Real-time Prediction of Dynamic Systems Based on Computer Modeling

Xianqiao Tong

Dissertation submitted to the faculty of the Virginia Polytechnic Institute and State University in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
In
Mechanical Engineering

Tomonari Furukawa, Chair
Mehdi Ahmadian
Saied Taheri
John B. Ferris
Craig A. Woolsey

March 25, 2014
Blacksburg, VA

Keywords: recursive Bayesian estimation, full-field measurement, computer modeling

Copyright 2014

Real-time Prediction of Dynamic Systems Based on Computer Modeling

Xianqiao Tong

ABSTRACT

This dissertation proposes a novel computer modeling (DTFLOP modeling) technique to predict the real-time behavior of dynamic systems. The proposed DTFLOP modeling classifies the computation into the sequential computation, which is conducted on the CPU, and the parallel computation, which is performed on the GPU and formulates the data transmission between the CPU and the GPU using the parameters of the memory access speed and the floating point operations to be carried out on the CPU and the GPU by relating the calculation rate respectively. With the help of the proposed DTFLOP modeling it is possible to estimate the time cost for computing the model that represents a dynamic system given a certain computer. The proposed DTFLOP modeling can be utilized as a general method to analyze the computation of a model related to a dynamic system and two real life systems are selected to demonstrate its performance, the cooperative autonomous vehicle system and the full-field measurement system.

For the cooperative autonomous vehicle system a novel parallel grid-based RBE technique is firstly proposed. The formulations are derived by identifying the parallel computation in the prediction and correction processes of the RBE. A belief fusion technique, which fuses not only the observation information but also the target motion information, has hen been proposed. The proposed DTFLOP modeling is validated using the proposed parallel grid-based RBE technique with the GPU implementation by comparing the estimated time cost with the actual time cost of the parallel grid-based RBE. The superiority of the proposed parallel grid-based RBE technique is investigated by a number of numerical examples in comparison with the conventional grid-based RBE technique. The belief fusion technique is examined by a simulated target search and rescue test and it is observed to maintain more information of the target compared with the conventional observation fusion technique and eventually leads to the better performance of the target search and rescue.

For the full-field measurement system a novel parallel DCT full-field measurement technique for measuring the displacement and strain field on the deformed surface of a structure is proposed. The proposed parallel DCT full-field measurement technique measures the displacement and strain field by tracking the centroids of the marked dots on the deformed surface. It identifies and develops the parallel computation in the image analysis and the field estimation processes and then is implemented into the GPU to accelerate the conventional full-field measurement techniques. The detail strategy of the GPU implementation is also developed and presented. The corresponding software package, which also includes a graphic user interface, and the hardware system consist of two digital cameras, LED lights and adjustable support legs to accommodate indoor or outdoor experimental environments are proposed. The proposed DTFLOP modeling is applied to the proposed parallel DCT full-field measurement technique to estimate its performance and the well match with the actual performance demonstrates the DTFLOP modeling. A number of both simulated and real experiments, including the tensile, compressive and bending experiments in the laboratory and outdoor environments, are performed to validate and demonstrate the proposed parallel DCT full-field measurement technique.

Acknowledgements

Firstly, I would like to thank Professor Tomonari Furukawa as my advisor for his endless support and guidance during my PhD study. He always supports and encourages me to move forward in the academic research and shares his ideas and philosophy with me generously. I will not be able to complete my PhD without Professor Furukawa and there are simply no words to express my gratitude. I also would like to thank my committee members, Professors Mehdi Ahmadian, Saied Taheri, John Ferris and Craig Woolsey who have provided valuable feedbacks and suggestions on my research.

Secondly, I am indebted to Professors Kenzo Nonami and Wenwei Yu for their time during my visit to Chiba University for collaborative research. Thanks to Professor Mark Haley for his arrangement at Chiba University and I am honored to know many good academic researchers there. I have to thank Tim, Josh, Brad, Kevin and Scott who help me conduct rail experiments at Norfolk Southern Inc.

In addition, I am thankful to Drs. Kunjin Ryu and Lin Chi Mak who I worked with for the MAGIC2010 completion. Special thanks to Drs. Shen Hin Lim, Jan Wei Pan and Jinquan Cheng for their advices and help of my work. I am thankful to get to know all the current CMS lab members, Boren, Kuya, Howard, Varun and Affan, and some of CMS alumni. It was very enjoyable to work with all of you.

Finally, I mush thank many people who made my life enjoyable and beautiful during my PhD study. Thanks to my friends, Rui Ma, Lvyin Cai, Yi Li, Bill, Alex, Josh and Heather, who worked and lived at the IALR in Danville. I must thank my parents and my girlfriend and they have always been there and have supported my life with patience and kindness. Thank you.

Dedication

This dissertation is dedicated to my parents, Yong Tong and Fang Zhang, who brought me up with endless love, sacrifice and unlimited patience and encouraged me to pursue a PhD degree. Without their continuing supports I would never be able to accomplish this work.

Contents

1 Introduction **1**

1.1 Real-time prediction 2

1.2 Objective 3

1.3 Approach 3

1.3.1 Part 1: Cooperative autonomous vehicle system 3

1.3.2 Part 2: Full-field measurement system 4

1.4 Original contributions 4

1.5 Publications 5

1.6 Outline of the dissertation 7

2 Literature Review **10**

2.1 Real-time prediction of dynamic systems 10

2.2 Part 1: Recursive Bayesian estimation 12

2.3 Part 2: Full-field measurement 14

2.4 Summary 16

3 DTFLOP Modeling **18**

3.1 Real-time prediction 18

3.2 DTFLOP modeling 20

3.2.1 Data transmission 21

3.2.2 Floating point operation 22

3.3 Summary 23

4 Part 1: Grid-based RBE and Observation Fusion **24**

4.1 Recursive Bayesian estimation 25

4.1.1 Motion model and sensor model 25

4.1.2 Fundamental processes . . . 25
4.2 Grid-based RBE . . . 27
4.2.1 Representation of target space and belief . . . 27
4.2.2 Prediction . . . 28
4.2.3 Correction . . . 29
4.3 Observation fusion . . . 29
4.4 Summary . . . 31

5 Part 1: Parallel Grid-based RBE and Belief Fusion 32
5.1 Parallel grid-based RBE . . . 33
5.1.1 Prediction . . . 33
5.1.2 Correction . . . 34
5.2 Belief fusion . . . 35
5.3 Validation of DTFLOP modeling . . . 36
5.3.1 GPU implementation . . . 36
5.3.2 Data transmission . . . 37
5.3.3 Floating point operations . . . 39
5.3.4 Estimated time cost . . . 39
5.3.5 Validation . . . 40
5.4 Numerical studies . . . 45
5.4.1 Test 1 . . . 45
5.4.2 Test 2 . . . 46
5.4.3 Test 3 . . . 47
5.4.4 Test 4 . . . 50
5.5 Summary . . . 53

6 Part 2: Full-field Measurements 55
6.1 Image analysis . . . 56
6.1.1 Speckle feature . . . 57
6.1.2 Dot feature . . . 58
6.2 Field estimation . . . 59
6.3 Summary . . . 60

7 Part 2: Parallel DCT Full-field Measurements **61**
7.1 Parallel image analysis process 62
7.2 Parallel MLS meshfree interpolation 63
7.3 GPU implementation 65
7.3.1 Shared buffer & look-up table 65
7.3.2 Implementation 66
7.4 System development 67
7.4.1 Hardware system 67
7.4.2 Graphic user interface (GUI) 68
7.5 Numerical studies 73
7.5.1 Performance estimation by DTFLOP modeling 73
7.5.2 Theoretical validation 73
7.5.3 Accuracy evaluation 77
7.5.4 Experimental validation 82
7.6 Railway experiments 84
7.6.1 Indoor laboratorial experiments 86
7.6.2 Outdoor field experiments 90
7.7 Summary 93

8 Conclusions and Future Work **97**
8.1 Conclusions 97
8.1.1 Part 1: Cooperative autonomous vehicle system 97
8.1.2 Part 2: Full-field measurement system 98
8.2 Future work 99

References **101**

A User Manual for Proposed Parallel DCT Full-field Measurement Technique **115**
A.1 A typical example 115
A.2 Preparation procedures 116
A.2.1 Specimen Preparation 116
A.2.2 Lamps and Lamp Settings 118
A.2.3 Cameras, Camera Settings and Calibration 119

List of Figures

3.1 Condition to capture real-time behavior of a dynamic system 19
3.2 Influential factors for computational time cost 20
3.3 Overview of DTFLOP modeling 21

4.1 Observation fusion technique for grid-based RBE 30

5.1 Belief fusion technique for grid-based RBE 36
5.2 GPU implementation of parallel grid-based RBE technique 37
5.3 Time cost of all components for Setup1 with fixed grid space 41
5.4 Time cost of all components for Setup2 with fixed grid space 42
5.5 Time cost of all components for Setup3 with fixed grid space 42
5.6 Time cost of all components for Setup1 with fixed kernel 44
5.7 Time cost of all components for Setup2 with fixed kernel 44
5.8 Time cost of all components for Setup3 with fixed kernel 44
5.9 Speedup vs. kernel radius 46
5.10 Time vs. kernel radius 47
5.11 Speedup vs. grid size 48
5.12 Time vs. grid size 48
5.13 Belief fusion (time vs grid size) 49
5.14 Belief fusion (time vs frequency) 50
5.15 Cooperative search and rescue (Test 4) 52
5.16 Distance to object and information entropy (Test 4) 52

6.1 Schematic diagram of the full-field measurement experimental setup . . 56
6.2 Speckle features and digital image correlation (source: google images, under fair use, 2014) 57

6.3 Dot features 58

7.1 A typical marked dot on captured image 62
7.2 MLS meshfree interpolation 64
7.3 GPU implementation for proposed parallel DCT full-field measurement technique 66
7.4 Hardware system for parallel DCT full-field measurement 67
7.5 GUI for parallel DCT full-field measurement 69
7.6 Widget tabs I 71
7.7 Widget tabs II 72
7.8 Estimated performance of proposed parallel DCT full-field measurement technique by DTFLOP modeling 74
7.9 Mean square error results 75
7.10 Performance of proposed parallel DCT full-field measurement technique 77
7.11 Speedup gain vs number of interpolated points 78
7.12 Full-field displacement measurement: proposed DCT (left) vs FEA (right) 78
7.13 Full-field strain measurement: proposed DCT (left) vs FEA (right) . . . 79
7.14 Marked dots 79
7.15 Standard deviation vs dot size 80
7.16 Accuracy vs dot density 81
7.17 Experimental setup for experimental validation 83
7.18 Strain comparison among strain gauge, extensometer and proposed DCT technique 83
7.19 Captured images for proposed parallel DCT full-field measurement technique 84
7.20 Full-field displacement measurements 85
7.21 Full-field strain measurements 85
7.22 Coordinate frame defined on the rail 86
7.23 Rail bending experiment 87
7.24 Vertical loading for rail bending experiment 87
7.25 Longitudinal strain ε_{xx} in rail bending experiment 88
7.26 Rail compression test 89
7.27 Rail compression results 90

7.28 Outdoor field test of rail 91
7.29 Displacement results of outdoor rail field test 92
7.30 Strain results of outdoor rail field test 92
7.31 Vertical force during train passing 94
7.32 Deformation results measured by proposed parallel DCT full-field measurement technique 95

A.1 Marked dots on an open-hole specimen 118
A.2 Specimen with brightest light 119
A.3 Specimen with some darkness 120
A.4 Shutter speed adjustment (white background) 122
A.5 Shutter speed adjustment (black dots) 123
A.6 Histogram 123
A.7 A typical dot pattern calibration board 125

List of Tables

5.1 Test computer system specifications I 40
5.2 Quantitative results for Test 1 43
5.3 Quantitative results for Test 2 43
5.4 Test computer system specifications II 45
5.5 Major parameters of simulated cooperative search and rescue 51

7.1 Computer configuration for theoretical validation 74
7.2 Equipment specification 82
7.3 Rail dimension 89
7.4 Longitudinal displacement measurements 90
7.5 Experimental parameters for outdoor field experiment 91

Chapter 1

Introduction

The concept of dynamical systems originates from Newtonian mechanics. In natural science and engineering disciplines, the evolution rule of dynamical systems is an implicit model that gives the state of the system with respect to the time. The model is either differential, difference or other time scale equations. Determining the future state requires iterating the model many times by advancing a small time step. To predict the behavior of dynamic systems one needs to iteratively solve such model. The precision of the prediction relies on how well such model is able to represent the real physics of dynamic systems. It is obvious to see that an accurate representation requires a more complicated model which may be computational expensive. Because of the rapid development of fast computing techniques the precise prediction of dynamic systems becomes possible by utilizing modern computers to solve the complicated model.

This dissertation presents a real-time prediction of dynamic systems based on computer modeling. For the purpose of this dissertation, the term real-time prediction is defined as the capability to predict the behavior of the dynamic system quicker or equivalent to its real behavior. Although a variety of techniques exist in the prediction of dynamic systems, this dissertation focuses on the method to model the computer hardware. In this Chapter, the background leading up to the recent interest in the real-time prediction of dynamic systems is briefly explained, followed by the primary objective of this dissertation. The approach taken to achieve this objective is then presented, and the original contributions arising from this work are summarized. Finally the contents of the remaining chapters of this dissertation are outlined.

1.1 Real-time prediction

The real-time prediction of dynamic systems starts in the study of the solar system within modern era. The regularity of such a system on time scales of centuries leads to the possibility of precise prediction of phenomena such as eclipses at the lead time (84). In the last century the real-time prediction of a great range of practical dynamic systems has been attempted and considerable progress has been made. Perhaps the best known examples of these are the turbulent fluids such as the atmosphere and ocean as well as the earthquake prediction for which the dynamic systems can be considered complicated because of its non-linear nature. The real-time prediction of dynamic systems typically suffers from two difficulties. The primary difficulty is that the model used to depict the physics of dynamic systems may have certain inadequacies as a representation of reality, which is also known as model errors. The other one is that there may be a strict time constraint to solve the relation and the corresponding computational requirement may be expensive.

