and 90(e) show that movements of the objects are not negligible. Table 18 demonstrates the mean peak values of the image gradients. Although the peak values in the image obtained by the proposed method are weaker than those in the reference image, the intensities are preserved by tracking the objects. Thus, it can be concluded that the proposed discrete switching controller was effective.

Table 18: Mean Peak Values of Image Gradients (Tracking Two Objects)

Object #	Proposed Method	Reference Image	Object in motion & camera at rest
1	131.22	192.06	73.7
2	124.54	178.72	63.25

6.6 Summary

This chapter presented a discrete switching controller to generate an arbitrary velocity profile for a quantized, compliant actuator-driven robotic vision system. The proposed method is designed to match the velocity of the object of interest only during the exposure window. During the exposure window, tracking motion is generated by combining and optimizing a series of pulse-like commands at each primitive. Then, residual vibrations are suppressed while the camera orientation system reaches the desired quantized position. Although classical feedback controllers with PWM quantization are capable of generating an arbitrary velocity profile, they causes high-frequency switching that would damage the thin, compliant mechanical structure. The proposed controller is an open-loop method and requires less frequent switching

than the existing methods.

The proposed method was investigated under various motion profiles to verify its effectiveness and compared with the classical method. For generating the desired velocity profiles, the proposed method exhibits better RMS position errors than the classical method. After the exposure window, the proposed controller shows an improved settling response, as it accounts for the system dynamics. Using the proposed controller, the camera orientation system tracked a moving object and obtained images. The image quality of the object of interest was comparable with the reference image, validating the effectiveness of the proposed method.

CHAPTER VII

CONCLUDING REMARKS

This thesis tightly synchronized motion control and image processing in the fast-moving robotic vision system, inspired by the neuro-ocular mechanism. By merging the system dynamics area and image processing area, two representative modes in the biological eye were reproduced in a single robotic eye platform, while existing camera platforms focused on a single function. The dynamics-based image processing methods were developed in coordination with inherently discrete and rapid movements in a fast-moving robotic eye. The proposed de-blurring method predicts the blur kernel based on the dynamic model of the robotic eye and movements induced by the control system. This mechanism was the key enabler of real-time panoramic image stitching with simultaneous motion de-blurring in a dynamic vision system, allowing for generic image sensors with a standard frame rate and significantly less computational load, and requiring no motion sensors. Conventional image de-blurring and stitching methods have analyzed image properties by various means. However, in general, they require much computation time or are not robust to blurry images for post-processing.

In the neuromotor system, the movements are generated by muscles that are essentially quantized, compliant actuators. Motion is achieved by discrete activation and thus muscles cannot produce continuously variable outputs. In an engineering system, this can be considered as velocity matching control of an impulsive dynamical system. The discrete switching controller was developed for the robotic eye driven by quantized, complaint actuation systems to match the velocity of the object of interest during an exposure window. The desired trajectory was achieved by combining and

optimizing a series of pulse-like commands at each primitive. After image acquisition, residual vibrations were suppressed while the camera orientation system reached to the desired quantized position. The proposed controller requires less switching than the conventional methods that helps avoid fatigue in the mechanism.

Several experiments were conducted to validate the dynamics-based image processing method and demonstrate the effectiveness of this ocular physiology-inspired approach. A system architecture was designed to realize the developed algorithms and achieve real-time performance by parallelism and visual-motor coordination. The improvement on processing times by the dynamics-based approaches was demonstrated. The quality of resultant images were compared with that of prior methods. The contributions of the proposed methods were also demonstrated by quantitative evaluation. The discrete switching controller reproduced smooth pursuit like movements, enabling the robotic eye to track an object of interest.

The research described in this thesis provides original contributions:

- A dynamics-based image de-blurring method was developed for a robotic vision system. The computation times of the dynamics-based approach was 45.1 times faster on average than the conventional approaches. In addition, the performance metric of the proposed method was 31.29 % better than the other algorithms on average. The proposed method was also evaluated by OCR technology and the text recognition rates of the proposed method was 57.8% point better than that of conventional methods.

- The dynamics-based approach was extended to the generation of panoramic images. Using the dynamics information, the algorithm was robust to degraded blurry input images, while panoramic images generated by the conventional methods exhibited significant artifacts. The performance metric of the proposed method was 41 % better than the other algorithms on average when blurry

images were given.

- A system architecture was developed to control multiple PZT actuators and enhance computation times of image processing. A blur kernel was estimated in a parallel manner during the exposure window. Therefore, the blurry image and the blur kernel were obtained at the same time. This architecture allowed for the robotic vision system to generate panoramic images in the course of rapid motion while incorporating the de-blurring technique in real-time with 30 fps for VGA-size images.

- A discrete switching controller was developed to generate an arbitrary velocity profile with reduced switching. The proposed controller was designed to match the velocity of an object of interest only during exposure windows. Thus, after images were obtained, vibration suppression commands were followed to reduce switching further. In comparison with the conventional PWM method, the proposed method showed improved RMS position errors during the exposure windows and enhanced settling times after image acquisition.

This thesis has presented original contributions, but a variety of studies can be addressed in future research.

- This thesis used a robotic vision system that has a single-DOF. The impact of verification experiments may be limited. Future work could investigate the developed dynamics-based methods for a multi-DOF robotic vision system.

- The robotic vision system employed a single camera. The established techniques can be extended to stereo camera systems to make use of depth information.

- One key assumption in this study was that the system was not experiencing external disturbances. Robust or isolation methods to external disturbances for the dynamics-based approaches can be examined.

- Visual algorithms such as recognition may be considered to investigate psychology in visual perception using a robotic eye system.

APPENDIX A

MODIFIED BLUR KERNEL FOR TRANSLATIONAL MOTION

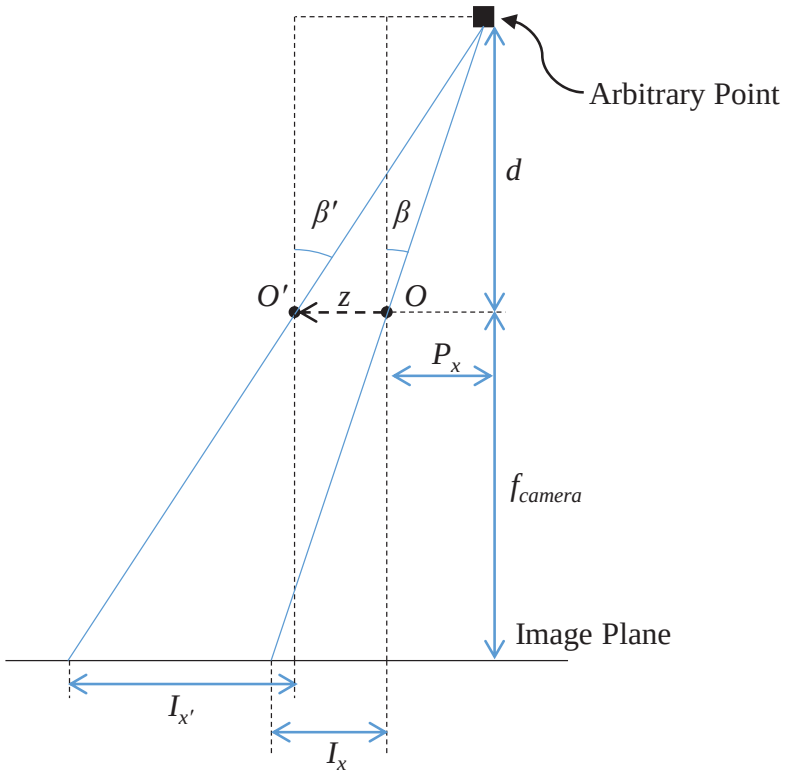


Figure 91: Pinhole camera model for translational motion

The camera orientation system used in this study can generate only pure rotational motion. If the camera orientation system experiences translational movements during the exposure window, a modified blur kernel can be introduced incorporating depth information.

Figure 91 shows the model of a pinhole camera experiencing translational motion in 2D space. An arbitrary point in the real world appears on the image plane at I_x,

which is a horizontal distance from the image center.

$$I_x = f_{camera} \tan(\beta) \tag{72}$$

where f_{camera} is the focal length and β is the angle between the normal axis of the image plane and the position vector of an arbitrary point. Since,

$$\tan(\beta) = \frac{P_x}{d} \tag{73}$$

where P_x is the horizontal distance from the pinhole in the real world and d is the depth. Therefore,

$$I_x = \frac{f_{camera}}{d} P_x \tag{74}$$

Let the camera moves by z as shown on the figure so the pinhole is moved from O to O'. To estimate motion of an arbitrary point, ΔI_x variation of the position on the image plane needs to be defined. Similarly, $I_{x'}$ the horizontal distance from the image center of the moved camera is

$$I_{x'} = \frac{f_{camera}}{d}(P_x + z) \tag{75}$$

Therefore,

$$\Delta I_x = I_x - I_{x'} = \frac{f_{camera}}{d} z \tag{76}$$

Although f_{camera} and z are constant for all pixels, position variation of a pixel is dependent on the depth. Therefore, a spatially invariant kernel model is not valid when a translational movement takes place during the exposure window.

Figure 92 shows the experimental setup to test motion blur caused by translational motion. A linear stage is used to generate translational motion during the exposure window. For this experiment, the PZT actuators in the camera orientation system are not operated. Since motion blur caused by translational motion is dependent on the depth of the scene, a Kinect sensor is used to obtain a depth map.

Figure 93 shows the tested scene where two checkerboards are placed with different distances from the camera orientation system. Figure 94 is an image showing the checkerboards with depth information.

Figures 95(a) - 95(b) are the obtained blurry images. It can be observed that the degrees of blurriness on two images are different due to different distances to the vision sensor. Figures 96(a) - 96(b) are the de-blurred images obtained by using a spatial invariant blur kernel determined by a distance to the red checkerboard. The blur kernel size is 27×27. It can be observed that the image of red checkerboard is improved while the image of the blue checkerboard exhibits artifacts. Figures 97(a) - 97(b) are the de-blurred images obtained by using modified blur kernels. The modified blur kernels are generated by incorporating the depth information. The blur kernel sizes for the red and blur checkerboards are 27×27 and 21×21, respectively. It can be observed that the image on both checkerboards is improved.