In general model errors are almost by definition quite hard to study in dynamic systems since they are caused by quite diverse factors which are not very well understood. The issue of model errors tends primarily to be an engineering rather than a theoretical study (109). On the other hand, in order to meet the strict time requirement of the real-time prediction by solving complicated model one needs to adopt high performance computing techniques or supercomputers. Nowadays, with the advanced development of semiconductor related innovations even personal computers are able to solve the complicated model of dynamic systems in a timely manner. A lot of efforts have been made to implement the complicated model into an efficient way with respect to the fast computation in a computer, such as parallelizing independent computations, reversing the order of certain computation to avoid duplicate memory copy and so on.

Besides the efficient implementation of a complicated model of dynamic systems, the computational capability of a computer largely relies on its hardwares including the computational units, the physical memories, the communication system that transfers data between components inside a computer and so on. However, the studies which relate the capabilities of those computer hardwares with the actual implementation of dynamic systems are limited. Such studies are expected to be important and able to benefit the real-time prediction of dynamic systems.

1.2 Objective

The objective of this dissertation is to develop the computer modeling which can be applied to predict dynamic systems in real-time and to demonstrate its performance with the applications of real life examples.

1.3 Approach

In order to achieve this objective, it is necessary to develop the computer modeling which represents the computational processes in the hardwares of a personal computer. The developed computer modeling classifies the computation of the implementation of the model which represents a dynamic system into two classes: the sequential and the parallel computation. The sequential computation is generally performed in Central Processing Unit (CPU) whereas the parallel computation is conducted in Graphics Processing Unit (GPU), which is widely used for the parallel computation in a personal computer although is originally used for sole computer graphic related applications. The developed modeling formulates the data transmission between the CPU and the GPU by the parameters of the memory access speed as well as the floating point operations to be carried out in the CPU and the GPU by relating the calculation rate respectively. Given the specification of a computer it is thus possible to estimate the time cost for computing the model that represents a dynamic system. The developed computer modeling can be utilized as a general method to analyze the computation of a model related to a dynamic system. To demonstrate the performance of the developed modeling two real life example systems are selected: the cooperative autonomous vehicle system and the full-field measurement system.

1.3.1 Part 1: Cooperative autonomous vehicle system

In the cooperative autonomous vehicle system a grid-based estimation technique, which deals with high uncertainty, is developed by identifying and analyzing the sequential and the parallel computation process, respectively. To further reduce the uncertainty of the estimation a belief fusion technique is developed by reducing the cooperative communication traffic but at the same time keeping the high level of useful estimation information. The developed computer modeling is then applied to the developed

estimation technique for the cooperative autonomous vehicle system to validate its performance. The validation starts with the identification of the data transmission process between the CPU and the GPU and the floating point operations process in the CPU and the GPU respectively, then constructs the formulation for the estimated time cost for each process and finally compare the estimated time cost with the actual time cost to compute the model of the system using a computer.

1.3.2 Part 2: Full-field measurement system

In the full-field measurement system a dot centroid tracking (DCT) technique is developed to measure the displacement and strain field of the surface deformation of a structure. In the DCT technique digital cameras keep capturing the images of the surface of a structure, which is applied by a number of marked dots, as the sole measurement. The developed DCT technique is consist of two processes: the image analysis process and the field estimation process. The image analysis process measures the darkness of each pixel, the smallest element of an image, in grayscale, identifies and extracts the dots marked on the structure, derives the dot centroids by pixel darkness information in the marked dots and computes the displacement measurements of the marked dots by tracking their centroids. The field estimation process defines a fine mesh for the surface of a structure and interpolates the displacement measurements of marked dots to the nodes of the mesh using moving least square meshfree shape function to construct the displacement field. The strain is by definition the differentiation of the displacement and thus the strain field is derived from the displacement field using the partial derivative shape function. The sequential and the parallel computation are identified in the image analysis process and the field estimation process respectively. The developed computer modeling is then applied to predict the computational time cost of the developed DCT technique before the actual implementation given a computer and then to demonstrate its superiority compared with the conventional techniques.

1.4 Original contributions

The principal contributions of this dissertation are enumerated as follows:

- The computer modeling that formulates data transmission and floating point operations, named as DTFLOP modeling, is presented.
- The performance of the proposed DTFLOP modeling is demonstrated in the cooperative autonomous vehicle system and the full-field measurement system, respectively.

The original contributions of the two example systems are summarized as follows:

- Cooperative autonomous vehicle system:
  - A novel grid-based recursive Bayesian estimation (RBE) technique which deals with non-linear systems using GPU is presented and its real-time advantage is demonstrated.
  - A belief fusion technique for the autonomous vehicle cooperation is presented and its superiority is demonstrated through the experiments of simulated autonomous vehicle cooperation.
  - The proposed DTFLOP modeling is validated and demonstrated for the proposed grid-based RBE technique.
- Full-field measurement system:
  - A novel DCT technique using GPU for full-field displacement and strain measurement of the surface of a structure is presented.
  - The advantages of the speed and accuracy of the proposed DCT technique is demonstrated by a series of practical experiments.
  - The proposed DTFLOP modeling is applied to the proposed DCT technique to predict the potential computational speedup advantage.

1.5 Publications

To date, components of the dissertation have been presented in the following publications:

[1] **Xianqiao Tong** and Tomonari Furukawa, "Hybrid DIC-DCT Method for Full-field Displacement and Strain Measurements", in preparation

[2] **Xianqiao Tong** and Tomonari Furukawa, "Real-time Noncontact Measurement of Surface Deformation of Rails", in preparation

[3] **Xianqiao Tong**, Tomonari Furukawa and Hugh F Durrant-Whyte, "Computer Modeling for Parallel Grid-based Recursive Bayesian Estimation Parallel Computation using Graphics Processing Unit ", *Journal of Uncertainty Analysis and Applications*, 2013

[4] **Xianqiao Tong**, Tomonari Furukawa and Saied Taheri, "Speed Enhancement of Displacement and Strain Field Measurement using Graphics Processing Unit", *ASME RTD Fall Technical Conference* (RTDF2012), Omaha, NE, USA, Oct 16-18, 2012

[5] **Xianqiao Tong**, Tomonari Furukawa and Hugh F Durrant-Whyte, "Modeling of Computer Performance for Real-time Parallel Grid-based Recursive Bayesian Estimation", *Second IASTED International Conference on Robotics* (Robo2011), Pittsburgh, PA, USA, Nov 7-9, 2011

[6] **Xianqiao Tong** and Tomonari Furukawa, "Using RGB-D Sensors for Grid-based Recursive Bayesian Estimation", *International Conference on Intelligent Unmanned Systems* (ICIUS2011), Chiba, Japan, Oct 31-Nov 2, 2011

[7] **Xianqiao Tong**, Tomonari Furukawa and Saied Taheri, "Real-time Displacement and Strain Measurement of Rail and Wheel Surfaces Based on Image Processing Technique", *ASME 2012 Joint Rail Conference* (JRC2012), Philadelphia, PA, USA, Apr 17-19, 2012

[8] Tomonari Furukawa, **Xianqiao Tong**, et al, "Implementation and Demonstration of SLAM by Autonomous Car Using Grid-based Scan-to-Map Matching", *International Conference on Intelligent Robots and Systems* (IROS2014), Chicago, Illinois, USA, Sep 14-18, 2014, submitted

[9] Kunjin Ryu, **Xianqiao Tong**, Tomonari Furukawa, Gamini Dissanayake and Jaime Valls Miro, "Map-based Semi-Autonomous Strategy for Urban Search and Rescue", *International Journal of Intelligent Unmanned Systems*, 2013, second review

[10] Tomonari Furukawa, **Xianqiao Tong**, et al., "Autonomous Robots for Monitoring Environments of Damaged Nuclear Power Plants", *International Conference on Intelligent Unmanned Systems* (ICIUS2011), Chiba, Japan, Oct 31-Nov 2, 2011

[11] Tomonari Furukawa, **Xianqiao Tong**, et al, "Parallel Grid-based Method and Belief Fusion: Real-time Cooperative Non-Gaussian Estimation", *Sixth International Conference on Industrial and Information Systems* (ICIIS2011), Sri Lanka, Aug 16-19, 2011

[12] Yi Xu, **Xianqiao Tong**, et al, "A Vision-Guided Robot Manipulator for Surgical Instrument Singulation in a Cluttered Environment", *International Conference on Robotics and Automation* (ICRA2014), Hong Kong, China, May 31-Jun 7, 2014, accepted

[13] Kunjin Ryu, **Xianqiao Tong** and Tomonari Furukawa, "The Platform-and-Hardware-in-the-loop Simulator" *A Workshop on Frontiers of Real-World Multi-Robot Systems*, Durham, NC, USA, Oct 10-11, 2011

[14] Tomonari Furukawa, Lin Chi Mak, Kunjin Ryu, **Xianqiao Tong** and Gamini Dissanayake, "Bayesian Search, Tracking, Localization and Mapping: A Unified Strategy for Multi-task Mission" *INFORMS Annual Meeting*, Charlotte, NC, USA, November 13-16, 2011

[15] Tomonari Furukawa, Lin Chi Mak, Kunjin Ryu and **Xianqiao Tong**, "The Platform-and-Hardware-in-the-loop Simulator for Multi-Robot Cooperation" *Performance Metrics for Intelligent Systems Workshop* (PerMIS'10), Baltimore, MD, USA, Sep 28-30, 2010

1.6 Outline of the dissertation

This dissertation is organized as follows:

- Chapter 2 reviews previous work on the real-time prediction techniques of dynamic systems, the RBE techniques for the state estimation of the cooperative autonomous vehicle system and the full-field displacement and strain measurement techniques for the surface deformation of a structure. These research efforts

further support the claims provided in this introductory chapter, and thus signify the objective of this dissertation.

- Chapter 3 presents the DTFLOP modeling for the real-time prediction of dynamic systems. The condition to capture the real-time behavior of a dynamic system is first described. The relationship between speed or accuracy and the performance of the real-time prediction is then presented. Finally formulations of the data transmission among processors and the floating point operations in each processor are presented by relating the computational implementation with hardware parameters given a computer.

- Chapter 4 describes the RBE techniques for the state estimation by updating the belief using the motion model and the observation model. Following this, the observation fusion technique for the autonomous vehicle cooperation and the corresponding formulations are presented.

- Chapter 5 presents a novel parallel grid-based RBE technique by identifying its sequential and parallel computation. A belief fusion technique is then presented to overcome the shortcomings of the conventional observation fusion technique and finally the proposed DTFLOP modeling is validated by the proposed grid-based RBE technique. This Chapter also presents numerical studies of the proposed grid-based RBE and belief fusion techniques and further compares with the conventional techniques.

- Chapter 6 describes the overall of the full-field displacement and strain measurement techniques for the surface deformation of a structure. The general formulations for the image analysis and field estimation processes are presented, respectively. The full-field measurement technique based on DCT is further presented and the computer vision techniques utilized are described.

- Chapter 7 presents a novel parallel DCT technique to measure the displacement and strain field for the surface deformation of a structure. The proposed parallel DCT technique starts with identifying the sequential and parallel computation process of the conventional DCT technique. The proposed DTFLOP modeling is then applied to predict the performance of the proposed DCT technique before the actual implementation. A number of experimental results in simulated and

real environments are presented to investigate and demonstrate the performance of the proposed parallel DCT technique in the end.

- Chapter 8 summarizes the contributions of the research presented by this dissertation and discusses areas fro potential future work.

Chapter 2

Literature Review

This Chapter reviews the past contributions concerned with the techniques discussed in this dissertation. Dynamic systems are described by constructing a mathematical model which represents its physics. With the help of advanced computing techniques the real-time prediction of dynamic systems becomes possible. The techniques which predict real-time behavior of dynamic systems are discussed in Section 3.1. As mentioned in the introductory section the proposed modeling technique for real-time prediction is validated and further demonstrated in the two application of real life examples. The first application is the cooperative autonomous vehicle system and it deals with the problem of probabilistically estimating the state of targets with the cooperation of multiple autonomous vehicles. In this scenario the recursive Bayesian estimation techniques, which estimate the state of a dynamic system by recursively using the motion model and the incoming observations, are reviewed in Section 4.1. The second application is the full-field measurement system which measures the surface deformation of a structure and the measurements are utilized to indicate the health of the structure. Section 2.3 covers those techniques to perform the full-field measurement.

2.1 Real-time prediction of dynamic systems

Dynamic systems are understood and described in the form of mathematical models which depict the physical behaviors. As the rapid development of advanced computing techniques researchers start to predict the dynamic systems by utilizing the mathematical model in the discrete time form. There are a full of history which advances

those techniques in the field of real-time prediction. Simulation is the other term which describes the technique to predict current or future behavior of dynamic systems in real-time. Physical simulation refers to the simulation in which physical objects are substituted for the real system (86). These physical objects are chosen because they are cheaper or smaller than the actual system. Interactive simulation or human in the loop simulation refers to the simulation which involve the human activities, such as the flight simulator or the driving simulator.

Computer simulation is to model a dynamic system on a computer so that it can be studied to see how the system works. By changing the variables in the computer simulation one can predict the current or future behavior of the dynamic system. It is seen to be a tool to virtually investigate and predict the behavior of a dynamic system. Computer simulation becomes a part of modeling many natural systems in physics, chemistry and biology, and human systems in economics and social science as well as in engineering to gain insight into the operation of those systems (117). The behavior of dynamic systems is predicted and investigated by changing the parameters in the computer simulation. Computer simulation varies from computer programs that run a few minutes to network-based groups of computers running for hours to ongoing simulations that run for days. The scale of systems being simulated by the computer simulation has far exceeded anything possible using traditional paper-and-pencil mathematical modeling (46). Computer simulation developed tightly with the rapid growth of the computer, following its first large-scale deployment during the Manhattan Project in World War II to simulate the process of nuclear detonation. In terms of the attribute of the dynamic systems the computer simulation can be classified into the stochastic simulation and the deterministic simulation (48). Stochastic simulation operates with variables that can change with certain probability and the behavior of the dynamic system predicted in the form of the most probable estimates with a probability. As mentioned above 12 hard spheres was simulated in the Manhattan Project using the Monte Carlo technique. Monte Carlo techniques rely on repeated sampling to obtain numerical results and typically run many times over in order to obtain the distribution of an unknown probabilistic entity. Monte Carlo techniques are generally useful for predicting the behavior of the dynamic systems which have many coupled degrees of freedom, such as guilds, disordered materials, strongly coupled solids, and cellular structures (64). On the other hand, the deterministic simulation contains no random

variables and no degree of randomness. The resulted outputs are unique in nature with respect to the dynamic system and parameters.

Nowadays the computational capability of personal computers has been increased dramatically and the simulation of dynamic systems becomes possible on a personal computer. The mathematical model which represents the dynamic system is numerically executed in the discrete time domain on the computer to predict its behavior. The condition of capturing the real-time behavior of the dynamic system is the computational time cost for the mathematical model of the dynamic system is less or equal to its physical counterpart, the time increment in the mathematical model (117). Thus, it is important to understand the computational capability of a given computer. Traditionally computational units on a personal computer only refer to the central processing units (CPUs), single core or multiple cores. The computer program which corresponds to the mathematical model of the dynamic system is executed on the CPU. Computational parallelism is performed by utilizing the multiple cores of a CPU but only limiting up to 8 cores. The computational capability of a computer can be modeled by relating the computer program to specifications of the CPU and memory bandwidth (87). On the other hand, graphics processing units (GPUs) are well known for its capability to execute complicated graphic computation and they have hundreds of cores to aid parallel computation. Recently GPUs can be programmed explicitly to perform general purpose parallel computation other than graphic related computation effectively and become powerful computational units along with CPUs. It is necessary to construct a novel computer modeling which considers not only the computational power of CPUs but also the additional computational capability GPUs brought in.

2.2 Part 1: Recursive Bayesian estimation

Recursive Bayesian estimation (RBE) is a probabilistic estimation technique to recursively update the state of a dynamic system. In the autonomous vehicle system RBE is used by the autonomous vehicle to estimate the state of moving targets using a probability density function or the belief. RBE allows the estimation of the belief of the state by updating the belief both in time and observation (104). There are two fundamental processes for the RBE: the prediction process and the correction process. The

prediction process updates the belief by the motion model whereas the correction process updates the belief through the current observation. If the observation is available the accuracy of the RBE can be maintained by the correction process using the valid observations. When the observation is not available the accuracy of the RBE heavily relies on the prediction process and the error accumulates due to the lack of the valid observation for the correction process and thus the belief is known to become heavily non-Gaussian. Recent years have seen the increasing need for non-Gaussian RBE.

One of such techniques is the modified ensemble Kalman filter (EnKF). The EnKF allows non-Gaussian estimation by minimizing a cost function defined by a non-Gaussian observation error with a pre-conditioned conjugate gradient method (43). Langevin Markov Chain Monte Carlo (MCMC) method, which represents the non-Gaussian belief by sampling it using a Markov chain and Langevin equation, could be a non-Gaussian RBE technique (7). Another sampling method is the interactive Particle Filter (PF), which is able to flexibly mitigate the belief space complexity (25). An Ensemble Kalman-Particle Predictor-Corrector Filter is a hybrid method that combines the advantages of EnKF and PF and is be able to effectively deal with high-dimensional non-Gaussian problems (72). A tree-based estimator approximates the posterior belief distribution at multiple resolutions to also be effective for high dimensional problems (100) whereas Maximum Likelihood State Estimation Method could also achieve non-Gaussian RBE (49) by using a finite Gaussian mixture model.

Out of all the non-Gaussian RBE techniques the grid-based RBE technique is one of the most accurate but time comsuming techniques since the entire space needs to be spatially discretized (12). The good accuracy is obtained by the subtle discretization of the target space but leads to an inefficient computation at the same time. Furukawa, et al. (30) (62) refined the grid-based RBE by developing a more general element-based RBE. The generalized element can help accurately represent the arbitary target space with only the small number of elements compared with the grid-based RBE technique so as to reduce the computation of the RBE. Lavis proposed an enhanced grid-based RBE that allows the update of not only the belief but also the target space (63). Because of the dynamic adjustment of the target space, the computation of the RBE is additionally reduced. Further, the parallel grid-based RBE has been proposed and it significantly accelerated the computation of the RBE and made its real-time implementation possible by utilizing the GPU's strong parallel computational capability (31). Despite that these

efforts successfully reduce the computation of the RBE to achieve the fast RBE, the accuracy of the RBE is not well kept when the prediction process dominates the RBE during the no-observation period. The time cost of one iteration of the RBE becomes critical for overcoming this issue because that only if it matches the time increment of the discrete target motion model the RBE can maintain the accuracy during the no-observation period.

2.3 Part 2: Full-field measurement

The accuracy measurement of the deformed surface of a structure is important to understand the health of the structure. There are primary two types of sensor which can achieve the measurement. Contact sensors, on the one hand, are able to provide the accuracy measurement of the deformed surface but the installation and the calibration require tedious and intensive labor work. Also the nature of the contact sensors may affect the behavior of the measured structure and lead to the inaccuracy measurement (9, 18, 21). On the other hand, the non-contact sensors are widely used for the surface deformation measurement and the structural health monitoring applications but suffer from the inaccuracy measurement (22, 99). For example in order to measure the stress in the rail one has to cut the rail and measure the elongation to get the accuracy measurement (65). With the advanced technology developed in the non-contact sensors the accuracy measurement of the deformed surface of a structure becomes feasible.

Optical sensors have been utilized in the structural health monitoring applications for the last several decades but not been widely used since the accuracy of the measurement is not comparable with the traditional contact sensors. During the last decade owing to the development of the semiconductor technology the resolution of the optical sensors improves dramatically and the sensor noise is reduced as well. A significant advantage of the optical sensors compared with the traditional sensors is that its capability to measurement the field of the deformed structure instead of that on one spot. The capability to achieve the full-field measurement benefits the understanding the behavior of deformed surface and eventually results in the higher-quality analysis of the health of a structure.

During past decades various non-contact optical techniques for the full-field measurement have been presented, including both interferometric techniques, such as holo-

graphic interferometry (111), speckle interferometry (55) and moire interferometry (42, 74), and non-interferometric techniques, such as the grid method (94) and digital image correlation (DIC) (116). Interferometric techniques require a coherent light source and a vibration-isolated platform to conduct full-field measurement in the laboratory. Interferometric techniques measure the deformation by recording the phase difference of the scattered light wave from the surface before and after the deformation. The measurement is represented in the form of the fringe patterns and thus fringe processing and phase analysis techniques are required in order to get the displacement and strain measurement. Non-interferometric techniques determine the surface deformation by comparing the gray intensity changes of the surface before and after the deformation, and generally have less strict requirements for the experimental conditions.

As a representative non-interferometric optical technique for the full-field measurement, the DIC technique has been widely accepted and commonly used as a powerful and flexible tool for the surface deformation measurement. It directly provides field displacement and strain by comparing the captured images of the surface before and after the deformation. In principle, DIC is a full-field measurement technique based on the digital image processing and numerical computing. DIC is first developed by Peters (93) in 1981 when digital image processing and numerical computing were still not advanced in development. There are a number of DIC techniques developed subsequently, such as digital speckle correlation method (116, 118), texture correlation (11), computer-aided speckle interferometry (CASI) (20) and electronic speckle photography (ESP) (95). Compared with the interferometric techniques DIC requires simple experimental setup and preparation, only white light source or natural light and provide the wide range of measurement sensitivity and resolution which relies on the different type of digital cameras. DIC full-field measurement technique has been widely used in the material characterization, structural health monitoring and modeling of the dynamic motion of a structure. Its capability of both the two dimensional and three dimensional full-field measurements draws large interest of the related company and several commercial packages have been in the market, such as Correlated Solutions (96), Trilion Quality Systems (110) and GOM optical measuring techniques (90).

Iliopoulos (51, 52) presents a dot centroid tracking (DCT) technique for full-field displacement and strain measurement by tracking the centroids of the marked dots

on the measured surface. The DCT technique has the advantage of the light computational load on its numerical computing process. The marked dots are attached on the measured surface and the positions of those dots are derived by the pixel intensity on the captured image. The displacement and strain field measurement is computed from the interpolation from the true measurement of the marked dots and there are a number of interpolation techniques to be selected for different requirements and applications. Pan and Furukawa applies the DCT full-field measurement technique in the characterization of composite materials and develops the data fusion approach to improve the accuracy of the measurement (75, 82). DCT techniques are suitable for the full-field measurement applications due to the fact that its easy setup and implementation and the accuracy of the measurement can be easily adjusted by utilizing cameras with different resolution. Although there are a lot of efforts have been made for the DCT techniques the speed of the DCT is still not fast enough to provide an accuracy full-field measurement in real-time. There has still not seen a complete product in the market which can provide the accurate and fast full-field measurement for the surface of a structure.

2.4 Summary

This Chapter reviewed the past contributions concerned with the techniques discussed in this dissertation. Dynamic systems are described by constructing a mathematical model which represents its physics. With the help of advanced computing techniques the real-time prediction of dynamic systems becomes possible. The techniques which predict real-time behavior of dynamic systems are discussed in Section 3.1. As mentioned in the introductory section the proposed modeling technique for real-time prediction is validated and further demonstrated in the two application of real life examples. The first application is the cooperative autonomous vehicle system and it deals with the problem of probabilistically estimating the state of targets with the cooperation of multiple autonomous vehicles. In this scenario the recursive Bayesian estimation techniques, which estimate the state of a dynamic system by recursively using the motion model and the incoming observations, are reviewed in Section 4.1. The second application is the full-field measurement system which measures the surface deformation of

a structure and the measurements are utilized to indicate the health of the structure. Section 2.3 covers those techniques to perform the full-field measurement.

Chapter 3

DTFLOP Modeling

This Chapter presents a computer modeling for the real-time prediction of dynamic systems to estimate the time cost of a computational implementation of a dynamic system by relating the hardware parameters with the computation of the implementation. The proposed computer modeling classifies the computation into the sequential computation and the parallel computation and expects those computation to be executed on the CPU and the GPU, respectively. The time cost of the computational implementation of a dynamic system is modeled by the time cost of the data transmission among the processors and the time cost of the floating point operations in each processor.

This Chapter is organized as follows. Section 3.1 describes the condition to capture the real-time behavior of a dynamic system and the relationship between the speed or accuracy and the performance of the real-time prediction is then presented. The formulations of the data transmission among processors and the floating point operations in each processor by relating the computational implementation with hardware parameters given a computer are presented in Section 3.2.

3.1 Real-time prediction

As known in the previous introductory Chapter a dynamic system is described in a mathematical form and further implemented numerically in a computer program in the discrete form. Assume that a dynamic system is described in the form of differential equations. The state of the dynamic system is defined as $\mathbf{x}$ and its derivative is $\dot{\mathbf{x}}$. Figure 3.1 shows the comparison between the real behavior and the predicted behavior

of the dynamic system. In the Figure 3.1, Δt represents the computational time cost of the implementation of the dynamic system and Δt_p represents the physical counterpart of Δt. The condition to capture the real-time behavior of the dynamic system is given by:

$$\Delta t \leq \Delta t_p, \tag{3.1}$$

which means that the computation has to be performed equal or faster than the physical counterpart of the dynamic system. It is obvious that the speed of computation relies on not only the numerical implementation of the dynamic system but also the computational capability or hardware specifications of a computer.

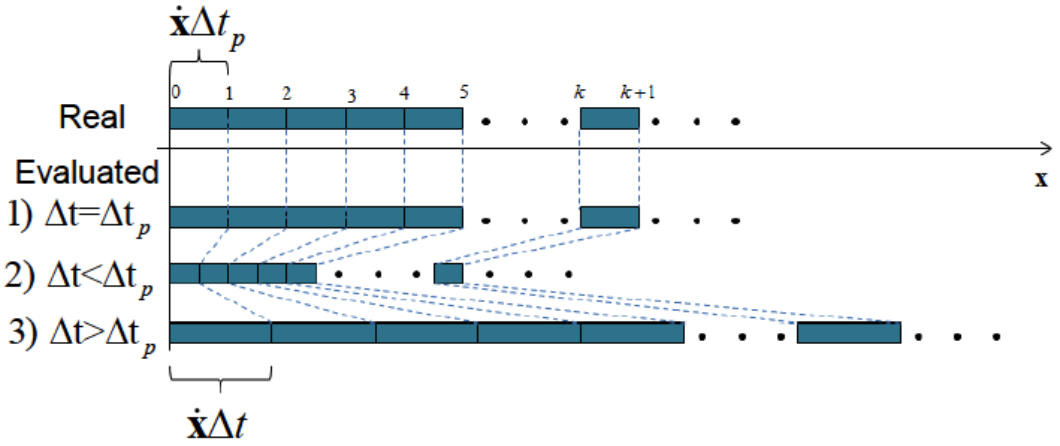
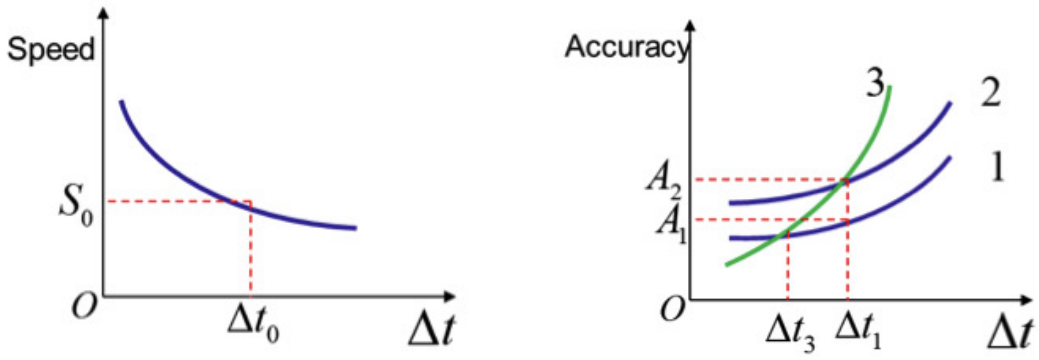


Figure 3.1: Condition to capture real-time behavior of a dynamic system

Figure 3.2(a) shows the relationship between the computational capability or speed given a certain computer specification and the actual computational time cost of an implementation of a dynamic system. It is shown that the computational speed is inversely related with the actual computational time cost. On the other hand, Figure 3.2(b) shows the relationship between the accuracy of an implementation of the dynamic system and the actual computational time cost of the implementation. As described in Figure 3.2(b) one can improve the implementation of a dynamic system to achieve better accuracy, from A_1 in the curve 1 to A_2 in the curve 2, and remain the same computational time cost Δt_1. The improved implementation can reduce the computational time cost, from Δt_1 in the curve 2 to Δt_3 in the curve 3, by remaining the original accuracy A_1. Regard to the condition to capture the real-time behavior of a

dynamic system, Equation (3.1), both increasing the speed and improving the accuracy would benefit the real-time prediction of a dynamic system.