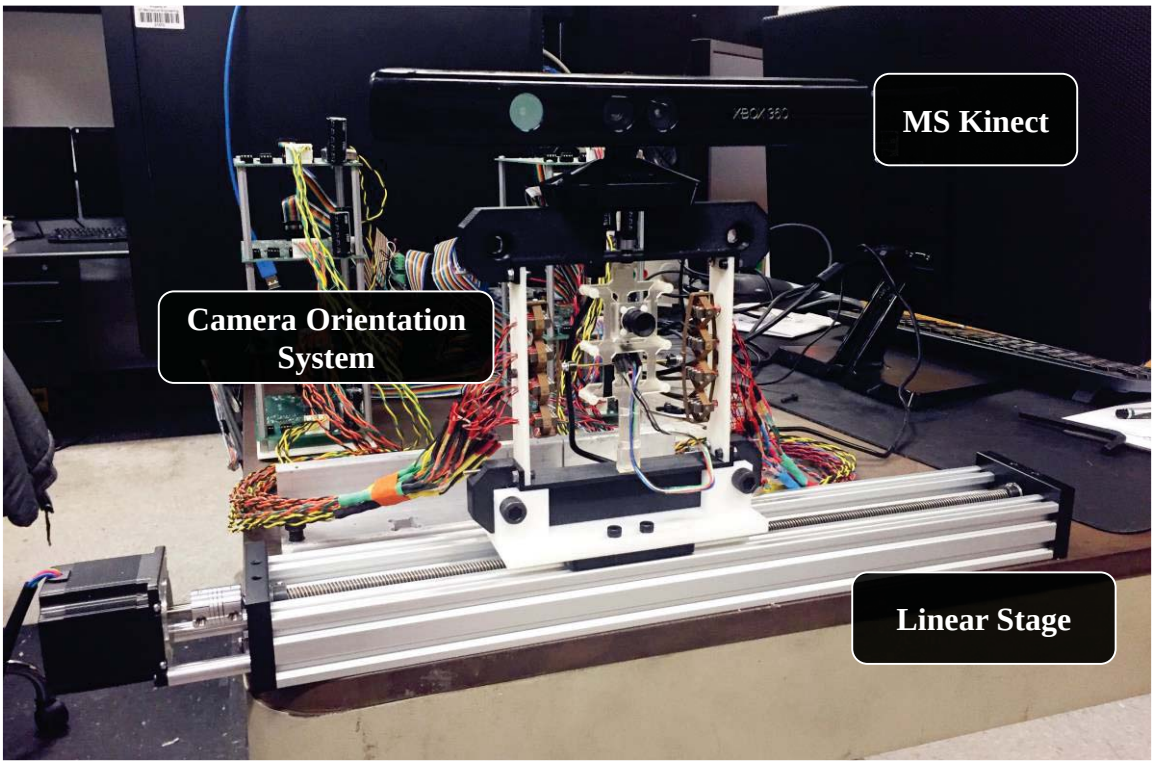


Figure 92: Experimental setup to test the de-blurring method for translational motion.

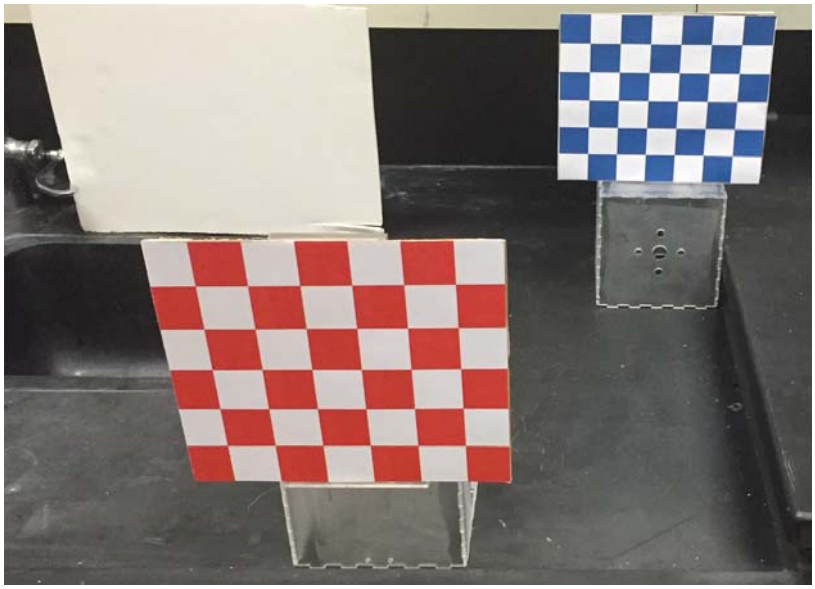


Figure 93: Scene to test motion blur caused by translational motion. Two checkerboards are placed with different distances to the vision system.

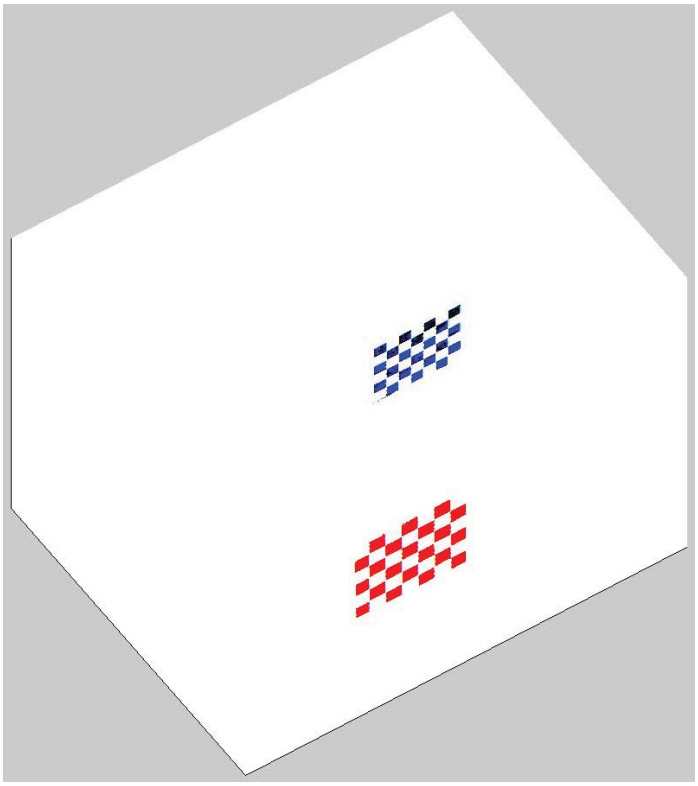
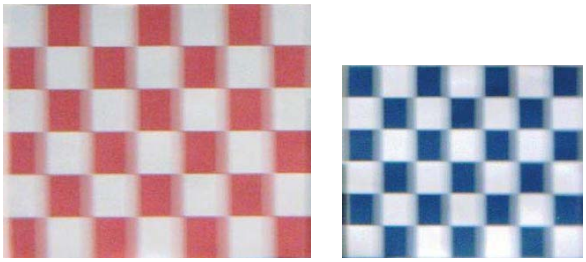
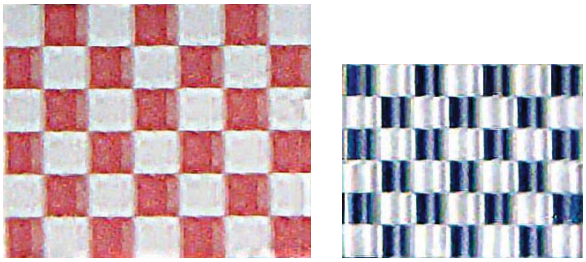


Figure 94: Image with depth information.



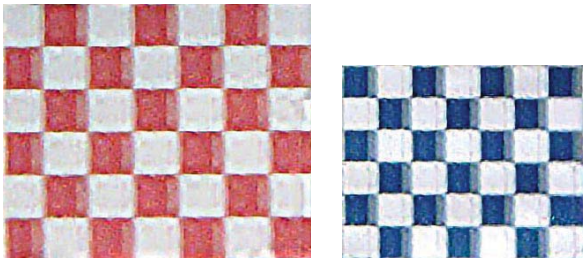
(a) (b)

Figure 95: Obtained blurry images.



(a) (b)

Figure 96: De-blurred images acquired using a spatially invariant kernel. The blur kernel size was 27 × 27.



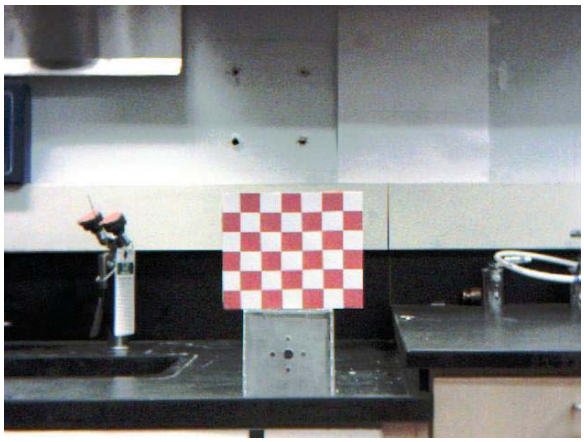
(a) (b)

Figure 97: De-blurred images acquired using modified blur kernels. The blur kernel size for the red checkerboard was 27 × 27. The blur kernel size for the blur checkerboard was 21 × 21.

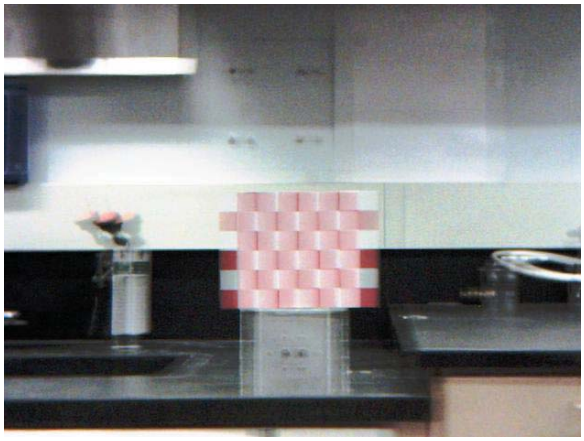
APPENDIX B

EVALUATION OF THE DE-BLURRING METHODS USING A CHECKERBOARD

The image de-blurring methods were evaluated by various means in Chapter 3. In this appendix, the de-blurring methods are evaluated using a red checkerboard as shown in Figure 98.



(a)



(b)

Figure 98: A red checkerboard is used to evaluate the de-blurring methods. (a) clean image obtained at rest. (b) blurry image obtained in the course of motion

Figure 98(a) is the reference image obtained at rest and Figure 98(b) is the blurry image obtained in the course of motion. On the red checkerboard region, pixel values between the red and white colors are clear. If blurriness exists, colors would be blended, and thus pixel values of those regions would be different from that of red and white colors. For evaluation, R values are extracted from the de-blurred RGB image on the red checkerboard region. Then, number of white pixels are counted; pixel value $\in [0, 255]$. Pixel values exceeding 235 are considered as white. The better image would exhibit more white pixels.