(a) Speed vs Computational time cost (b) Accuracy vs Computational time cost

Figure 3.2: Influential factors for computational time cost

3.2 DTFLOP modeling

As considered a computational implementation of a dynamic system one can classify the computation into the sequential computation and the parallel computation. In a typical personal computer, which is consist of one CPU and one GPU as the computational units, the sequential computation is performed by the CPU whereas the parallel computation is performed by the GPU. Assume that there is no overlap time between the sequential and parallel computation. The total time cost of an implementation on a computer can be modeled as the time cost of data transmission and the time cost of the computation in both the CPU and the GPU. The proposed DTFLOP modeling, acronym of **D**ata **T**ransmission and **FL**oating point **OP**erations, is shown in Figure 3.3. It describes the sequential computation on the CPU, the parallel computation on the GPU and the data transmission. Therefore, the total time cost of an implementation of a dynamic system is given by:

$$\Delta t = \Delta t_{\text{trans}} + \Delta t_{\text{C}} + \Delta t_{\text{G}}, \tag{3.2}$$

where Δt_{trans} represents the time cost of the data transmission, Δt_{C} represents the computational time cost on the CPU and Δt_{G} represents the computational time cost

on the GPU. The time cost of the data transmission is consist of not only the data transmission between the CPU and the GPU but also the one inside the CPU and the GPU with respect to the physical memory specification.

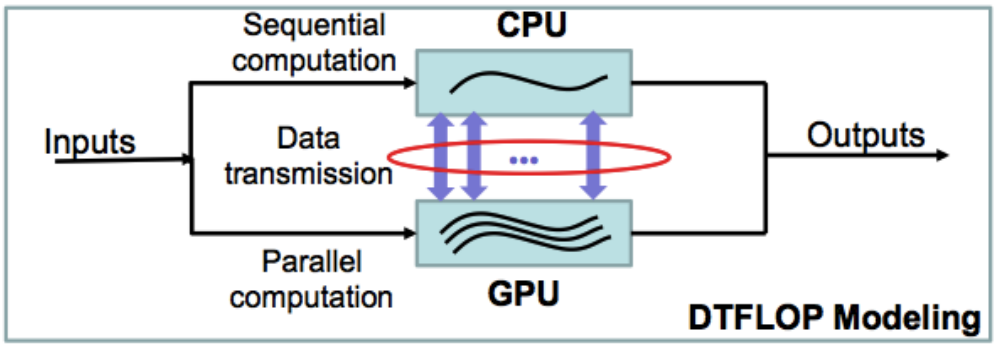


Figure 3.3: Overview of DTFLOP modeling

3.2.1 Data transmission

The amount of the data transmitted in the unit of bytes is defined as

$$A = PN \tag{3.3}$$

where P is the precision of the numerical representation (e.g. P is 8 bytes per numerical unit for type "double") and N is defined as the number of data transmitted. Since the precision is constant, derivation of the amount of data transmitted can be made in terms of the number of data transmitted. The time cost of the data transmission can be classified into three categories given a typical computer consist of one CPU and one GPU. The time cost of the data transmission from the CPU to the GPU and the time cost from the GPU to the CPU fall in the two categories. Since the GPU has a hierarchy of the global memory and the local memory the third category is the time cost of the data transmission inside the GPU. Thus, the time cost of data transmission is given by

$$\Delta t_{\text{trans}} = \Delta t_{\text{CG}} + \Delta t_{\text{GC}} + \Delta t_{\text{GG}}, \tag{3.4}$$

where Δt_{CG}, Δt_{GC} and Δt_{GG} represents the time cost of the data transmission from the CPU to the GPU, from the GPU to the CPU and inside the GPU, respectively.

Each component of the time cost of the data transmission can be further broken down with respect to the number of data transmitted and the physical hardware parameters. The time cost of the data transmission from the CPU to the GPU is given by

$$\Delta t_{\mathrm{CG}} = P\frac{N_{\mathrm{CG}}}{B_{\mathrm{CG}}}, \tag{3.5}$$

where N_{CG} and B_{CG} are the the total number of the data transmitted and the copy bandwidth with the unit of bytes/sec from the CPU's memory to the GPU's global memory respectively. The time cost of the data transmission from the GPU to the CPU is given by

$$\Delta t_{\mathrm{GC}} = P\frac{N_{\mathrm{GC}}}{B_{\mathrm{GC}}}, \tag{3.6}$$

where N_{GC} and B_{GC} are the the total number of the data transmitted and the copy bandwidth with the unit of bytes/sec from the GPU's global memory to the CPU's memory respectively. The time cost of the data transmission inside the GPU is given by

$$\Delta t_{\mathrm{GG}} = P\frac{N_{\mathrm{GG}}}{B_{\mathrm{GG}}}, \tag{3.7}$$

where N_{GG} and B_{GG} are the the total number of the data transmitted and the copy bandwidth with the unit of bytes/sec between the GPU's global memory to the GPU's local memory respectively. Due to that the copy bandwidth from the GPU's global memory to the GPU's local memory and the one in opposite direction are the same one does not need to discriminate the copy bandwidth in the two directions. It is to be noted here that these parameters of copy bandwidths are inherent for a given computer and can be determined experimentally.

3.2.2 Floating point operation

The computational capability of a processor, CPU or GPU, is defined as the speed for performing floating point operations. FLOPS, acronym of **FL**oating point **O**perations **P**er **S**econd, is a typical measure for the computational capability of a processor. The time cost of the sequential computation performed by the CPU is given by

$$\Delta t_{\mathrm{C}} = \frac{N_{\mathrm{C}}}{V_{\mathrm{C}}} \tag{3.8}$$

where N_{C} is the number of floating point operations performed by the CPU and V_{C} is the computation rate of the CPU with the unit of FLOPS. Similarly, the time cost of the parallel computation performed by the GPU is given by

$$\Delta t_{\mathrm{G}} = \frac{N_{\mathrm{G}}}{V_{\mathrm{G}}} \tag{3.9}$$

where N_{G} represents the number of floating point operations performed by the GPU and V_{G} is the computation rate of the GPU with the unit of FLOPS. It is also to be noted here that the computation rates, V_{C} and V_{G}, are inherent for the specific CPU and GPU configuration and can be determined experimentally.

3.3 Summary

In the beginning of this Chapter, the condition to capture the real-time behavior of a dynamic system was described and then the relationship between the speed or accuracy and the performance of the real-time prediction was analyzed. The performance of the real-time prediction would be benefited both by increasing the speed of the implementation and improving the accuracy. The DTFLOP modeling, which identifying the sequential computation and the parallel computation, has been presented. The time cost of an implementation of a dynamic system was modeled by the time cost of the data transmission and the time cost of the computation in the CPU or the GPU and the corresponding formations have been derived in the end.

Chapter 4

Part 1: Grid-based RBE and Observation Fusion

This Chapter describes the grid-based RBE and the observation fusion techniques for the target estimation in two dimensional space. The RBE techniques are known as the ability to probabilistically estimate the state of a target with uncertainty. The prediction and correction processes are presented as the two fundamental processes of the RBE technique. In order to deal with the non-Gaussian system the grid-based RBE technique is presented as it discretizes the target space in terms of grid cells. The accuracy of the grid-based RBE technique relies on the resolution of the discretization. The observation fusion technique for cooperative estimation is also presented and it fuses the observation from all the valid observations and synchronizes for all the autonomous vehicles.

This Chapter is organized as follows. Section 4.1 firstly describes the motion model and the sensor model of the system and then derives the formulations of the prediction process and correction process of the RBE technique. In addition, the formulations for the grid-based RBE technique are presented in Section 4.2. In the end, the observation fusion technique is discussed and the corresponding formulations are presented.

4.1 Recursive Bayesian estimation

4.1.1 Motion model and sensor model

Consider the jth target, t_j, out of total n_t targets, the motion of which in discretely given by

$$\mathbf{x}_{k+1}^{t_j} = \mathbf{f}^{t_j}(\mathbf{x}_k^{t_j}, \mathbf{u}_x^{t_j}, \mathbf{w}_k^{t_j}), \tag{4.1}$$

where $\mathbf{x}_k^{t_j} \in \mathcal{X}^{t_j}$ is the state of the target t_j at time step k, $\mathbf{u}_k^{t_j} \in \mathcal{U}^{t_j}$ is the set of control inputs for the target t_j, and $\mathbf{w}_k^{t_j} \in \mathcal{W}^{t_j}$ is the system noise of the target t_j.

The ith sensor platform or autonomous vehicle, s_i, out of total n_s sensor platforms carries a sensor to observe target t_j. The motion model of the s_i sensor platform is given by

$$\mathbf{x}_{k+1}^{s_i} = \mathbf{f}^{s_i}(\mathbf{x}_k^{s_i}, \mathbf{u}_k^{s_i}) \tag{4.2}$$

where $\mathbf{x}_k^{s_i} \in \mathcal{X}^{s_i}$ and $\mathbf{u}_k^{s_i} \in \mathcal{U}^{s_i}$ represent the state and control input of the sensor platform s_i, respectively.

The sensor has an "observable region" as its physical limitation and the observable region is determined not only by the properties of the sensor but also the properties of the target. Defining the probability of detection $0 < P_d(\mathbf{x}_k^{t_j}|\mathbf{x}_k^{s_i}) \leq 1$ from these factors as a reliability measure for detecting the target t_j, the observable region can be expressed as ${}^{s_i}\mathcal{X}_o^{t_j} = \{\mathbf{x}_k^{t_j}|0 < P_d(\mathbf{x}_k^{t_j}|\mathbf{x}_k^{s_i}) \leq 1\}$. Accordingly, the state of the target t_j observed from the sensor platform s_i, ${}^{s_i}\mathbf{z}_k^{t_j} \in \mathcal{X}^{t_j}$, is given by

$${}^{s_i}\mathbf{z}_k^{t_j} = \begin{cases} {}^{s_i}\mathbf{h}^{t_j}(\mathbf{x}_k^{t_j}, \mathbf{x}_k^{s_i}, {}^{s_i}\mathbf{v}_k^{t_j}) & \mathbf{x}_k^{t_j} \in {}^{s_i}\mathcal{X}_o^{t_j} \\ \\ \emptyset & \mathbf{x}_k^{t_j} \notin {}^{s_i}\mathcal{X}_o^{t_j} \end{cases} \tag{4.3}$$

where ${}^{s_i}\mathbf{v}_k^{t_j}$ represents the observation noise, and $\emptyset$ represents an "empty element", indicating that the observation contained no information on the target or that target is unobservable when it is not within the observable region.

4.1.2 Fundamental processes

RBE forms a basis to the estimation of nonlinear non-Gaussian systems. Let a sequence of the states of the sensor platform s_i and a sequence of the observations by this sensor platform from time step 1 to time step k be $\tilde{\mathbf{x}}_{1:k}^{s_i} \equiv \{\tilde{\mathbf{x}}_l^{s_i}|\forall l \in \{1, ..., k\}\}$ and ${}^{s_i}\tilde{\mathbf{z}}_{1:k}^{t_j} \equiv \{{}^{s_i}\tilde{\mathbf{z}}_l^{t_j}|\forall l \in \{1, ..., k\}\}$, respectively. Notice here that $\tilde{(\cdot)}$ represents an instance

of variable $(\cdot)$. Given a prior belief of the target t_j in terms of probability density function as $p(\tilde{\mathbf{x}}_0^{t_j})$ and sequences of states and observations as $\tilde{\mathbf{x}}_{1:k}^{s_i}$ and ${}^{s_i}\tilde{\mathbf{z}}_{1:k}^{t_j}$, the RBE estimates the belief of the target at any time step k, $p(\mathbf{x}_k^{t_j}|{}^{s_i}\tilde{\mathbf{z}}_{1:k}^{t_j}, \tilde{\mathbf{x}}_{1:k}^{s_i})$, recursively through the two processes, prediction and correction.

4.1.2.1 Prediction

The prediction process computes the belief of the current state $p(\mathbf{x}_k^{t_j}|{}^{s_i}\tilde{\mathbf{z}}_{1:k-1}^{t_j}, \tilde{\mathbf{x}}_{1:k-1}^{s_i})$ from the belief in the previous time step $p(\mathbf{x}_{k-1}^{t_j}|{}^{s_i}\tilde{\mathbf{z}}_{1:k-1}^{t_j}, \tilde{\mathbf{x}}_{1:k-1}^{s_i})$. The prediction is carried out by Chapman-Kolmogorov equation and given by

$$p(\mathbf{x}_k^{t_j}|{}^{s_i}\tilde{\mathbf{z}}_{1:k-1}^{t_j}, \tilde{\mathbf{x}}_{1:k-1}^{s_i}) = \int_{\mathcal{X}^{t_j}} p(\mathbf{x}_k^{t_j}|\mathbf{x}_{k-1}^{t_j}) p(\mathbf{x}_{k-1}^{t_j}|{}^{s_i}\tilde{\mathbf{z}}_{1:k-1}^{t_j}, \tilde{\mathbf{x}}_{1:k-1}^{s_i}) d\mathbf{x}_{k-1}^{t_j}, \tag{4.4}$$

where $p(\mathbf{x}_k^{t_j}|\mathbf{x}_{k-1}^{t_j})$ is a probabilistic Markov motion model which maps the probability of transition from the previous state $\mathbf{x}_{k-1}^{t_j}$ to the current state $\mathbf{x}_k^{t_j}$. Notice that the update at $k = 1$ is carried out by letting $p(\mathbf{x}_{k-1}^{t_j}|{}^{s_i}\tilde{\mathbf{z}}_{1:k-1}^{t_j}, \tilde{\mathbf{x}}_{1:k-1}^{s_i}) = p(\tilde{\mathbf{x}}_0^{t_j})$. Equation (4.4) indicates that the performance of the prediction process relies on the target motion model $p(\mathbf{x}_k^{t_j}|\mathbf{x}_{k-1}^{t_j})$. Due to the fact that the target motion model is usually non-Gaussian when only prediction process applies to the RBE the belief could eventually become heavily non-Gaussian.

4.1.2.2 Correction

The correction process computes the belief $p(\mathbf{x}_k^{t_j}|{}^{s_i}\tilde{\mathbf{z}}_{1:k}^{t_j}, \tilde{\mathbf{x}}_{1:k}^{s_i})$ given the corresponding state estimated with the observations up to the previous time step $p(\mathbf{x}_k^{t_j}|{}^{s_i}\tilde{\mathbf{z}}_{1:k-1}^{t_j}, \tilde{\mathbf{x}}_{1:k-1}^{s_i})$ and a new observation ${}^{s_i}\tilde{\mathbf{z}}_k^{t_j}$. The equation is derived by applying formulas for marginal distribution and conditional independence and given by

$$p(\mathbf{x}_k^{t_j}|{}^{s_i}\tilde{\mathbf{z}}_{1:k}^{t_j}, \tilde{\mathbf{x}}_{1:k}^{s_i}) = \frac{l(\mathbf{x}_k^{t_j}|{}^{s_i}\tilde{\mathbf{z}}_k^{t_j}, \tilde{\mathbf{x}}_k^{s_i}) p(\mathbf{x}_k^{t_j}|{}^{s_i}\tilde{\mathbf{z}}_{1:k-1}^{t_j}, \tilde{\mathbf{x}}_{1:k-1}^{s_i})}{\int_{\mathcal{X}^{t_j}} l(\mathbf{x}_k^{t_j}|{}^{s_i}\tilde{\mathbf{z}}_k^{t_j}, \tilde{\mathbf{x}}_k^{s_i}) p(\mathbf{x}_k^{t_j}|{}^{s_i}\tilde{\mathbf{z}}_{1:k-1}^{t_j}, \tilde{\mathbf{x}}_{1:k-1}^{s_i}) d\mathbf{x}_k^{t_j}}, \tag{4.5}$$

where $l(\mathbf{x}_k^{t_j}|{}^{s_i}\tilde{\mathbf{z}}_k^{t_j}, \tilde{\mathbf{x}}_k^{s_i})$ represents the observation likelihood of $\mathbf{x}_k^{t_j}$ given ${}^{s_i}\tilde{\mathbf{z}}_k^{t_j}$ and $\tilde{\mathbf{x}}_k^{s_i}$. The observation likelihood is defined with reference to the probability of the detection and is given by

$$l(\mathbf{x}_k^{t_j}|{}^{s_i}\tilde{\mathbf{z}}_k^{t_j}, \tilde{\mathbf{x}}_k^{s_i}) = \begin{cases} p(\mathbf{x}_k^{t_j}|{}^{s}\tilde{\mathbf{z}}_k^{t_j}, \tilde{\mathbf{x}}_k^{s_i}) & \tilde{\mathbf{z}}_k^{t_j} \in {}^{s_i}\mathcal{X}_o^{t_j} \\ 1 - P_d(\mathbf{x}_k^{t_j}|\tilde{\mathbf{x}}_k^{s_i}) & \tilde{\mathbf{z}}_k^{t_j} \notin {}^{s_i}\mathcal{X}_o^{t_j} \end{cases} \tag{4.6}$$

where $p(\mathbf{x}_k^{t_j}|{}^s\tilde{\mathbf{z}}_k^{t_j}, \tilde{\mathbf{x}}_k^{s_i})$ is the probabilistic representation of the sensor model defined in Equation (4.3). When the target is within the observable region a positive observation is obtained and the observation likelihood is a probability density function given the current observation. When the target is out of the observable region the negative observation is defined with respect to the probability of detection as the observation likelihood. Due to the fact that the observation likelihood of the negative observation is non-Gaussian, when the negative observation occurs in the RBE the object belief would immediately become heavily non-Gaussian.

The prediction and correction processes, as described in Equations (4.4) and (4.5), essentially require the evaluation of a function at an arbitrary point in the target space $\mathcal{X}^{t_j}$, $\mathbf{f}(\mathbf{x}^{t_j})$, and the integration of a function over the target space, $I = \int_{\mathcal{X}^{t_j}} \mathbf{f}(\mathbf{x}^{t_j}) d\mathbf{x}^{t_j}$, in their numerical implementation.

4.2 Grid-based RBE

4.2.1 Representation of target space and belief

Consider that the ith sensor platform or autonomous vehicle, s_i, observes the jth target, t_j. The grid-based RBE achieves non-Gaussian belief estimation by first representing the arbitrary target space $\mathcal{X}^{t_j}$ in terms of a set of grid cells by constructing a rectangular space $\mathcal{X}^{r_j}$ that covers the target space. For simplicity let us consider a two-dimensional target space and it is represented as $\mathbf{m}^{t_j} = [x^{t_j}, y^{t_j}] \in \mathcal{X}^{t_j}$. The creation of a rectangular space $\mathcal{X}^{r_j}$ is achieved then by defining the minimum and maximum values of the target space

$$x_{\min}^{t_j} = \min\{x^{t_j}\}, x_{\max}^{t_j} = \max\{x^{t_j}\}$$
$$y_{\min}^{t_j} = \min\{y^{t_j}\}, y_{\max}^{t_j} = \max\{y^{t_j}\}$$

and subsequently creating a rectangular space as $\mathcal{X}^{r_j} = \{\mathbf{m} | \forall x \in [x_{\min}^{t_j}, x_{\max}^{t_j}], \forall y \in [y_{\min}^{t_j}, y_{\max}^{t_j}]\} \supseteq \mathcal{X}^{t_j}$ where $\mathbf{m} = [x, y]$. The grid space is further introduced by discretizing the rectangular space by n_x and n_y grid cells in two directions, respectively. The dimensions of a grid cell are defined as $\Delta x^{r_j} = (x_{\max}^{t_j} - x_{\min}^{t_j})/n_x$ and $\Delta y^{r_j} = (y_{\max}^{t_j} - y_{\min}^{t_j})/n_y$. This results in introducing the center of each grid cell as

$$\bar{\mathbf{m}}_{i_x,i_y}^{r_j} = [\bar{x}_{i_x}^{r_j}, \bar{y}_{i_y}^{r_j}] = [(i_x - 0.5)\Delta x^{r_j} + x_{\min}^{t_j}, (i_y - 0.5)\Delta y^{r_j} + y_{\min}^{t_j}], \tag{4.7}$$

where $\forall i_x \in \{1, ..., n_x\}$ and $\forall i_y \in \{1, ..., n_y\}$. Each grid cell is defined as

$$\mathcal{X}_{i_x,i_y}^{r_j} = \{\mathbf{m} || x - \bar{x}_{i_x}^{r_j}| < \frac{1}{2}\Delta x^{r_j}, |y - \bar{y}_{i_y}^{r_j}| < \frac{1}{2}\Delta y^{r_j}\}. \tag{4.8}$$

Note that $\bigcup_{i_x=1}^{n_x}\bigcup_{i_y=1}^{n_y}\mathcal{X}_{i_x,i_y}^{r_j} = \mathcal{X}^{r_j}$ and $\bigcap_{i_x=1}^{n_x}\bigcap_{i_y=1}^{n_y}\mathcal{X}_{i_x,i_y}^{r_j} = \varnothing$. Finally, the selection of grid cells that represent the target space is performed by selecting a grid cell when its center is located in the target space, $\mathcal{X}_{i_x,i_y}^{r_j} \subset \mathcal{X}^{t_j}$ if $\bar{\mathbf{x}}_{i_x,i_y}^{r_j} \in \mathcal{X}^{t_j}$. The approximate target space derived by the processes described above is $\mathcal{X}^{t_j} \approx \{\mathcal{X}_1^{r_j}, \mathcal{X}_2^{r_j}, ..., \mathcal{X}_{n_g}^{r_j}\}$, where n_g is the number of grid cells approximating the target space.

The belief is represented by a probability density function over the target space. Since the target space of arbitrary shape with n_g grid cells can always be covered by a rectangular space of grid cells $n_x \times n_y$ ($n_g \leq n_x n_y$), the position of each grid cell of the target space can be described in a two-dimensional integer space as $[i_x, i_y]$ where $i_x \in \{1, ..., n_x\}$ and $i_y \in \{1, ..., n_y\}$. With the integer representation, let the belief at the grid cell $[i_x, i_y]$ be $p^{i_x,i_y}(\cdot)$. The prediction and the correction processes of the grid-based RBE are formulated as follows:

4.2.2 Prediction

The prediction process of grid-based RBE requires the numerical evaluation of Equation (4.4). Given the belief of the previous state $p^{i_x,i_y}(\mathbf{x}_{k-1}^{t_j}|^{s_i}\tilde{\mathbf{z}}_{1:k-1}^{t_j}, \tilde{\mathbf{x}}_{1:k-1}^{s_i})$ as well as the Malkov motion model $p^{i_x,i_y}(\mathbf{x}_k^{t_j}|\mathbf{x}_{k-1}^{t_j})$ constructed in the matrix of size $I_x \times I_y$ as the convolution kernel, the belief of the current state can be numerically predicted as

$$p^{i_x,i_y}(\mathbf{x}_k^{t_j}|^{s_i}\tilde{\mathbf{z}}_{1:k-1}^{t_j}, \tilde{\mathbf{x}}_{1:k-1}^{s_i}) = p^{i_x,i_y}(\mathbf{x}_{k-1}^{t_j}|^{s_i}\tilde{\mathbf{z}}_{1:k-1}^{t_j}, \tilde{\mathbf{x}}_{1:k-1}^{s_i}) \otimes p^{I_x,I_y}(\mathbf{x}_k^{t_j}|\mathbf{x}_{k-1}^{t_j}), \tag{4.9}$$

where $\otimes$ indicates the two-dimensional convolution of the belief of the previous state with the Markov motion model. Therefore, the belief of the current state is given by

$$\begin{aligned} &p^{i_x,i_y}(\mathbf{x}_k^{t_j}|^{s_i}\tilde{\mathbf{z}}_{1:k-1}^{t_j}, \tilde{\mathbf{x}}_{1:k-1}^{s_i}) \\ &\quad = \sum_{\beta=1}^{I_y}\sum_{\alpha=1}^{I_x} p^{\alpha,\beta}(\mathbf{x}_k^{t_j}|\mathbf{x}_{k-1}^{t_j}) p^{i_x-\alpha+1,i_y-\beta+1}(\mathbf{x}_{k-1}^{t_j}|^{s_i}\tilde{\mathbf{z}}_{1:k-1}^{t_j}, \tilde{\mathbf{x}}_{1:k-1}^{s_i}). \end{aligned} \tag{4.10}$$

4.2.3 Correction

The correction process of grid-based RBE corresponds to the numerical computation of Equation (4.5). Given the predicted belief $p(\mathbf{x}_k^{t_j}|^{s_i}\tilde{\mathbf{z}}_{1:k-1}^{t_j}, \tilde{\mathbf{x}}_{1:k-1}^{s_i})$ and the new observation likelihood $l(\mathbf{x}_k^{t_j}|^{s_i}\tilde{\mathbf{z}}_k^{t_j}, \tilde{\mathbf{x}}_k^{s_i})$, the belief at each grid cell $[i_x, i_y]$ is updated as

$$p^{i_x,i_y}(\mathbf{x}_k^{t_j}|^{s_i}\tilde{\mathbf{z}}_{1:k}^{t_j}, \tilde{\mathbf{x}}_{1:k}^{s_i}) = \frac{q^{i_x,i_y}(\mathbf{x}_k^{t_j}|^{s_i}\tilde{\mathbf{z}}_{1:k}^{t_j}, \tilde{\mathbf{x}}_{1:k}^{s_i})}{A_c \sum_{\alpha=1}^{n_x}\sum_{\beta=1}^{n_y} q^{\alpha,\beta}(\mathbf{x}_k^{t_j}|^{s_i}\tilde{\mathbf{z}}_{1:k}^{t_j}, \tilde{\mathbf{x}}_{1:k}^{s_i})}, \tag{4.11}$$

where A_c is the area of a grid cell and

$$q^{i_x,i_y}(\mathbf{x}_k^{t_j}|^{s_i}\tilde{\mathbf{z}}_{1:k}^{t_j}, \tilde{\mathbf{x}}_{1:k}^{s_i}) = l^{i_x,i_y}(\mathbf{x}_k^{t_j}|^{s_i}\tilde{\mathbf{z}}_k^{t_j}, \tilde{\mathbf{x}}_k^{s_i}) p^{i_x,i_y}(\mathbf{x}_k^{t_j}|^{s_i}\tilde{\mathbf{z}}_{1:k-1}^{t_j}, \tilde{\mathbf{x}}_{1:k-1}^{s_i}). \tag{4.12}$$

4.3 Observation fusion

Figure 4.1 shows the schematic diagram of the observation fusion technique for the grid-based RBE where the internal process of the sensor platform or autonomous vehicle s_i is particularly shown. It is noted that the diagram is completed for the centralized estimation where the process of the leader sensor platform is indicated by the red dotted block simply because the process of the decentralized estimation is more complicated and needs unimportant explanations. After moving and sensing as shown in the upper-right block, the sensor platform creates an observation likelihood and corrects the current belief. In the leader sensor platform, the likelihood is a fused observation likelihood, which is created from not only its own observation likelihood but also the observation likelihoods from other sensor platforms. The fused observation likelihood combined at the leader sensor platform is given by

$$l(\mathbf{x}_k^{t_j}|^{s}\tilde{\mathbf{z}}_k^{t_j}, \tilde{\mathbf{x}}_k^{s}) = \prod_{1\le i\le n_s} l(\mathbf{x}_k^{t_j}|^{s_i}\tilde{\mathbf{z}}_k^{t_j}, \tilde{\mathbf{x}}_k^{s_i}), \tag{4.13}$$

where ${}^{s}\tilde{\mathbf{z}}_k^{t_j} = \{{}^{s_1}\tilde{\mathbf{z}}_k^{t_j}, {}^{s_2}\tilde{\mathbf{z}}_k^{t_j}, \ldots, {}^{s_{n_s}}\tilde{\mathbf{z}}_k^{t_j}, \}$ and $\tilde{\mathbf{x}}_k^{s} = \{\tilde{\mathbf{x}}_k^{s_1}, \tilde{\mathbf{x}}_k^{s_2}, \ldots, \tilde{\mathbf{x}}_k^{s_{n_s}}\}$. The grid-based RBE then predicts the corrected belief with the target motion model and recursively updates and maintains the belief through the correction and prediction processes. The belief is synchronized by sending that of the leader sensor platform after a certain period of time since the beliefs of the non-leader sensor platforms are maintained based on their own observations and thus become different as time passes.