Figure 99 shows the de-blurred images by the proposed and compared methods. The compared algorithms include parameters that need to be manually defined by the users. For a fair comparison, the parameters are adjusted a number of times to find the best possible results. It should also be noted that the compared algorithms include additional image processing such as noise reduction and sharpening, while no additional image processing is applied for the proposed method.

Figure 100 shows the number of white pixels counted on the checkerboard region. The total number of pixels on the red checkerboard region is 18500. It can be observed that the number of white pixels in the reference image is close to half of the total pixel number on the checkerboard region. It can be observed that the proposed method shows the best results and all methods recovered the blurry image in part.

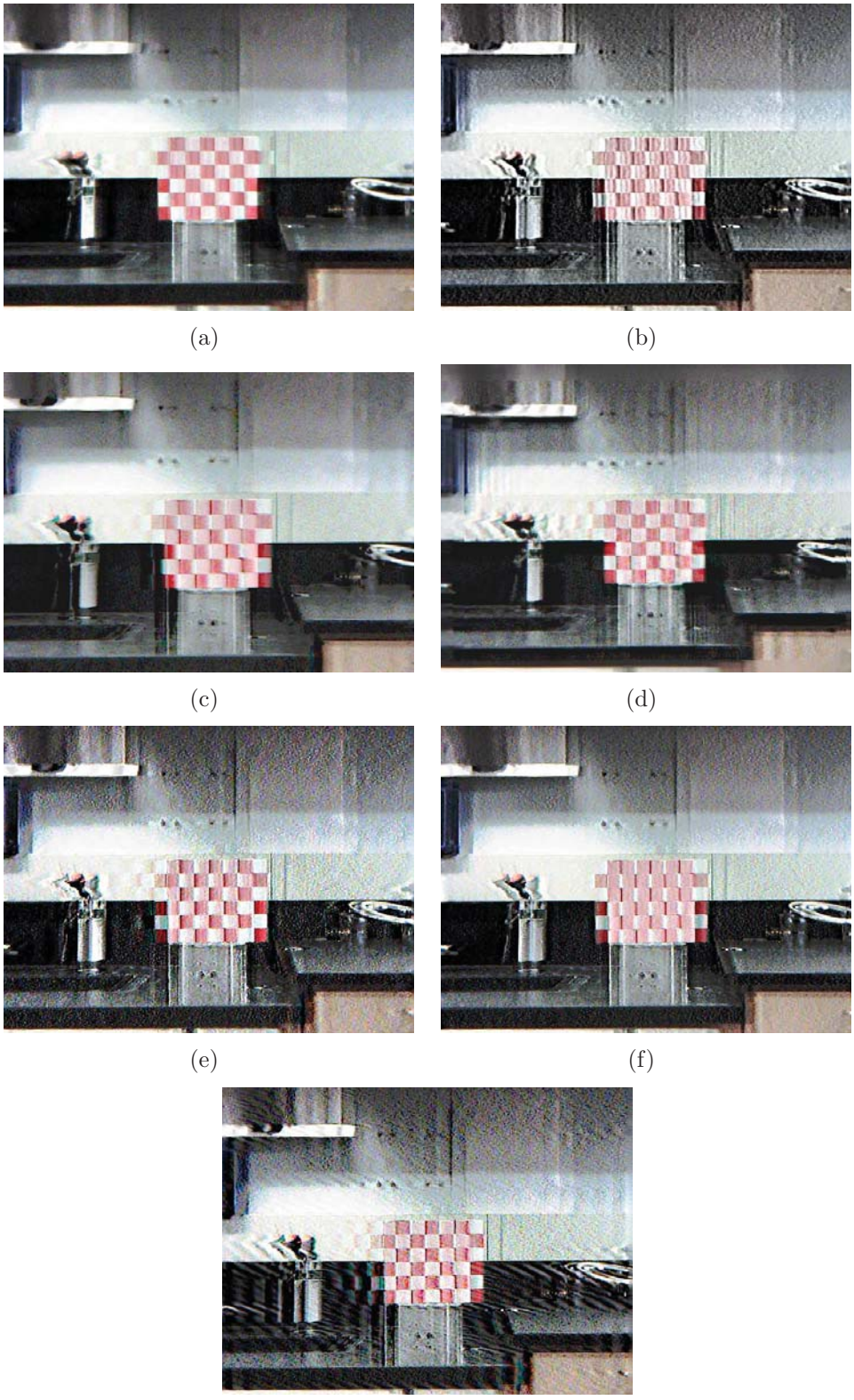


Figure 99: De-blurred images. (a) Proposed method (b) Goldstein *et al*. (c) Fergus *et al*. (d) Whyte *et al*. (e) Shan *et al*. (f) Xu *et al*. (g) Cho *et al*.

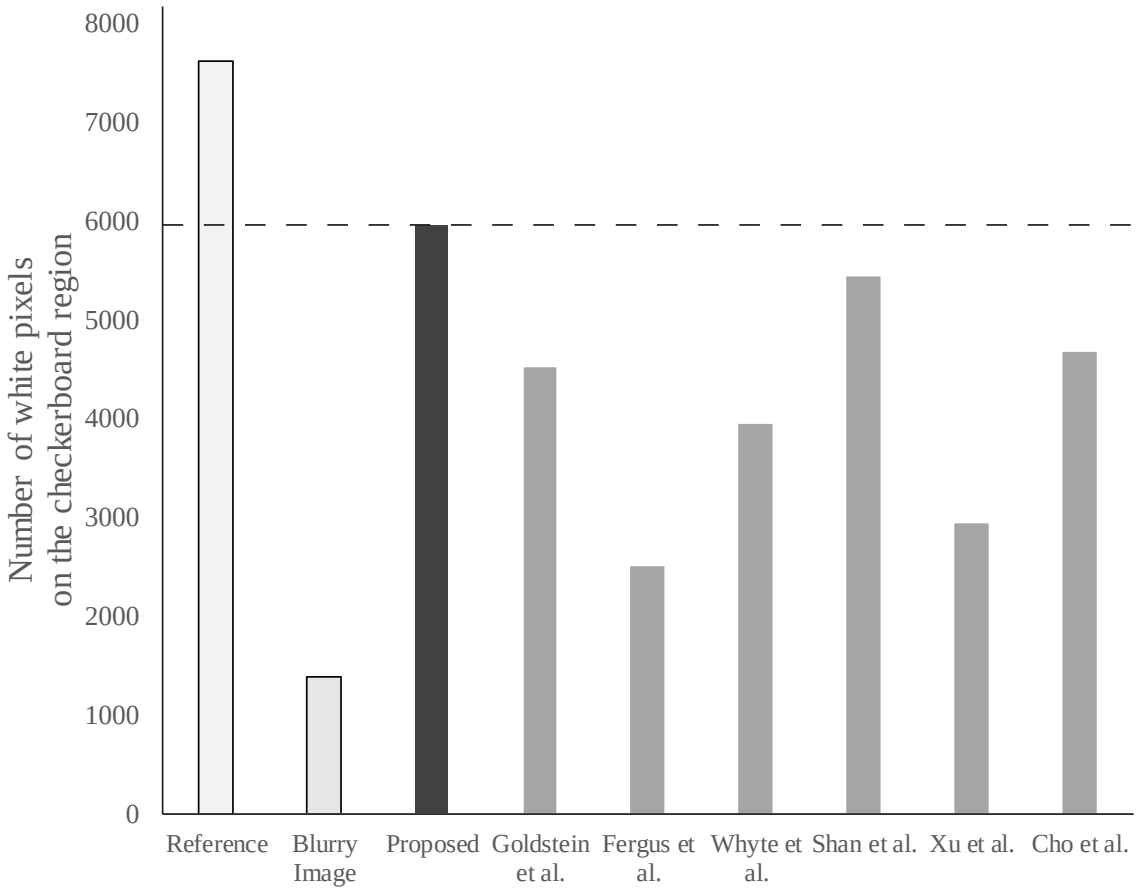


Figure 100: Number of white pixels on the checkerboard region.

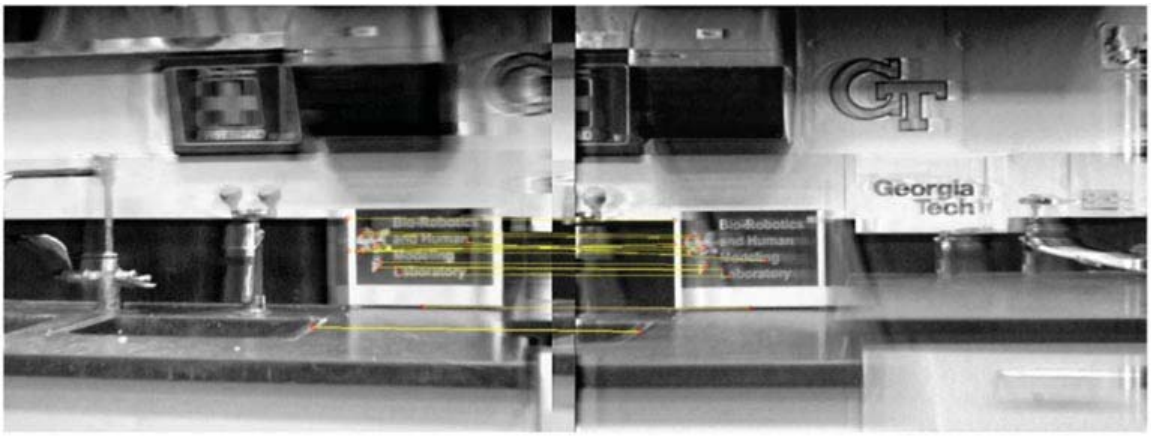
APPENDIX C

IMPROVED IMAGE PROCESSING BY DE-BLURRING

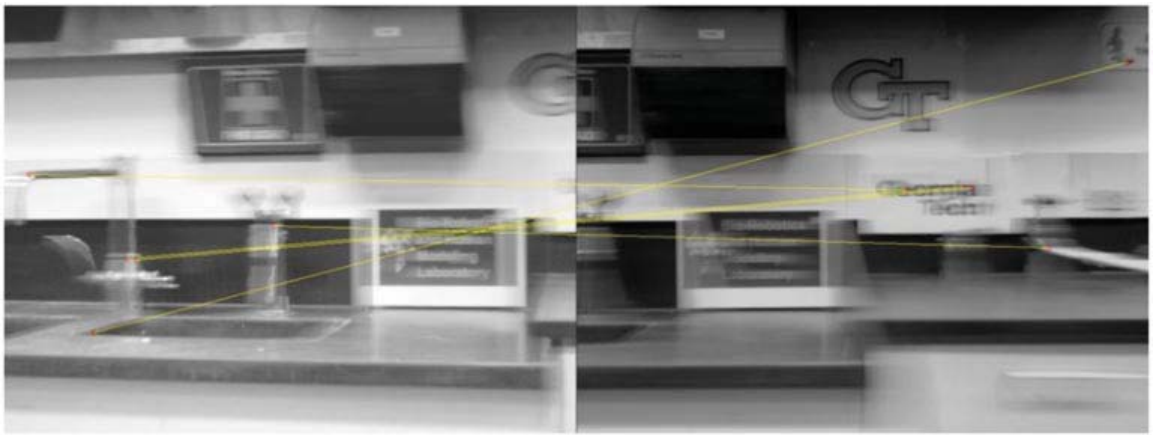
Most of the existing image processing algorithms used clean images because it is challenging to accurately extract feature points or edges from degraded blurry images. The dynamics-based image de-blurring method was incorporated to improve the quality of the panoramic image in Chapter 5. In fact, the de-blurring method can benefit other image processing algorithms by recovering high-frequency information.

Figures 101 and 102 show comparisons of the SIFT and RANSAC results between the de-blurred and blurry images. The degree of blurriness of Figure 101 is more severe than that of Figure 102. It can be observed from Figure 101 that the de-blurred image set has an improved result, while most of the matched features in the blurry image set are inaccurate. Although both sets have an accurate feature matching in Figure 102, it can be observed that the de-blurred image set has an improved result with increased number of feature matching.

The effectiveness of the de-blurring method for image processing can also be checked by other means. Figures 103, 104, and 105 show panoramic images generated by the compared methods discussed in Chapter 5 when the de-blurred images were given. Although minor artifacts exist, it can be observed that all programs successfully generated the images without feeding additional clean images.

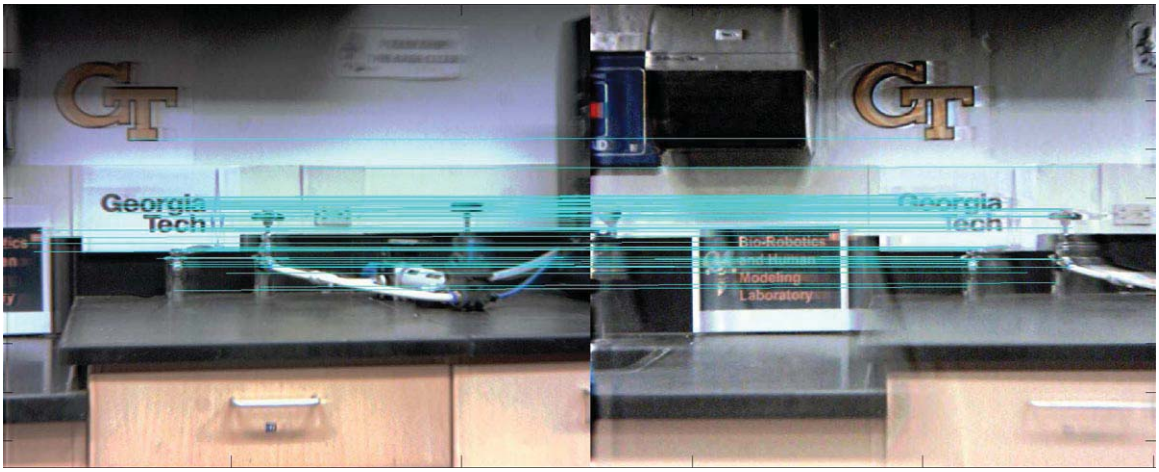


(a)

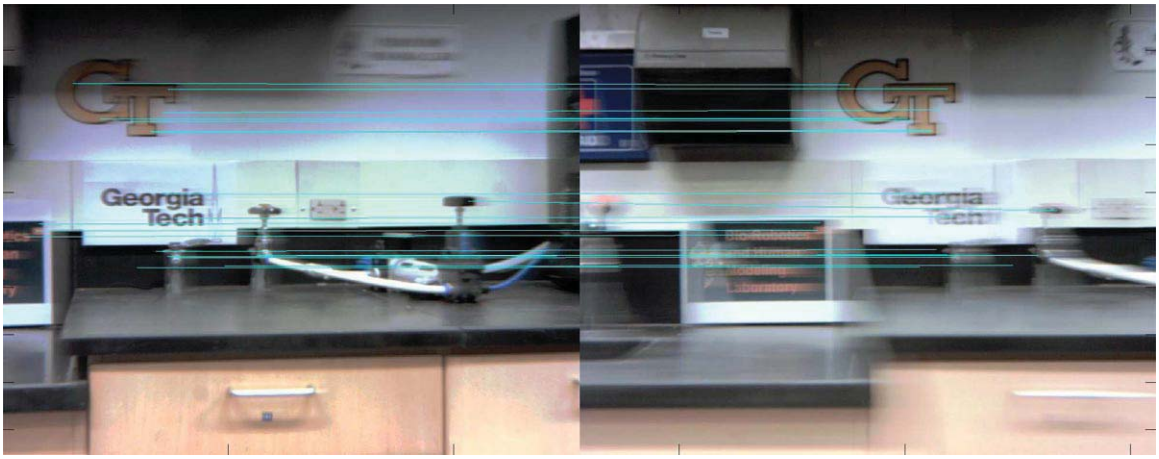


(b)

Figure 101: Examples of image stitching by SIFT and RANSAC. (a) Matched feature points from a de-blurred image set. (b) Matched feature points from a blurry image set.



(a)



(b)

Figure 102: Examples of image stitching by SIFT and RANSAC. (a) Matched feature points from a de-blurred image set. (b) Matched feature points from a blurry image set.

Figure 103: Panoramic image generated by MS ICE given the de-blurred images.

Figure 104: Panoramic image generated by PTGui given the de-blurred images.

Figure 105: Panoramic image generated by Panoweaver given the de-blurred images.

APPENDIX D

PANORAMIC IMAGE GENERATION

In Chapter 5, the experiments for panorama generation were conducted in the laboratory. An additional experiment was conducted on the first floor atrium of the Love building (Georgia Tech), as shown in Figure 106.

Figures 107 and 108 show panoramic results of the proposed and compared method, respectively. Since no output images were generated for the compared methods using the same set of images, two additional images with low degree of motion blur were provided to avoid failure.

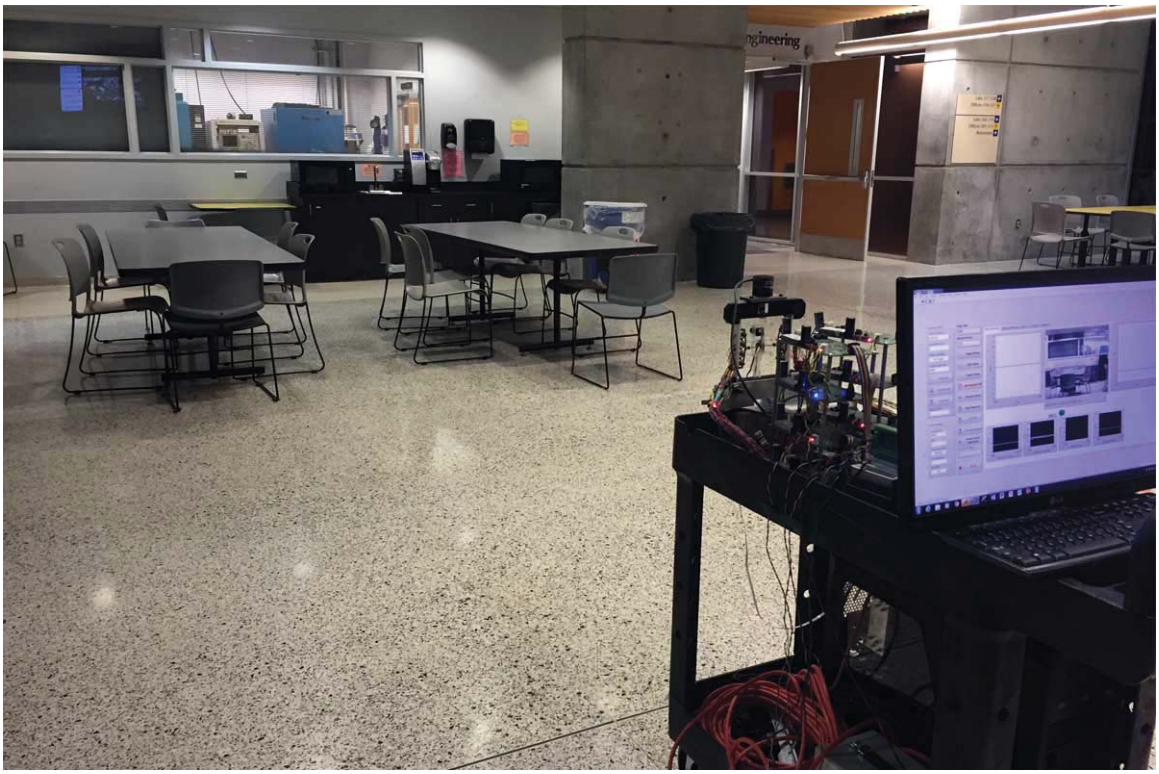


Figure 106: Experimental setup for panorama generation.

It can be seen that the proposed method generates stitching results with no geometric distortion or misalignment as shown in Figure 107(a). Since no additional image processing was applied such as blending, the result exhibits minor seams.

The panoramic image generated by MS ICE shows successive results. However, the entire image is curved or distorted. The panoramic image generated by PT-Gui exhibits substantial geometrical misalignments and the entire image is distored. Panoweaver software was not effective in this experiment as shown in Figure 108(c).

(a)

(b)

Figure 107: Panoramic results of the proposed method. (a) Dynamics-based method incorporating with real-time image de-blurring. (b) Clean static image for reference.

(a)

(b)

(c)

Figure 108: Panoramic results of the compared method. (a) MS ICE (b) PTGui (c) Panoweaver

APPENDIX E

ROBOTIC VISION SYSTEM FIRMWARE

Figure 109 shows a GUI controlling the robotic vision system. The GUI allows users to change control parameters, camera parameters, and operation modes which are programmed in the FPGAs. Various control methods discussed in this study can be selected and the user can enter the desired location either manually or automatically. The GUI displays live image streams and the position of the camera orientation system. Various parameters programmed in both FPGAs and RT-Level can be monitored and saved.

The code for the robotic vision system are implemented in FPGAs, RT Level, CPU, and GPU. Most of the core functions are programmed in the FPGAs to take advantage of fast sampling rates and parallelism. Figure 110 shows the state diagram representation of functions programmed in the FPGAs. In order to control distributed actuators and determine key parameters for image processing, related processes are executed in a parallel fashion. The sampling rates of each process are tabulated in Table 19.