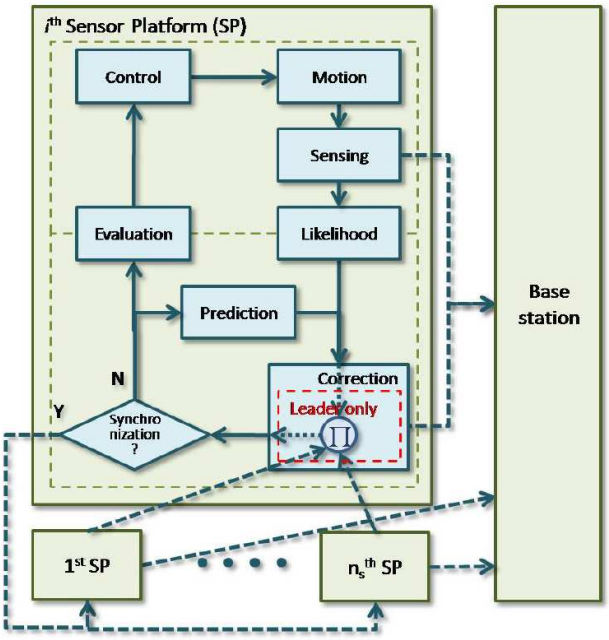


Figure 4.1: Observation fusion technique for grid-based RBE

The observation fusion technique for the grid-based RBE has its strength in need for communicating only observation likelihoods, which do not contain correlated information and thus could be smaller in terms of the data size [(40), (70)]. However, the collection of observation likelihoods from other sensor platforms clearly slows down the grid-based RBE of the leader sensor platform, thereby making the estimated belief more unreliable. The speed of the grid-based RBE could be improved by performing the observation fusion less frequently. Since the correction only occurs in the observation fusion, the reduction of observation fusion however results in the loss of information from the other sensor platforms and thus the unreliability of the estimated belief. Moreover, the information from the other sensor platforms is strictly limited to observations. Even if a sensor platform has found a more accurate motion model of the target, the belief of the leader sensor platform cannot be improved.

4.4 Summary

The motion model and the sensor model of a system was described in this Chapter, following by the formulations of the prediction and the correction process, two fundamental processes of the RBE technique. The formulations of the grid-based RBE technique have been then derived by discretizing the target space and numerically evaluating the formulations of the RBE technique. Lastly, the observation fusion technique for the grid-based RBE was described for the cooperative estimation and the corresponding formulations were presented.

Chapter 5

Part 1: Parallel Grid-based RBE and Belief Fusion

This Chapter presents the novel parallel grid-based RBE and the belief fusion techniques for the target estimation. The proposed parallel grid-based RBE technique identifies the parallel computation in the prediction and the correction processes and implemented into the GPU to accelerate the conventional grid-based RBE technique. The belief fusion technique for cooperative estimation is presented and it fuses the belief instead of the observation likelihood, in conventional observation fusion technique, to achieve accurate estimation. Since the fused belief contains not only the observation information but also the target motion information one does not need to perform belief fusion frequently so as to reduce the communication load and further benefit for the cooperative estimation. The DTFLOP modeling is validated by the proposed parallel grid-based RBE technique through a series of parametric studies in the end.

This Chapter is organized as follows. Section 5.1 firstly presents the novel parallel grid-based RBE technique and the formulations of the prediction and the correction processes are derived, respectively. The novel belief fusion technique is then presented in Section 5.2 and the comparison with the conventional observation fusion technique is discussed. In addition, Section 5.3 validates the DTFLOP modeling by the proposed parallel grid-based RBE technique. Finally, a series of numerical examples are presented in Section 5.4 and the advantages of the proposed parallel grid-based RBE and the belief fusion techniques are shown.

5.1 Parallel grid-based RBE

5.1.1 Prediction

The parallel implementation of the prediction process of the grid-based RBE technique is straightforward. Since the prediction at each node, given by Equation (4.10), is performed independently, the prediction process is able to achieve a parallel efficiency of 100% in an ideal environment. However, this equation also shows that the computational time for the prediction process is largely dominated by the size of the convolution kernel, which represents the target motion model. In order for the best performance, it is important that an appropriate size of convolution kernel, which needs to be big enough to capture the motion of the target but small enough to perform fast computation, is chosen.

Since the RBE designed with high frequency results in using the target motion model well approximated by a Gaussian probability density, the prediction process of the grid-based RBE technique can be reformulated with the Gaussian assumption as a pre-process and accelerated to achieve the maximum performance. With the Gaussian assumption, the convolution kernel in the matrix of the size $I_x \times I_y$ can be separated into two vector kernels in the name of the separable convolution, a column kernel of length I_x and a row kernel of length I_y. Therefore, the matrix for the motion model of target t_j is given by

$$p^{I_x,I_y}(\mathbf{x}_k^{t_j}|\mathbf{x}_{k-1}^{t_j}) = {}^{c}p^{I_x}(\mathbf{x}_k^{t_j}|\mathbf{x}_{k-1}^{t_j}){}^{r}p^{I_y}(\mathbf{x}_k^{t_j}|\mathbf{x}_{k-1}^{t_j}), \tag{5.1}$$

where ${}^{c}p^{I_x}(\mathbf{x}_k^{t_j}|\mathbf{x}_{k-1}^{t_j})$ and ${}^{r}p^{I_y}(\mathbf{x}_k^{t_j}|\mathbf{x}_{k-1}^{t_j})$ are the column kernel and the row kernel, respectively. At the same time, the size of convolution kernel is reduced from $I_x \times I_y$ to $I_x + I_y$. Substituting Equation (5.1) into Equation (4.9), the belief of the current state can be predicted as

$$\begin{aligned} &p^{i_x,i_y}(\mathbf{x}_k^{t_j}|{}^{s_i}\tilde{\mathbf{z}}_{1:k-1}, \tilde{\mathbf{x}}_{1:k-1}^{s_i}) \\ &\quad = \left[p^{i_x,i_y}(\mathbf{x}_{k-1}^{t_j}|{}^{s_i}\tilde{\mathbf{z}}_{1:k-1}^{t_j}, \tilde{\mathbf{x}}_{1:k-1}^{s_i}) \otimes {}^{c}p^{I_x}(\mathbf{x}_k^{t_j}|\mathbf{x}_{k-1}^{t_j})\right] \otimes {}^{r}p^{I_y}(\mathbf{x}_k^{t_j}|\mathbf{x}_{k-1}^{t_j}), \end{aligned} \tag{5.2}$$

which means that the prediction process of the grid-based RBE technique is broken down into two steps:

$$u^{i_x,i_y}(\mathbf{x}_k^{t_j}|{}^{s_i}\tilde{\mathbf{z}}_{1:k-1}^{t_j}, \tilde{\mathbf{x}}_{1:k-1}^{s_i})$$

$$
\begin{aligned}
&= p^{i_x,i_y}(\mathbf{x}_{k-1}^{t_j}|^{s_i}\tilde{\mathbf{z}}_{1:k-1}^{t_j}, \tilde{\mathbf{x}}_{1:k-1}^{s_i}) \otimes {}^{c}p^{I_x}(\mathbf{x}_k^{t_j}|\mathbf{x}_{k-1}^{t_j}) \\
&= \sum_{\alpha=1}^{I_x} {}^{c}p^{\alpha}(\mathbf{x}_k^{t_j}|\mathbf{x}_{k-1}^{t_j}) p^{i_x-\alpha+1,i_y}(\mathbf{x}_{k-1}^{t_j}|^{s_i}\tilde{\mathbf{z}}_{1:k-1}^{t_j}, \tilde{\mathbf{x}}_{1:k-1}^{s_i});
\end{aligned}
\tag{5.3}
$$

and

$$
\begin{aligned}
&p^{i_x,i_y}(\mathbf{x}_k^{t_j}|^{s_i}\tilde{\mathbf{z}}_{1:k-1}^{t_j}, \tilde{\mathbf{x}}_{1:k-1}^{s_i}) \\
&\quad = u^{i_x,i_y}(\mathbf{x}_k^{t_j}|^{s_i}\tilde{\mathbf{z}}_{1:k-1}^{t_j}, \tilde{\mathbf{x}}_{1:k-1}^{s_i}) \otimes {}^{r}p^{I_y}(\mathbf{x}_k^{t_j}|\mathbf{x}_{k-1}^{t_j}) \\
&\quad = \sum_{\beta=1}^{I_y} {}^{r}p^{\beta}(\mathbf{x}_k^{t_j}|\mathbf{x}_{k-1}^{t_j}) u^{i_x,i_y-\beta+1}(\mathbf{x}_{k-1}^{t_j}|^{s_i}\tilde{\mathbf{z}}_{1:k-1}^{t_j}, \tilde{\mathbf{x}}_{1:k-1}^{s_i}).
\end{aligned}
\tag{5.4}
$$

These equations show that the prediction process at each grid cell is carried out by performing two one-dimensional convolutions each in horizontal and vertical direction instead of the original one two-dimensional convolution while remaining completely parallelization. For the first equation, the number of floating point operations for each grid cell is seen $2I_x$ since I_x times of one multiplication and one summation are necessary, whereas the number of floating point operations for the second one is $2I_y$ via the similar observation. Having a total of n_g grid cells, the total number of floating point operations for the prediction process is thus given by

$$
N_p = 2n_g \left(I_x + I_y\right). \tag{5.5}
$$

This is considerably small compared to that of the original formulation which is derived as $2n_g I_x I_y$ via Equation (4.10) since $I_x + I_y \ll I_x I_y$ for an appropriate prediction process.

5.1.2 Correction

The parallelization of the correction process of the grid-based RBE technique requires the breakdown of Equation (4.11) and identification of the parallelizable sub-processes. The correction process is given by

$$
p^{i_x,i_y}(\mathbf{x}_k^{t_j}|^{s_i}\tilde{\mathbf{z}}_{1:k}^{t_j}, \tilde{\mathbf{x}}_{1:k}^{s_i}) = \frac{q^{i_x,i_y}(\mathbf{x}_k^{t_j}|^{s_i}\tilde{\mathbf{z}}_{1:k}^{t_j}, \tilde{\mathbf{x}}_{1:k}^{s_i})}{A_c \sum_{\alpha=1}^{n_x}\sum_{\beta=1}^{n_y} q^{\alpha,\beta}(\mathbf{x}_k^{t_j}|^{s_i}\tilde{\mathbf{z}}_{1:k}^{t_j}, \tilde{\mathbf{x}}_{1:k}^{s_i})}, \tag{5.6}
$$

where A_c is the area of a grid cell and

$$q^{i_x,i_y}(\mathbf{x}_k^{t_j}|^{s_i}\tilde{\mathbf{z}}_{1:k}^{t_j},\tilde{\mathbf{x}}_{1:k}^{s_i}) = l^{i_x,i_y}(\mathbf{x}_k^{t_j}|^{s_i}\tilde{\mathbf{z}}_k^{t_j},\tilde{\mathbf{x}}_k^{s_i})p^{i_x,i_y}(\mathbf{x}_k^{t_j}|^{s_i}\tilde{\mathbf{z}}_{1:k-1}^{t_j},\tilde{\mathbf{x}}_{1:k-1}^{s_i}). \tag{5.7}$$

By observing the mathematical operations, the correction process can be broken down into following three steps:

1. Calculate $q^{i_x,i_y}(\mathbf{x}_k^{t_j}|^{s_i}\tilde{\mathbf{z}}_{1:k}^{t_j},\tilde{\mathbf{x}}_{1:k}^{s_i})$ by multiplying the predicted belief $p^{i_x,i_y}(\mathbf{x}_k^{t_j}|^{s_i}\tilde{\mathbf{z}}_{1:k-1}^{t_j},\tilde{\mathbf{x}}_{1:k}^{s_i})$ by the observation likelihood $l^{i_x,i_y}(\mathbf{x}_k^{t_j}|^{s_i}\tilde{\mathbf{z}}_k^{t_j},\tilde{\mathbf{x}}_k^{s_i})$;
2. Sum $\sum_{\alpha=1}^{n_x}\sum_{\beta=1}^{n_y} q^{\alpha,\beta}\left(\mathbf{x}_k^{t_j}|^{s_i}\tilde{\mathbf{z}}_{1:k}^{t_j},\tilde{\mathbf{x}}_{1:k}^{s_i}\right)$ and multiply the sum by A_c;
3. Calculate $p^{i_x,i_y}(\mathbf{x}_k^{t_j}|^{s_i}\tilde{\mathbf{z}}_{1:k}^{t_j},\tilde{\mathbf{x}}_{1:k}^{s_i})$ by dividing $q^{i_x,i_y}(\mathbf{x}_k^{t_j}|^{s_i}\tilde{\mathbf{z}}_{1:k}^{t_j},\tilde{\mathbf{x}}_{1:k}^{s_i})$ by $A_c\sum_{\alpha=1}^{n_x}\sum_{\beta=1}^{n_y} q^{\alpha,\beta}(\mathbf{x}_k^{t_j}|^{s_i}\tilde{\mathbf{z}}_{1:k}^{t_j},\tilde{\mathbf{x}}_{1:k}^{s_i})$.

The breakdown indicates that Steps 1 and 3 are the grid-wise processes, which can be conducted completely in parallel whereas Step 2 cannot be performed in parallel.

5.2 Belief fusion

Figure 5.1 shows the schematic diagram of the belief fusion technique for the cooperative target estimation. The difference of the proposed belief fusion technique from the conventional observation fusion technique can be found in the location of the communication of the leader sensor platform. While the leader sensor platform in the conventional observation fusion technique communicates with other sensor platforms within the correction process, the proposed belief fusion technique has the communication outside the grid-based RBE process. As a result, the data to receive and fuse are not the observation likelihoods but the beliefs. This change overcomes the problems addressed in the conventional observation fusion technique. Without having the communication inside the grid-based RBE process, the speed and the accuracy of the grid-based RBE technique are kept high. In addition, the communication of the beliefs rather than the observations magnifies the reliability of the belief by reflecting the complete information on the past observations and target motions rather than only the observations.