The host PC is mainly used for parameter setting, system monitoring, and image acquisition, as shown in Figure 111. The user interface, receiving parameters for image processing from FPGAs, and image processing are operated in a parallel fashion. This enables the dynamics-based processing methods to be performed in a pipelining fashion, as shown in Figure 112.

The RT-Level is used to generate commands for smooth pursuit or velocity control, and write data into a file.

Figure 113 shows a GUI controlling two linear stages that are used in the smooth-pursuit experiments. The GUI allows users to change motion parameters of both linear stages independently, while control parameters are not allowed to be adjusted via GUI. If the velocity of the desired motion is not constant, then the motion profile needs to be pre-programmed. For operating the linear stages, the control functions are programmed in both RTOS and FPGAs. Once the desired position is determined in the RT-Level, the control signals are generated in the FPGAs as shown in Figure 114. The desired position can be interrupted or updated at every control sample.

Table 19: Sampling rates of the key functions in FPGAs [kHz]

Receive Desire Position	100
Switching Process	400
Position Estimation	100
Sync & Trigger Camera	1
Monitor Exposure Window	100
Read Encoder	1
Save Data	2

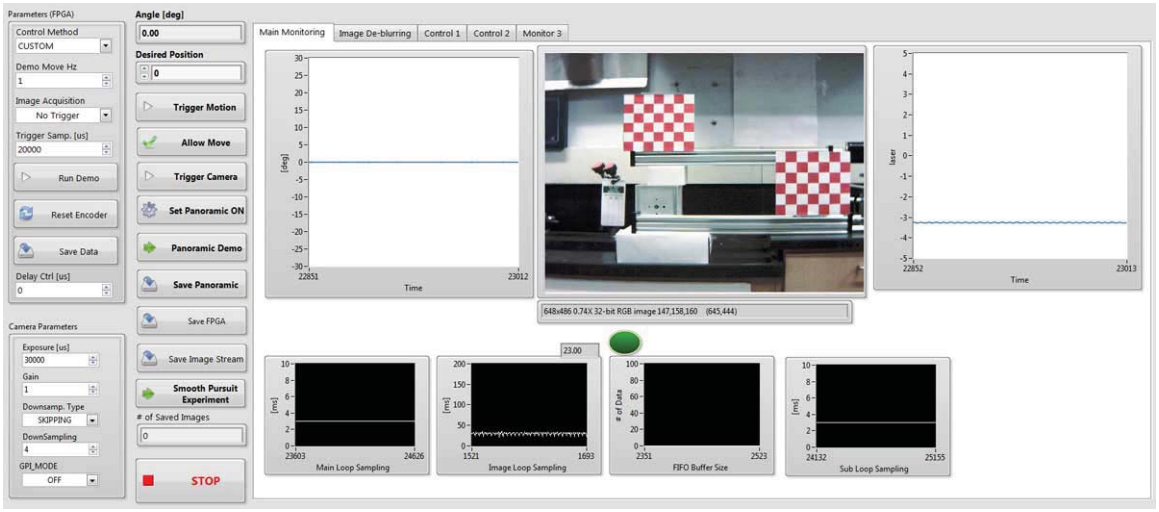


Figure 109: GUI for the robotic vision system

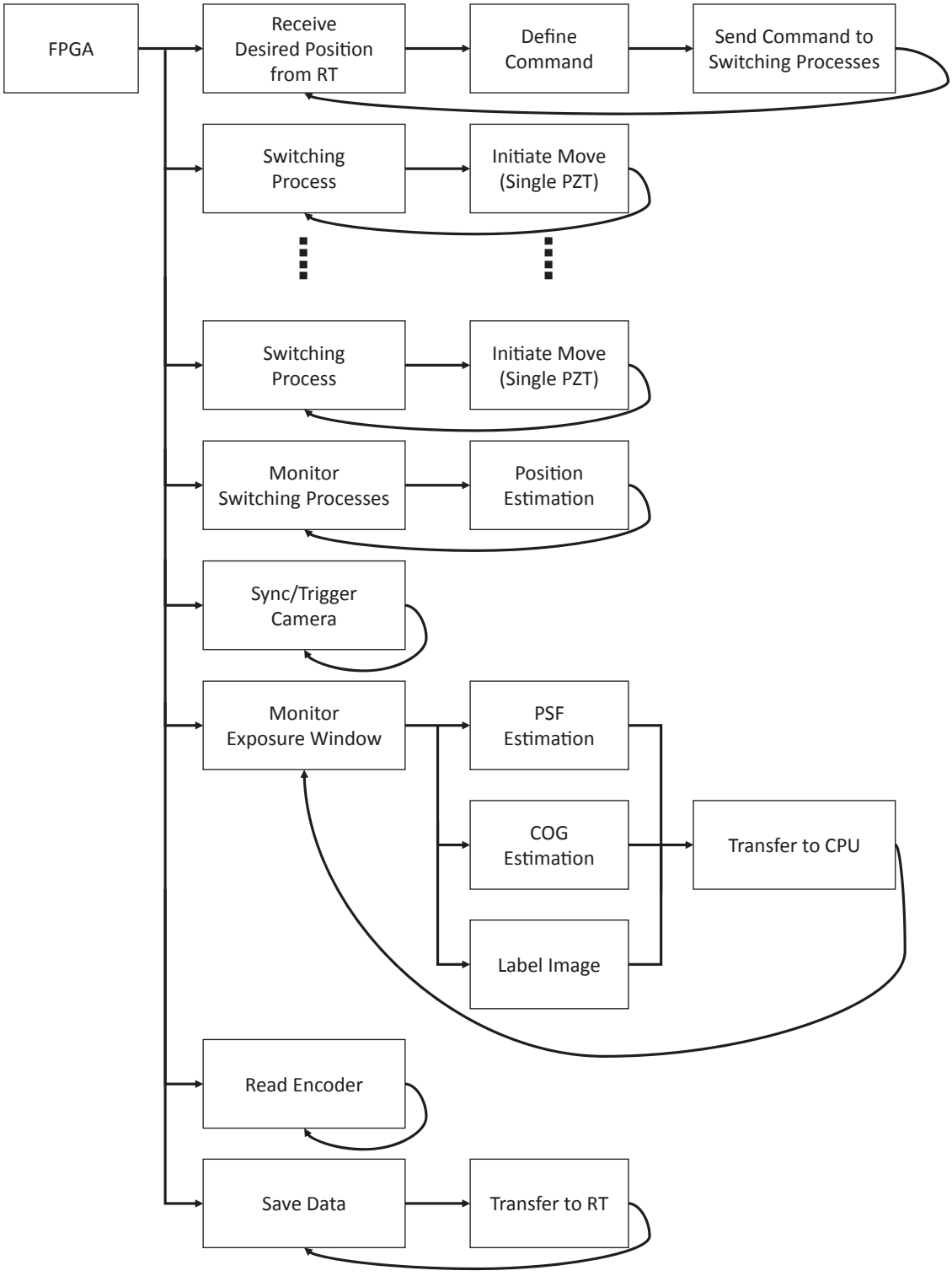


Figure 110: State diagram representation of functions in the FPGAs.

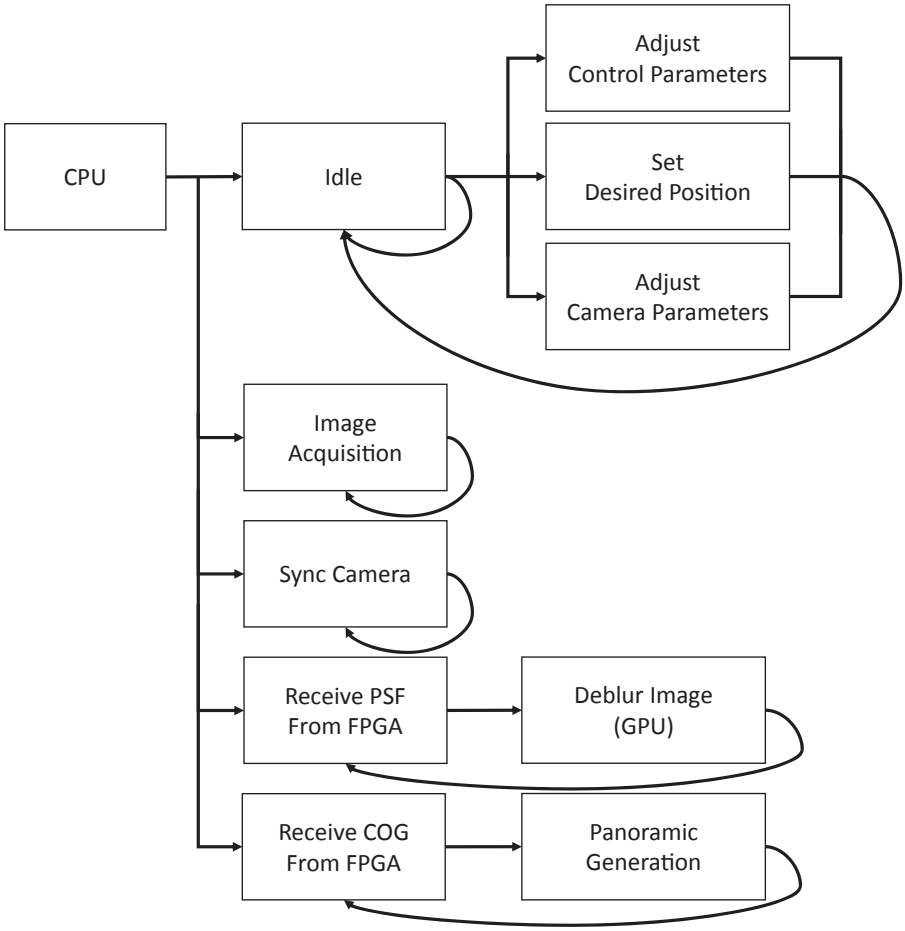


Figure 111: State diagram representation of functions in the CPU.

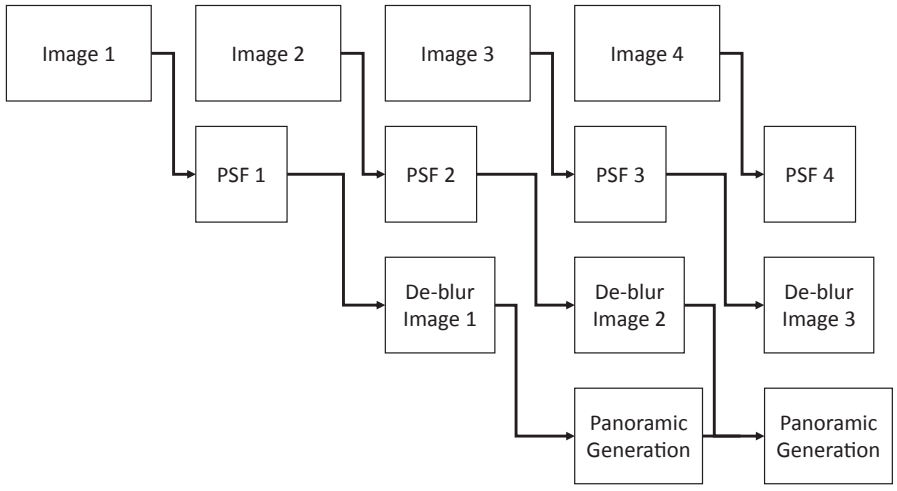


Figure 112: The dynamics-based image processing methods are executed in a pipelining fashion.

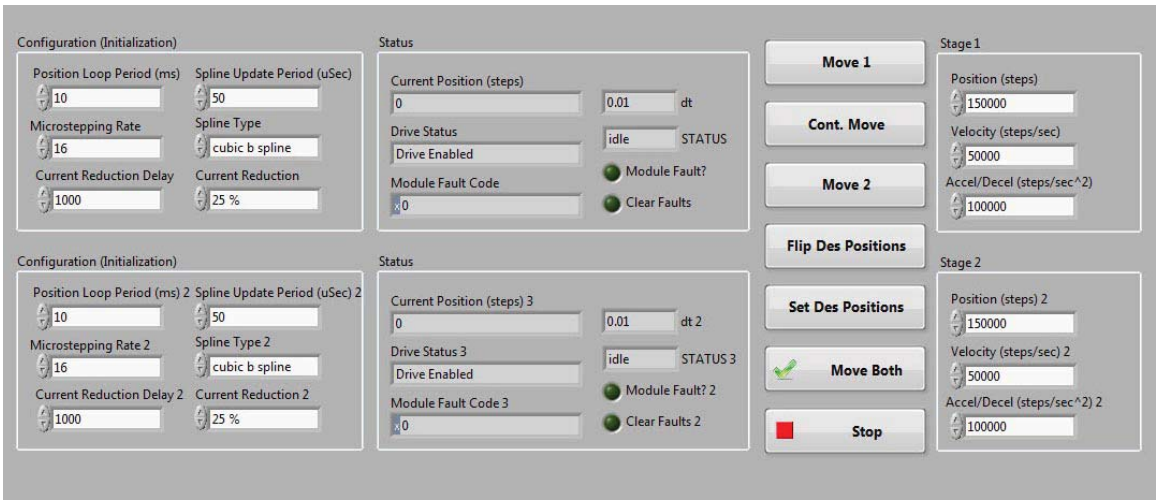


Figure 113: GUI for operating two linear stages

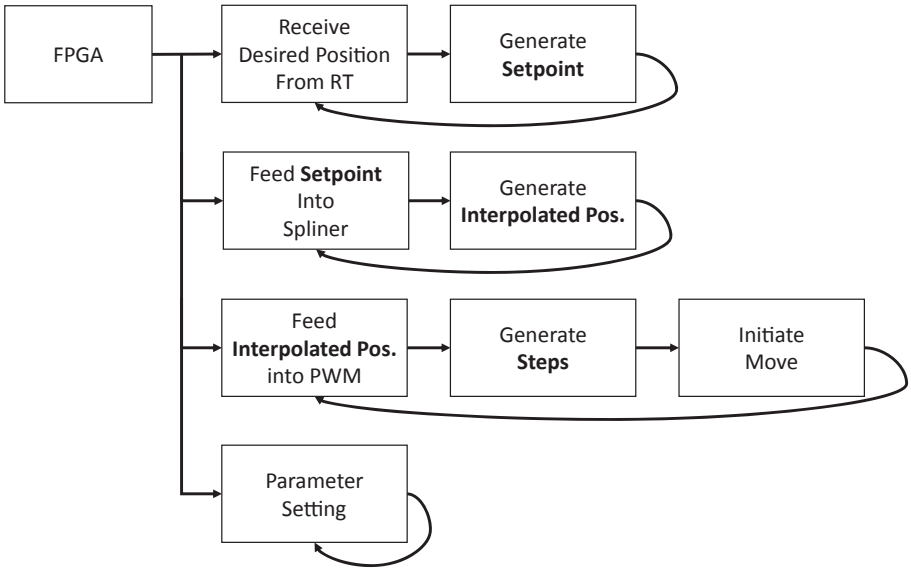


Figure 114: State diagram representation of functions in the FPGAs for operating linear stages.

APPENDIX F

DESIGN OF A 2-DOF CAMERA POSITIONER

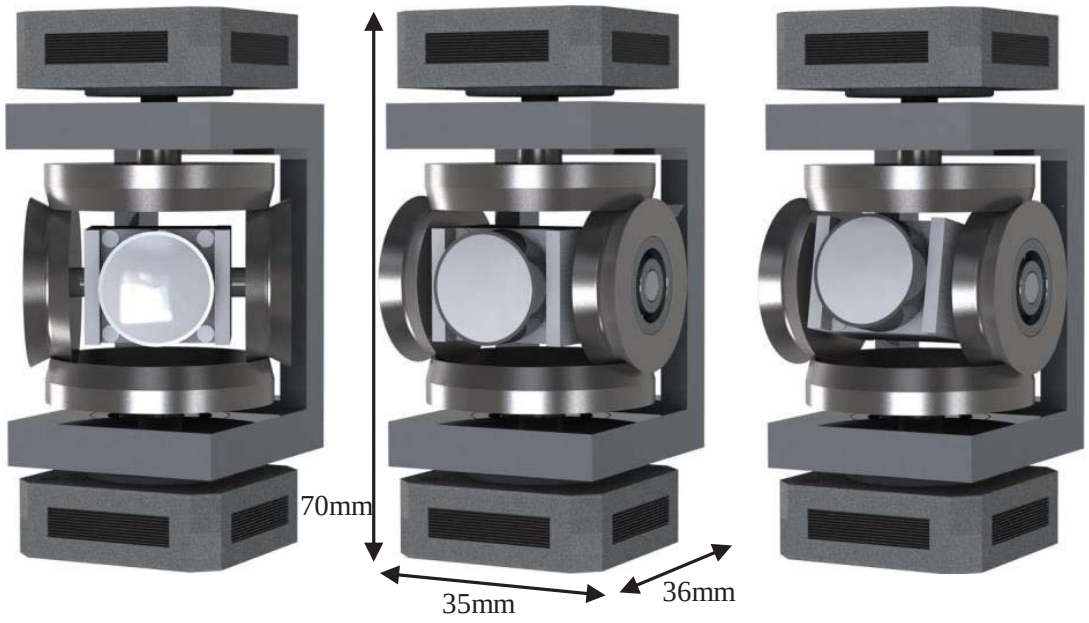


Figure 115: CAD model of a 2-DOF mechanism

The robotic vision system used in this study has a single-DOF. Although the developed dynamics-based approaches are applicable to 2D motion, the impact of verification experiments may be limited. A 2-DOF platform was designed as shown in Figure 115. Panning and tiling is realized by an orthogonal roller engagement mechanism. Friction-drive rollers are used instead of bevel gears. In replacement of the antagonistic cellular PZT actuators, a pair of piezoelectric rotary stepper motors is used, which is also suitable for an open-loop control and quantization.

Figure 116: Exploded view of the CAD model

REFERENCES

[1] Ahissar, E. and Arieli, A., "Seeing via miniature eye movements: A dynamic hypothesis for vision," *Frontiers in Computational Neuroscience*, vol. 6, no. 89, pp. 1–27, 2012.

[2] Asano, S., Maruyama, T., and Yamaguchi, Y., "Performance comparison of fpga, gpu and cpu in image processing," in *Field Programmable Logic and Applications, 2009. FPL 2009. International Conference on*, pp. 126–131, IEEE, 2009.

[3] Bae, H., Fowlkes, C., and Chou, P., "Accurate motion deblurring using camera motion tracking and scene depth," in *Applications of Computer Vision (WACV), 2013 IEEE Workshop on*, pp. 148–153, Jan 2013.

[4] Bar, L., Berkels, B., Rumpf, M., and Sapiro, G., "A variational framework for simultaneous motion estimation and restoration of motion-blurred video," in *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pp. 1–8, 2007.

[5] Barmack, N., "Modification of eye movements by instantaneous changes in the velocity of visual targets," *Vision research*, vol. 10, no. 12, pp. 1431–1441, 1970.

[6] Bay, H., Tuytelaars, T., and Van Gool, L., "Surf: Speeded up robust features," in *Computer Vision–ECCV 2006*, pp. 404–417, Springer, 2006.

[7] Becker, W. and Jurgens, R., "An analysis of the saccadic system by means of double step stimuli," *Vision Research*, vol. 19, no. 9, pp. 967 – 983, 1979.

[8] Bedell, H. E. and Lott, L. A., "Suppression of motion-produced smear during smooth pursuit eye movements," *Current Biology*, vol. 6, no. 8, pp. 1032–1034, 1996.

[9] Ben-Ezra, M. and Nayar, S., "Motion-based motion deblurring," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 26, no. 6, pp. 689–698, 2004.

[10] Bishop, C. M. and others, *Pattern recognition and machine learning*, vol. 1. springer New York, 2006.

[11] Burr, D. C., "Motion smear," *Nature*, vol. 284, no. 13, pp. 164–165, 1980.

[12] Cannata, G. and Maggiali, M., "Models for the design of bioinspired robot eyes," *IEEE Transactions on Robotics*, vol. 24, pp. 27–44, Feb 2008.

[13] Carpenter, R. H., *Movements of the eyes (2nd rev)*. Pion Limited, 1988.

[14] CEDRAT, "Piezo products catalogue version 3.2, cedrat inc. (2007)."

[15] Chapra, S. C. and Canale, R. P., *Numerical methods for engineers*, vol. 2. McGraw-Hill New York, 2012.

[16] Chen, J., Liu, Y., Gu, Q., Aoyama, T., Takaki, T., and Ishii, I., "Robot-mounted 500-fps 3-d shape measurement using motion-compensated coded structured light method," in *Robotics and Biomimetics (ROBIO), 2014 IEEE International Conference on*, pp. 1989–1994, Dec 2014.

[17] Chen, S., Bedell, H. E., and Ömen, H., "A target in real motion appears blurred in the absence of other proximal moving targets," *Vision research*, vol. 35, no. 16, pp. 2315–2328, 1995.

[18] Cho, S. and Lee, S., "Fast motion deblurring," *ACM Transactions on Graphics (TOG)*, vol. 28, no. 5, pp. 145:1–8, 2009.