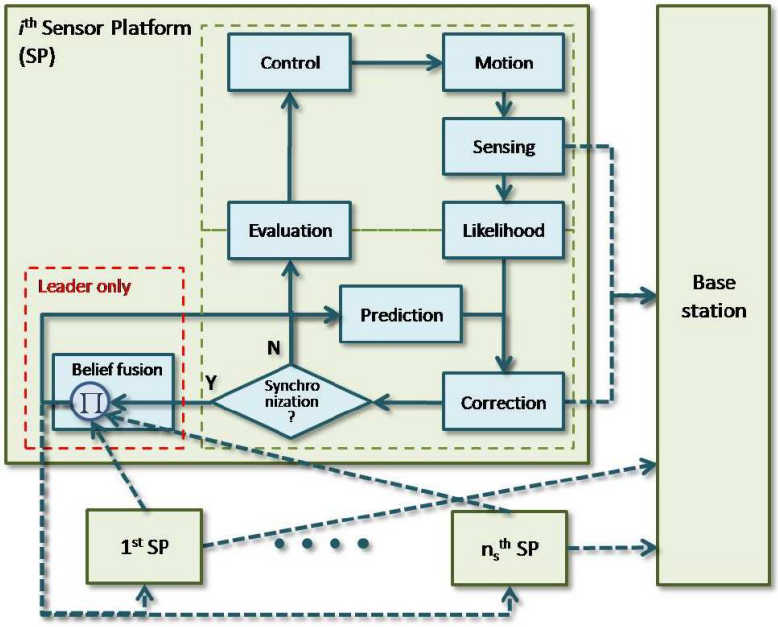


Figure 5.1: Belief fusion technique for grid-based RBE

The formulation of the belief fusion is given by

$$p(\mathbf{x}_k^{t_j}|^s\tilde{\mathbf{z}}_k^{t_j}, \tilde{\mathbf{x}}_k^s) = \frac{q_s(\mathbf{x}_k^{t_j}|^s\tilde{\mathbf{z}}_{1:k}^{t_j}, \tilde{\mathbf{x}}_{1:k}^s)}{\int_{\mathcal{X}^{t_j}} q_s(\mathbf{x}_k^{t_j}|^s\tilde{\mathbf{z}}_{1:k}^{t_j}, \tilde{\mathbf{x}}_{1:k}^s) d\mathbf{x}_k^{t_j}} \tag{5.8}$$

where $q_s(\mathbf{x}_k^{t_j}|^s\tilde{\mathbf{z}}_{1:k}^{t_j}, \tilde{\mathbf{x}}_{1:k}^s)$ is given by

$$q_s(\mathbf{x}_k^{t_j}|^s\tilde{\mathbf{z}}_{1:k}^{t_j}, \tilde{\mathbf{x}}_{1:k}^s) = \prod_{1\leq i\leq n_s} p(\mathbf{x}_k^{t_j}|^{s_i}\tilde{\mathbf{z}}_k^{t_j}, \tilde{\mathbf{x}}_k^{s_i}). \tag{5.9}$$

5.3 Validation of DTFLOP modeling

5.3.1 GPU implementation

Figure 5.2 shows the schematic diagram of the GPU implementation for the proposed parallel grid-based RBE technique. For the computational efficiency, the GPU stores the entire data in the global memory and performs the parallel grid-based RBE technique using local memories. As a result, the data transmission between the CPU's memory and the GPU's local memories are carried out via the GPU's global memory,

and all the parallelizable floating point operations are executed using the local memories. For the prediction process, the data to be transmitted from the CPU to the GPU's local memories are the previous belief $p(\mathbf{x}_{k-1}^{t_j}|^{s_i}\tilde{\mathbf{z}}_{1:k-1}^{t_j}, \tilde{\mathbf{x}}_{1:k-1}^{s_i})$ and the target motion model $p(\mathbf{x}_k^{t_j}|\mathbf{x}_{k-1}^{t_j})$. Since the predicted belief is in the local memories, the correction needs only the observation likelihood to be transmitted in addition. After performing the multiplication of $p(\mathbf{x}_k^{t_j}|^{s_i}\tilde{\mathbf{z}}_{1:k-1}^{t_j}, \tilde{\mathbf{x}}_{1:k-1}^{s_i})$ and observation likelihood $l(\mathbf{x}_k^{t_j}|^{s_i}\tilde{\mathbf{z}}_k^{t_j}, \tilde{\mathbf{x}}_k^{s_i})$ using GPU's local memories, the result $q(\mathbf{x}_k^{t_j}|^{s_i}\tilde{\mathbf{z}}_{1:k}^{t_j}, \tilde{\mathbf{x}}_{1:k}^{s_i})$ is transmitted to the CPU's memory to calculate the sum $A_c \sum_{\alpha=1}^{n_x} \sum_{\beta=1}^{n_y} q^{\alpha,\beta}(\mathbf{x}_k^{t_j}|^{s_i}\tilde{\mathbf{z}}_{1:k}^{t_j}, \tilde{\mathbf{x}}_{1:k}^{s_i})$. The sum is then transmitted back to the GPU's local memories to perform the parallel divisions and then update the belief to be $p(\mathbf{x}_k^{t_j}|^{s_i}\tilde{\mathbf{z}}_{1:k}^{t_j}, \tilde{\mathbf{x}}_{1:k}^{s_i})$. Finally, the belief is transmitted back to the CPU's memory for the next iteration of the parallel grid-based RBE technique.

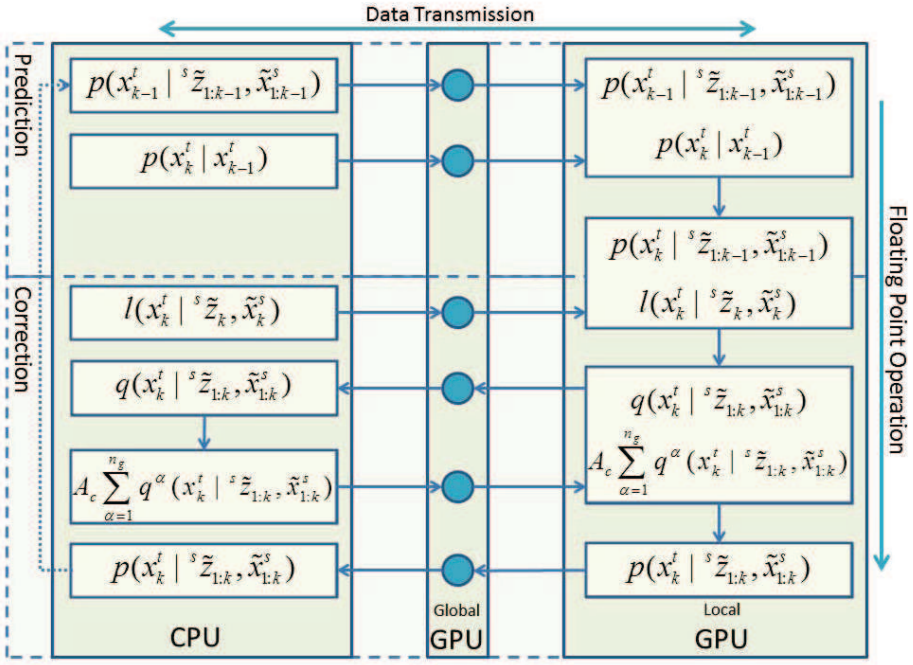


Figure 5.2: GPU implementation of parallel grid-based RBE technique

5.3.2 Data transmission

Regard to the parallel grid-based RBE technique, the number of the data of the belief and the target motion model for the prediction process are n_g and $I_x + I_y$, respectively. The same number of data, n_g and $I_x + I_y$, are transmitted to the GPU's local memory

to perform the parallel prediction process. In the correction process, the number of the data of the observation likelihood to be transmitted from the CPU's memory to the GPU's local memory through the GPU's global memory is n_g whereas the number of the data of the result $q(\mathbf{x}_k^{t_j}|^{s_i}\tilde{\mathbf{z}}_{1:k}^{t_j}, \tilde{\mathbf{x}}_{1:k}^{s_i})$ to be transmitted from the GPU's local memory to the CPU's memory through the GPU's global memory is similarly n_g. The number of the data of the sum, $A_c \sum_{\alpha=1}^{n_x} \sum_{\beta=1}^{n_y} q^{\alpha,\beta}(\mathbf{x}_k^{t_j}|^{s_i}\tilde{\mathbf{z}}_{1:k}^{t_j}, \tilde{\mathbf{x}}_{1:k}^{s_i})$, to be then transmitted to the GPU's local memory to perform the parallel divisions is 1, and finally the number of the data to be transmitted back to the CPU's memory for the next iteration of the parallel grid-based RBE technique is n_g.

By observing the data transmission processes, the total number of the data transmitted from the CPU's memory to the GPU's global memory is given by

$$\begin{aligned} N_{\mathrm{CG}} = &\quad (n_g + I_x + I_y) + (1 + n_g) \\ = &\quad 2n_g + I_x + I_y + 1, \end{aligned} \tag{5.10}$$

and all the data are transmitted continuously from the GPU's global memory to the GPU's local memory:

$$N_{\mathrm{GL}} = N_{\mathrm{CG}} = 2n_g + I_x + I_y + 1. \tag{5.11}$$

The total number of the data transmitted from the GPU's local memory to the GPU's global memory is

$$N_{\mathrm{LG}} = n_g + n_g = 2n_g, \tag{5.12}$$

and that from the GPU's global memory to the CPU's memory similarly becomes

$$N_{\mathrm{GC}} = N_{\mathrm{LG}} = 2n_g. \tag{5.13}$$

Since the copy bandwidth from the GPU's global memory to the GPU's local memory and the one in the opposite direction are the same, the number of the data transmitted inside the GPU is given by

$$N_{\mathrm{GG}} = N_{\mathrm{GL}} + N_{\mathrm{LG}} = 4n_g + I_x + I_y + 1. \tag{5.14}$$

5.3.3 Floating point operations

The number of floating point operations performed on the GPU for the prediction process of the parallel grid-based RBE technique is $2n_g(I_x + I_y)$ as Equation (5.5) indicated. The number of floating point operations performed on the GPU for the correction process is identified as $2n_g$ in total since n_g parallel multiplications and n_g parallel divisions are performed for Steps 1 and 3 respectively, whereas the number of floating point operations performed on the CPU is n_g by n_g summations for Step 2. As a consequence, the total number of floating point operations performed on the CPU and the GPU for one iteration of the parallel grid-based RBE technique are respectively given by

$$N_{\mathrm{C}} = n_g, \tag{5.15}$$

$$N_{\mathrm{G}} = 2n_g(I_x + I_y) + 2n_g = 2n_g(I_x + I_y + 1). \tag{5.16}$$

5.3.4 Estimated time cost

The total time cost of the data transmission is consist of the time cost from the CPU's memory to the GPU's global memory, from the GPU's global memory to the CPU's memory and between GPU's global memory and GPU's local memory. By substituting Equation (5.10), (5.13) and (5.14) into Equation (3.5), (3.6) and (3.7) respectively, the time cost of the data transmission for each component are respectively given by

$$\Delta t_{\mathrm{CG}} = P\frac{2n_g + I_x + I_y + 1}{B_{\mathrm{CG}}}, \tag{5.17}$$

$$\Delta t_{\mathrm{GC}} = P\frac{2n_g}{B_{\mathrm{GC}}}, \tag{5.18}$$

$$\Delta t_{\mathrm{GG}} = P\frac{4n_g + I_x + I_y + 1}{B_{\mathrm{GG}}}. \tag{5.19}$$

If the same numerical representations P are selected in the three types of the data transmission and the substitution of Equation (5.17), (5.18) and (5.19) into Equation (3.4), the total time cost of the data transmission is given by

$$\Delta t_{\mathrm{trans}} = P\left(\frac{2n_g + I_x + I_y + 1}{B_{\mathrm{CG}}} + \frac{2n_g}{B_{\mathrm{GC}}} + \frac{4n_g + I_x + I_y + 1}{B_{\mathrm{GG}}}\right) \tag{5.20}$$

Since the number of floating point operations performed on the CPU and the GPU are known from the previous section the computational time cost can be determined.

By substituting Equation (5.15) into Equation (3.8), the time cost of the sequential computation performed on the CPU is given by

$$\Delta t_{\mathrm{C}} = \frac{n_g}{V_{\mathrm{C}}}. \tag{5.21}$$

Similarly, substituting Equation (5.16) into Equation (3.9) the time cost of the parallel computation performed on the GPU is given by

$$\Delta t_{\mathrm{G}} = 2n_g \frac{I_x + I_y + 1}{V_{\mathrm{G}}}. \tag{5.22}$$

5.3.5 Validation

This subsection shows the validation of the proposed DTFLOP modeling using the proposed parallel grid-based RBE technique in different computer hardware setups. Table 5.1 shows the setup specifications which have been available for the validation and the other investigations. Among three setups, Setup 1 is the fastest in both CPU and GPU whereas Setup 3 is the slowest.

Table 5.1: Test computer system specifications I

Setup	Processor	Memory	GPU
1	Intel Dual-Core,2.70GHz	4.0GB	Nvidia GF GT220
2	Intel Dual-Core,2.40GHz	4.0GB	Nvidia GF GT320M
3	Intel Dual-Core,2.40GHz	4.0GB	Nvidia GF GS8400

This set of tests is aimed to validate the proposed DTFLOP modeling by estimating the total time cost Δt of one iteration of the parallel grid-based RBE technique and comparing it with the actual time cost experimentally measured in three different computer setups. Each component, $\Delta t_{\mathrm{trans}}$, Δt_{G} or Δt_{C}, is also compared with the actual time cost respectively. Needless to say, the convolution kernel size $I_x + I_y$ and grid space size n_g are the two major factors which affect the speed of the proposed parallel grid-based RBE. Two tests are thus conducted by each varying the convolution kernel size and the grid space size.

5.3.5.1 Test 1

Test 1 is performed by fixing the grid space size of the proposed parallel grid-based RBE technique to 1000×1000 and varying convolution kernel size $I_x = I_y = i$ from 1

to 200. The convolution kernel size over 200 is not explored since it is unlikely that the target motion model requires such a large convolution kernel. The square convolution kernel is because of the insignificance in changing size in both x and y directions, and this additionally allows visualization of results in two dimensional space.