[19] Choi, C. and Christensen, H., "Rgb-d object tracking: A particle filter approach on gpu," in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pp. 1084–1091, Nov 2013.

[20] Cianchetti, M., Ranzani, T., Gerboni, G., Falco, I. D., Laschi, C., and Menciassi, A., "Stiff-flop surgical manipulator: Mechanical design and experimental characterization of the single module," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3576–3581, Nov 2013.

[21] Comaniciu, D., Ramesh, V., and Meer, P., "Kernel-based object tracking," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 25, no. 5, pp. 564–577, 2003.

[22] Coughlin, D. J. and Rome, L. C., "The roles of pink and red muscle in powering steady swimming in scup, stenotomus chrysops," *American zoologist*, vol. 36, no. 6, pp. 666–677, 1996.

[23] DiCarlo, J. J., Zoccolan, D., and Rust, N. C., "How does the brain solve visual object recognition?," *Neuron*, vol. 73, no. 3, pp. 415–434, 2012.

[24] Dodge, R., "Five types of eye movement in the horizontal meridian plane of the field of regard," *American Journal of Physiology–Legacy Content*, vol. 8, no. 4, pp. 307–329, 1903.

[25] Easypano Holdings Inc., "Panoweaver," 2015.

[26] Enderle, J. D. and Wolfe, J. W., "Time-optimal control of saccadic eye movements," *Biomedical Engineering, IEEE Transactions on*, no. 1, pp. 43–55, 1987.

[27] Fergus, R., Singh, B., Hertzmann, A., Roweis, S. T., and Freeman, W. T., "Removing camera shake from a single photograph," *ACM Transactions on Graphics (TOG)*, vol. 25, no. 3, pp. 787–794, 2006.

[28] Fischler, M. A. and Bolles, R. C., "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[29] Fish, D., Brinicombe, A., Pike, E., and Walker, J., "Blind deconvolution by means of the richardson–lucy algorithm," *JOSA A*, vol. 12, no. 1, pp. 58–65, 1995.

[30] Fixot, R. S. *Americal Journal of Opthamology*, 1957.

[31] Garcia, V., Debreuve, E., Nielsen, F., and Barlaud, M., "K-nearest neighbor search: Fast gpu-based implementations and application to high-dimensional feature matching," in *2010 IEEE International Conference on Image Processing*, pp. 3757–3760, Sept 2010.

[32] GigaPan, "Epic pro user guide (2016)."

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

[34] Götz, M., Dittmann, F., and Pereira, C. E., "Deterministic mechanism for run-time reconfiguration activities in an rtos," in *Industrial Informatics, 2006 IEEE International Conference on*, pp. 693–698, IEEE, 2006.

[35] Grundmann, M., Kwatra, V., Castro, D., and Essa, I., "Calibration-free rolling shutter removal," in *Computational Photography (ICCP), 2012 IEEE International Conference on*, pp. 1–8, April 2012.

[36] Gu, J., Hitomi, Y., Mitsunaga, T., and Nayar, S., "Coded rolling shutter photography: Flexible space-time sampling," in *Computational Photography (ICCP), 2010 IEEE International Conference on*, pp. 1–8, March 2010.

[37] Ham, R. V., Sugar, T. G., Vanderborght, B., Hollander, K. W., and Lefeber, D., "Compliant actuator designs," *IEEE Robotics Automation Magazine*, vol. 16, pp. 81–94, September 2009.

[38] Hamming, R., *Numerical methods for scientists and engineers*. Courier Corporation, 2012.

[39] Harris, C. and Stephens, M., "A combined corner and edge detector.," in *Alvey vision conference*, vol. 15, p. 50, Citeseer, 1988.

[40] Harris, C. M. and Wolpert, D. M., "Signal-dependent noise determines motor planning," *Nature*, vol. 394, no. 6695, pp. 780–784, 1998.

[41] Henley, S. L., "Seamless multi-camera panoramic imaging with distortion correction and selectable field of view," Aug. 12 1997. US Patent 5,657,073.

[42] Holcombe, A. O., "Seeing slow and seeing fast: two limits on perception," *Trends in cognitive sciences*, vol. 13, no. 5, pp. 216–221, 2009.

[43] Hyde, J. M. and Seering, W. P., "Using input command pre-shaping to suppress multiple mode vibration," in *Robotics and Automation, 1991. Proceedings., 1991 IEEE International Conference on*, pp. 2604–2609 vol.3, Apr 1991.

[44] Jia, J., "Single image motion deblurring using transparency," in *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pp. 1–8, IEEE, 2007.

[45] Joshi, N., Kang, S. B., Zitnick, C. L., and Szeliski, R., "Image deblurring using inertial measurement sensors," *ACM Transactions on Graphics (TOG)*, vol. 29, no. 4, p. 30, 2010.

[46] Judge, S. J., Wurtz, R. H., and Richmond, B. J., "Vision during saccadic eye movements. i. visual interactions in striate cortex," *J Neurophysiol*, vol. 43, no. 4, pp. 1133–55, 1980.

[47] Jurgen, W. and Segal, S. S., "Effect of motor unit recruitment on functional vasodilatation in hamster retractor muscle," *The Journal of physiology*, vol. 524, no. 1, pp. 267–278, 2000.

[48] Jürgens, R., Becker, W., and Kornhuber, H., "Natural and drug-induced variations of velocity and duration of human saccadic eye movements: evidence for a control of the neural pulse generator by local feedback," *Biological cybernetics*, vol. 39, no. 2, pp. 87–96, 1981.

[49] Katoch, R. and Ueda, J., "Trajectory planning for antagonistic non-linearly quantized sma actuator arrays based on evolutionary optimization," in *2015 American Control Conference (ACC)*, pp. 2631–2636, July 2015.

[50] Kilbride, S., Kim, M., and Ueda, J., "Real time image de-blurring and image stitching for muscle inspired camera orientation system," in *Advanced Robotics and its Social Impacts (ARSO2014), 2014 IEEE International Workshop on*, to be publihsed 2014.

[51] Kim, M. D. and Ueda, J., "Dynamics-based motion deblurring for a biologically-inspired camera positioning mechanism," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2689–2694, Nov 2013.

[52] Kim, M. D. and Ueda, J., "Real-time image de-blurring and image processing for a robotic vision system," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1899–1904, May 2015.

[53] Kim, M. D. and Ueda, J., "Discrete switching commands for tracking and vibration suppression using a quantized, compliant camera orientation system," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5177–5182, May 2016.

[54] Kim, M. D. and Ueda, J., "Real-time panoramic image generation and motion deblurring by using dynamics-based robotic vision," *IEEE/ASME Transactions on Mechatronics*, vol. 21, pp. 1376–1387, June 2016.

[55] Kim, M. D. and Ueda, J., "Dynamics-based motion de-blurring for a pzt-driven, compliant camera orientation mechanism," *The International Journal of Robotics Research*, vol. 34, no. 4-5, pp. 653–673, 2015.

[56] Koo, I. M., Jung, K., Koo, J. C., Nam, J. D., Lee, Y. K., and Choi, H. R., "Development of soft-actuator-based wearable tactile display," *IEEE Transactions on Robotics*, vol. 24, pp. 549–558, June 2008.

[57] Laffranchi, M., Tsagarakis, N. G., and Caldwell, D. G., "Analysis and development of a semiactive damper for compliant actuation systems," *IEEE/ASME Transactions on Mechatronics*, vol. 18, pp. 744–753, April 2013.

[58] Lee, Y.-C., Lan, C.-C., Chu, C.-Y., Lai, C.-M., and Chen, Y.-J., "A pan-tilt orienting mechanism with parallel axes of flexural actuation," *Mechatronics, IEEE/ASME Transactions on*, vol. 18, no. 3, pp. 1100–1112, 2013.

[59] Lehrer, S. S., "The regulatory switch of the muscle thin filament: Ca 2+ or myosin heads?," *Journal of muscle research and cell motility*, vol. 15, no. 3, pp. 232–236, 1994.

[60] Lenz, A., Anderson, S., Pipe, A., Melhuish, C., Dean, P., and Porrill, J., "Cerebellar-inspired adaptive control of a robot eye actuated by pneumatic artificial muscles," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 39, no. 6, pp. 1420–1433, 2009.

[61] Lesmana, M. and Pai, D., "A biologically inspired controller for fast eye movements," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pp. 3670–3675, 2011.

[62] Levin, A., Weiss, Y., Durand, F., and Freeman, W. T., "Understanding and evaluating blind deconvolution algorithms," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pp. 1964–1971, IEEE, 2009.

[63] Liu, X. and El Gamal, A., "Simultaneous image formation and motion blur restoration via multiple capture," in *Acoustics, Speech, and Signal Processing, 2001. Proceedings.(ICASSP'01). 2001 IEEE International Conference on*, vol. 3, pp. 1841–1844, IEEE, 2001.

[64] López, M. B., Hannuksela, J., Silvén, O., and Vehviläinen, M., "Graphics hardware accelerated panorama builder for mobile phones," in *IS&T/SPIE Electronic Imaging*, pp. 72560D–72560D, International Society for Optics and Photonics, 2009.

[65] Lourenço, M., Barreto, J. P., and Vasconcelos, F., "srd-sift: Keypoint detection and matching in images with radial distortion," *Robotics, IEEE Transactions on*, vol. 28, no. 3, pp. 752–760, 2012.

[66] Lowe, D. G., "Object recognition from local scale-invariant features," in *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, vol. 2, pp. 1150–1157, Ieee, 1999.

[67] Luebke, D., "Cuda: Scalable parallel programming for high-performance scientific computing," in *Biomedical Imaging: From Nano to Macro, 2008. ISBI 2008. 5th IEEE International Symposium on*, pp. 836–838, May 2008.

[68] Mark, A. G., Palagi, S., Qiu, T., and Fischer, P., "Auxetic metamaterial simplifies soft robot design," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4951–4956, May 2016.

[69] Martini, F. H., Bartholomew, E. F., and Ober, W. C., *Essentials of anatomy & physiology*. Pearson, 2013.

[70] Matin, E., "Saccadic suppression: a review and an analysis.," *Psychological bulletin*, vol. 81, no. 12, p. 899, 1974.

[71] Mayer-Lindenberg, F., "High-level fpga programming through mapping process networks to fpga resources," in *Reconfigurable Computing and FPGAs, 2009. ReConFig '09. International Conference on*, pp. 302–307, Dec 2009.

[72] Mei, C. and Rives, P., "Single view point omnidirectional camera calibration from planar grids," in *Proceedings 2007 IEEE International Conference on Robotics and Automation*, pp. 3945–3950, April 2007.

[73] Microsoft, "Image composite editor," 2015.

[74] Miskin, J. and MacKay, D. J., "Ensemble learning for blind image separation and deconvolution," in *Advances in independent component analysis*, pp. 123–141, Springer, 2000.

[75] Morrow, J., Shin, H. S., Phillips-Grafflin, C., Jang, S. H., Torrey, J., Larkins, R., Dang, S., Park, Y. L., and Berenson, D., "Improving soft pneumatic actuator fingers through integration of soft sensors, position and force control, and rigid fingernails," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5024–5031, May 2016.

[76] NationalInstruments, "Compactrio developers guide," *http://www.ni.com/compactriodevguide/*, 2009.

[77] New House Internet Services BV, "PTGui," 2015.

[78] Nickolls, J., Buck, I., Garland, M., and Skadron, K., "Scalable parallel programming with cuda," *Queue*, vol. 6, no. 2, pp. 40–53, 2008.

[79] Okumura, K.-I., Raut, S., Gu, Q., Aoyama, T., Takaki, T., and Ishii, I., "Real-time feature-based video mosaicing at 500 fps," in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pp. 2665–2670, Nov 2013.

[80] Ozel, S., Skorina, E. H., Luo, M., Tao, W., Chen, F., Pan, Y., and Onal, C. D., "A composite soft bending actuation module with integrated curvature sensing," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4963–4968, May 2016.

[81] Papoulis, A., *Signal analysis*, vol. 191. McGraw-Hill, 1977.

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

[83] Pauwels, K., Tomasi, M., Diaz Alonso, J., Ros, E., and Van Hulle, M. M., "A comparison of fpga and gpu for real-time phase-based optical flow, stereo, and local image features," *Computers, IEEE Transactions on*, vol. 61, no. 7, pp. 999–1012, 2012.

[84] Peleg, S. and Herman, J., "Panoramic mosaics by manifold projection," in *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*, pp. 338–343, IEEE, 1997.

[85] Pfeil, J., "Camera system for recording images, and associated method," Apr. 30 2012. US Patent App. 14/113,924.

[86] Pitzer, B., Osentoski, S., Jay, G., Crick, C., and Jenkins, O., "Pr2 remote lab: An environment for remote development and experimentation," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pp. 3200–3205, May 2012.

[87] Raskar, R., Agrawal, A., and Tumblin, J., "Coded exposure photography: motion deblurring using fluttered shutter," *ACM Transactions on Graphics (TOG)*, vol. 25, no. 3, pp. 795–804, 2006.

[88] Rav-Acha, A. and Peleg, S., "Two motion-blurred images are better than one," *Pattern recognition letters*, vol. 26, no. 3, pp. 311–317, 2005.

[89] Richardson, W. H., "Bayesian-based iterative method of image restoration," *JOSA*, vol. 62, no. 1, pp. 55–59, 1972.

[90] Richter, H., Misawa, E. A., and O'Dell, B. D., "Stability analysis of discrete linear systems with quantized input," in *Proceedings of the 2002 American Control Conference (IEEE Cat. No.CH37301)*, vol. 4, pp. 2991–2996 vol.4, May 2002.

[91] Robinson, D. A., "The mechanics of human saccadic eye movement," *The Journal of Physiology*, vol. 174, no. 2, pp. 245–264, 1964.

[92] Rolfs, M., Jonikaitis, D., Deubel, H., and Cavanagh, P., "Predictive remapping of attention across eye movements," *Nature neuroscience*, vol. 14, no. 2, pp. 252–256, 2010.

[93] Rousselet, G. A., Fabre-Thorpe, M., and Thorpe, S. J., "Parallel processing in high-level categorization of natural images," *Nature neuroscience*, vol. 5, no. 7, pp. 629–630, 2002.

[94] Russo, L. M., Pedrino, E. C., Kato, E., and Roda, V. O., "Image convolution processing: A gpu versus fpga comparison," in *Programmable Logic (SPL), 2012 VIII Southern Conference on*, pp. 1–6, IEEE, 2012.

[95] Schultz, J. and Ueda, J., "A camera positioner driven by muscle-like actuation," in *Biomedical Robotics and Biomechatronics (BioRob), 2012 4th IEEE RAS EMBS International Conference on*, pp. 719–724, 2012.

[96] Schultz, J. and Ueda, J., "Experimental verification of discrete switching vibration suppression," *Mechatronics, IEEE/ASME Transactions on*, vol. 17, no. 2, pp. 298–308, 2012.

[97] Schultz, J. and Ueda, J., "Nested piezoelectric cellular actuators for a biologically inspired camera positioning mechanism," *Robotics, IEEE Transactions on*, vol. 29, pp. 1125–1138, Oct 2013.

[98] Schultz, J. and Ueda, J., "Two-port network models for compliant rhomboidal strain amplifiers," *Robotics, IEEE Transactions on*, vol. 29, pp. 42–54, Feb 2013.

[99] Schultz, J. A. and Ueda, J., "Intersample discretization of control inputs for flexible systems with quantized cellular actuation," in *ASME 2010 Dynamic Systems and Control Conference*, pp. 421–428, American Society of Mechanical Engineers, 2010.

[100] Schulz, S., Ltkebohle, I., and Wachsmuth, S., "An affordable, 3d-printable camera eye with two active degrees of freedom for an anthropomorphic robot," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pp. 764–771, Oct 2012.

[101] Schuon, S. and Diepold, K., "Comparison of motion de-blur algorithms and real world deployment," *Acta Astronautica*, vol. 64, no. 11, pp. 1050–1065, 2009.

[102] Seok, S., Onal, C. D., Cho, K. J., Wood, R. J., Rus, D., and Kim, S., "Meshworm: A peristaltic soft robot with antagonistic nickel titanium coil actuators," *IEEE/ASME Transactions on Mechatronics*, vol. 18, pp. 1485–1497, Oct 2013.

[103] Shakernia, O., Ma, Y., Koo, T. J., and Sastry, S., "Landing an unmanned air vehicle: Vision based motion estimation and nonlinear control," *Asian journal of control*, vol. 1, no. 3, pp. 128–145, 1999.

[104] Shan, Q., Xiong, W., and Jia, J., "Rotational motion deblurring of a rigid object from a single image," in *2007 IEEE 11th International Conference on Computer Vision*, pp. 1–8, Oct 2007.

[105] Shan, Q., Jia, J., and Agarwala, A., "High-quality motion deblurring from a single image," *ACM Transactions on Graphics (TOG)*, vol. 27, no. 3, p. 73, 2008.

[106] Shi, M. and Zheng, J. Y., "A slit scanning depth of route panorama from stationary blur," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1, pp. 1047–1054 vol. 1, 2005.

[107] Singer, N. C. and Seering, W. P., "Preshaping command inputs to reduce system vibration," *Journal of Dynamic Systems, Measurement, and Control*, vol. 112, no. 1, pp. 76–82, 1990.

[108] Skorina, E. H., Luo, M., Ozel, S., Chen, F., Tao, W., and Onal, C. D., "Feedforward augmented sliding mode motion control of antagonistic soft pneumatic actuators," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2544–2549, May 2015.

[109] Song, Y. and Zhang, X., "An active binocular integrated system for intelligent robot vision," in *Intelligence and Security Informatics (ISI), 2012 IEEE International Conference on*, pp. 48–53, 2012.

[110] Spica, R., Giordano, P., and Chaumette, F., "Coupling visual servoing with active structure from motion," in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pp. 3090–3095, May 2014.

[111] Szeliski, R., "Image alignment and stitching: A tutorial," *Found. Trends. Comput. Graph. Vis.*, vol. 2, pp. 1–104, Jan. 2006.

[112] Takahashi, I. and Noguchi, T., "A new quick-response and high-efficiency control strategy of an induction motor," *Industry Applications, IEEE Transactions on*, no. 5, pp. 820–827, 1986.

[113] Thorpe, S., Fize, D., Marlot, C., and others, "Speed of processing in the human visual system," *nature*, vol. 381, no. 6582, pp. 520–522, 1996.

[114] Tong, J., Stevenson, S. B., and Bedell, H. E., "Signals of eye-muscle proprioception modulate perceived motion smear," *Journal of Vision*, vol. 8, no. 14, pp. 1–6, 2008.

[115] Tsagarakis, N. G., Laffranchi, M., Vanderborght, B., and Caldwell, D. G., "A compact soft actuator unit for small scale human friendly robots," in *Robotics and Automation, 2009. ICRA '09. IEEE International Conference on*, pp. 4356–4362, May 2009.

[116] Ueda, J., Secord, T., and Asada, H., "Large effective-strain piezoelectric actuators using nested cellular architecture with exponential strain amplification mechanisms," *Mechatronics, IEEE/ASME Transactions on*, vol. 15, no. 5, pp. 770–782, 2010.

[117] Ueda, J., MacNair, D., and Brown, E., "Quantized control of compliant cellular actuator arrays for biological movement generation," in *ASME 2012 5th Annual Dynamic Systems and Control Conference joint with the JSME 2012 11th Motion and Vibration Conference*, pp. 233–241, American Society of Mechanical Engineers, 2012.

[118] Villgrattner, T. and Ulbrich, H., "Optimization and dynamic simulation of a parallel three degree-of-freedom camera orientation system," in *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pp. 2829–2836, Oct 2010.

[119] Villgrattner, T. and Ulbrich, H., "Design and control of a compact high-dynamic camera-orientation system," *Mechatronics, IEEE/ASME Transactions on*, vol. 16, no. 2, pp. 221–231, 2011.

[120] Whyte, O., Sivic, J., Zisserman, A., and Ponce, J., "Non-uniform deblurring for shaken images," *International journal of computer vision*, vol. 98, no. 2, pp. 168–186, 2012.

[121] Xu, L. and Jia, J., "Two-phase kernel estimation for robust motion deblurring," in *Computer Vision–ECCV 2010*, pp. 157–170, Springer, 2010.

[122] Xu, L. and Jia, J., "Depth-aware motion deblurring," in *Computational Photography (ICCP), 2012 IEEE International Conference on*, pp. 1–8, IEEE, 2012.

[123] Yang, Q., Wang, C., Gao, Y., Qu, H., and Chang, E. Y., "Inertial sensors aided image alignment and stitching for panorama on mobile phones," in *Proceedings of the 1st international workshop on Mobile location-based service*, pp. 21–30, ACM, 2011.

[124] Young, L. and Stark, L., "Variable feedback experiments testing a sampled data model for eye tracking movements," *Human Factors in Electronics, IEEE Transactions on*, vol. HFE-4, no. 1, pp. 38–51, 1963.

[125] Zhao, X., Gu, Q., Aoyama, T., Takaki, T., and Ishii, I., "A fast multi-camera tracking system with heterogeneous lenses," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2671–2676, Nov 2013.