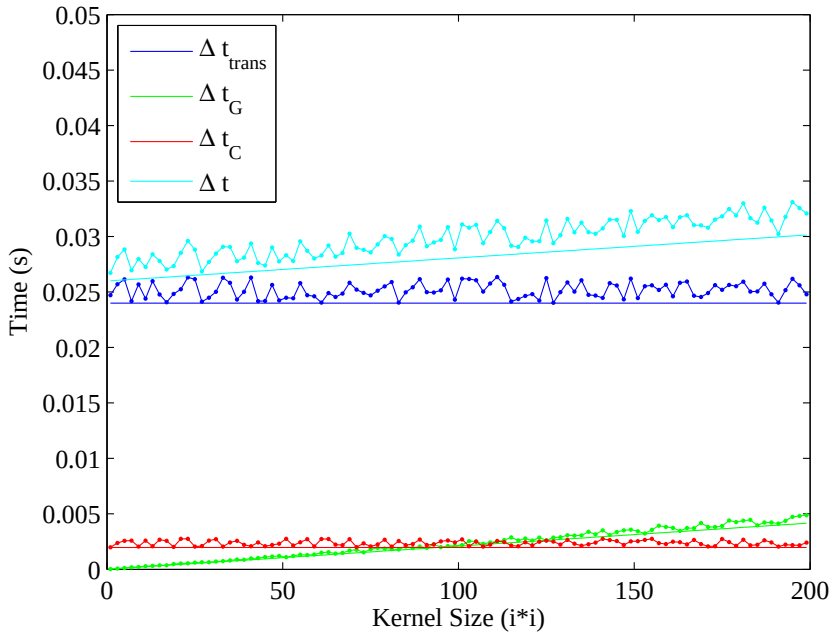


Figure 5.3: Time cost of all components for Setup1 with fixed grid space

The results of all the components of the time cost for the three computer setups are shown in Figure 5.3, 5.4 and 5.5, respectively. Each solid line represents the estimated total and component time costs whereas each solid dot line with the same color represents the corresponding actual time cost. These figures primarily show that the total and component time costs estimated by the proposed DTFLOP modeling well match to the actual time cost. Values listed in Table 5.2 also support this and indicate the effectiveness of the proposed DTFLOP modeling since the average and the maximum relative errors are below 7% and 12% respectively. While the time cost of the data transmission is seen to contribute the most, it is also seen that the time cost on GPU increases the total time cost with the increase in convolution kernel size particularly when the GPU is of low quality. It is thus important to use a high performance GPU if fast RBE with large convolution kernel size is necessary.

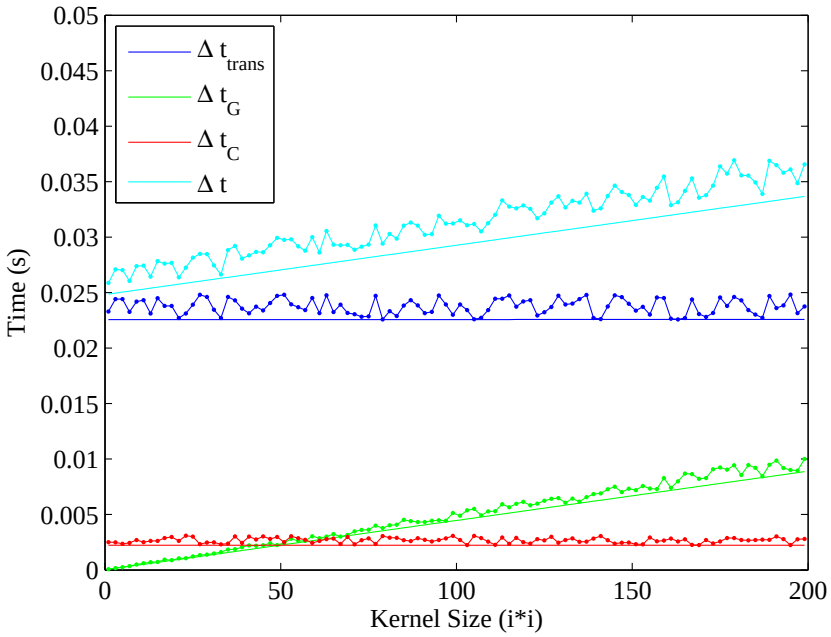


Figure 5.4: Time cost of all components for Setup2 with fixed grid space

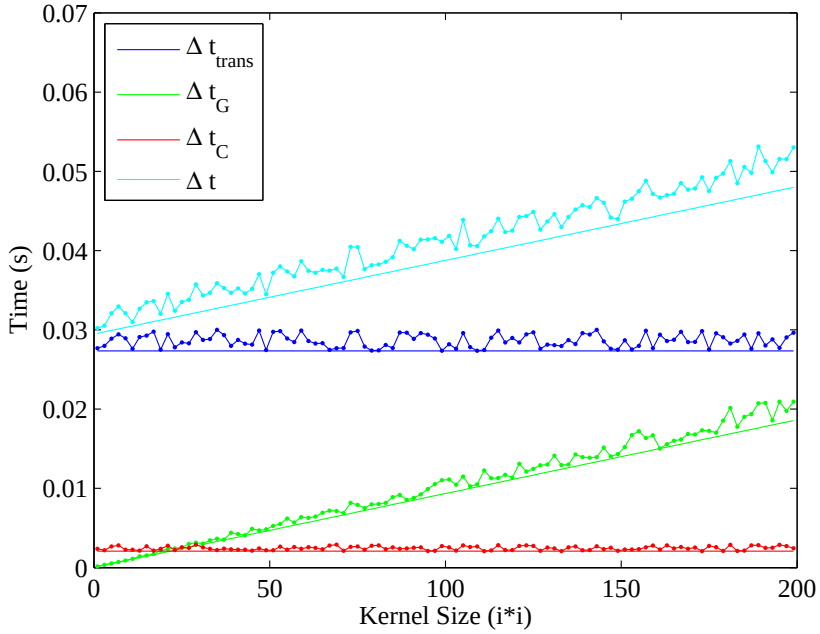


Figure 5.5: Time cost of all components for Setup3 with fixed grid space

Table 5.2: Quantitative results for Test 1

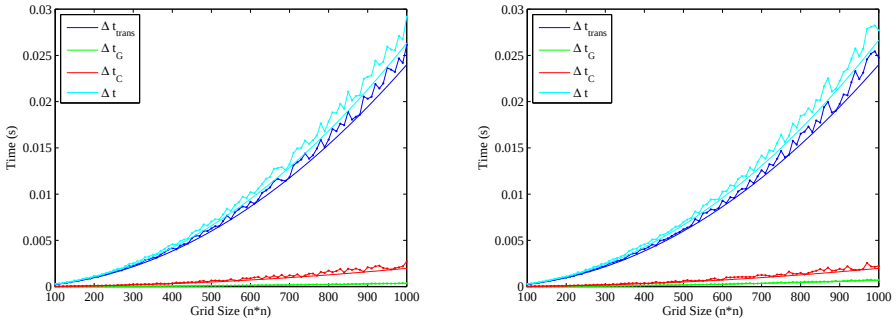
	Average Relative Error		
Setup	1	2	3
Δt_{trans}	1.159 ms	1.165 ms	1.305 ms
Δt_{G}	0.216 ms	0.462 ms	0.856 ms
Δt_{C}	0.402 ms	0.446 ms	0.382 ms
Δt	1.777 ms	2.073 ms	2.543 ms
	(5.88%)	(6.55%)	(6.05%)
	Maximum Relative Error		
Setup	1	2	3
Δt_{trans}	2.351 ms	2.254 ms	2.670 ms
Δt_{G}	0.716 ms	1.464 ms	3.259 ms
Δt_{C}	0.779 ms	0.857 ms	0.818 ms
Δt	3.228 ms	4.149 ms	6.081 ms
	(10.63%)	(11.24%)	(11.45%)

5.3.5.2 Test 2

Test 2 is performed by fixing the convolution kernel size of the proposed parallel grid-based RBE technique to 16×16 or 32×32 and varying grid space size $n_x = n_y = n$ from 100 to $1,000$. These convolution kernel sizes often represent the target motion model with sufficient accuracy, and the grid space size $n = 1,000$, which creates $1,000,000$ grid cells, also provide good accuracy in many practical problems. Similarly to Test 1, the square grid size enables two dimensional visualization of results.

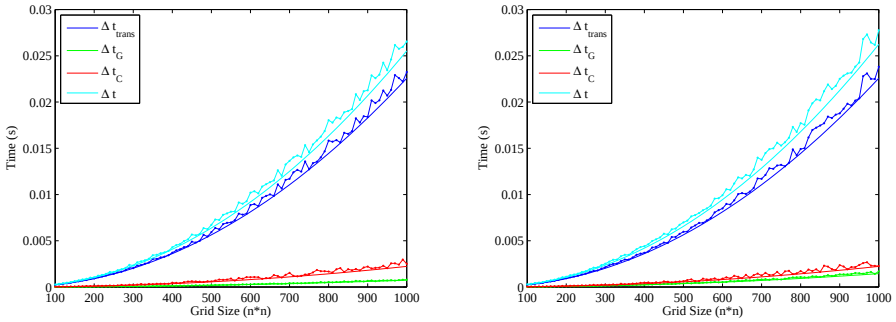
Table 5.3: Quantitative results for Test 2

	Average Relative Error		
Setup	1	2	3
Δt	0.513 ms	0.530 ms	0.617 ms
	(5.59%)	(5.68%)	(5.90%)
	Maximum Relative Error		
Setup	1	2	3
Δt	2.140 ms	2.491 ms	2.835 ms
	(10.08%)	(10.64%)	(10.26%)



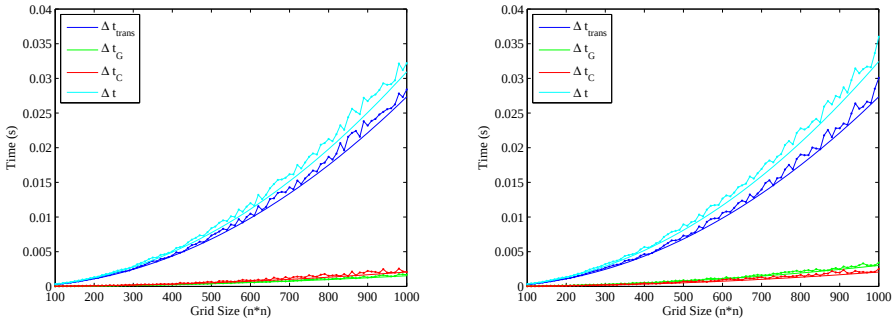
(a) Kernel Size = 16 (b) Kernel Size=32

Figure 5.6: Time cost of all components for Setup1 with fixed kernel



(a) Kernel Size = 16 (b) Kernel Size=32

Figure 5.7: Time cost of all components for Setup2 with fixed kernel



(a) Kernel Size = 16 (b) Kernel Size=32

Figure 5.8: Time cost of all components for Setup3 with fixed kernel

The results of Test 2 for all the components of the time cost for the three computer setups are shown in Figure 5.6, 5.7 and 5.8, respectively. These figures firstly show that the proposed DTFLOP modeling is also able to well estimate the actual performance of the proposed parallel grid-based RBE regardless of different grid space size. Similarly to Test 1, Table 5.3 shows the small average and maximum relative errors, which are below 6% and 11% respectively. Secondly from these results, it is seen that the total time cost is dominated by the time cost of the data transmission particularly when the ratio of the grid space size to the convolution kernel size is large. Since the data transmission rate is determined by the quality of the memory, the utilization of a high quality memory is the first priority for the fast RBE.

5.4 Numerical studies

This section presents the results of a series of qualitative and quantitative tests carried out to investigate capability of the proposed parallel grid-based RBE and the belief fusion techniques. The setup specifications are shown in Table 5.4. Test 1-3 investigate the validity and the real-time performance of the proposed parallel grid-based RBE and belief fusion techniques through algebraic computations whereas its applicability to cooperative search and tracking is examined in Test 4.

Table 5.4: Test computer system specifications II

CPU	Intel Core2Duo, 2.4GHz
RAM	3.25 GB
GPU	Nvidia GeForce 8400GS

5.4.1 Test 1

Test 1 is aimed to investigate the real-time performance of the prediction process of the proposed parallel grid-based RBE with varying convolution kernel size and target space since its speedup and computational time are governed by them. As the indication of the proposed prediction formulation that is 100% parallelizable of the proposed parallel grid-based RBE, it is expected to see an acceleration compared with the conventional prediction process. Speedup is defined as the ratio of the time cost of one iteration of

the conventional sequential grid-based RBE to the proposed parallel grid-based RBE. The convolution kernel,which represents the target motion model, and target space are created artificially.

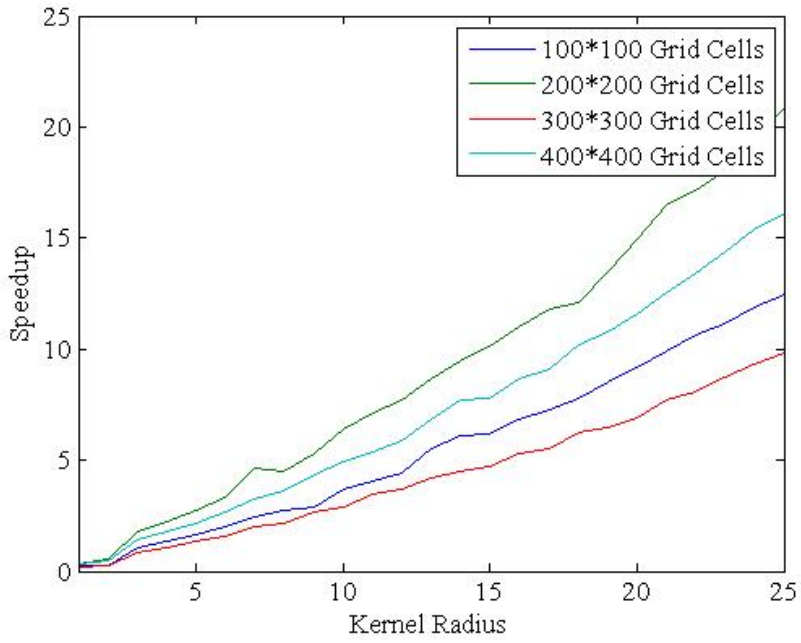


Figure 5.9: Speedup vs. kernel radius

Figure 5.9 shows the results of speedup whereas the results of the time cost required for one iteration of the prediction process of the proposed parallel grid-based RBE with respect to varying convolution kernel size is shown in Figure 5.10. The results of speedup show that the speedup increases in a quadratic fashion with the increase in the convolution kernel size. The time cost of the prediction process of the proposed parallel grid-based RBE stays low and indicates its real-time capability regardless of convolution kernel size. Since the time cost does not increase much with the reasonable chosen convolution kernel the convolution kernel size should be selected simply to capture the motion of a target.

5.4.2 Test 2

Having verified the real-time performance of the prediction process of the proposed parallel grid-based RBE, the one of all the prediction, correction and belief fusion

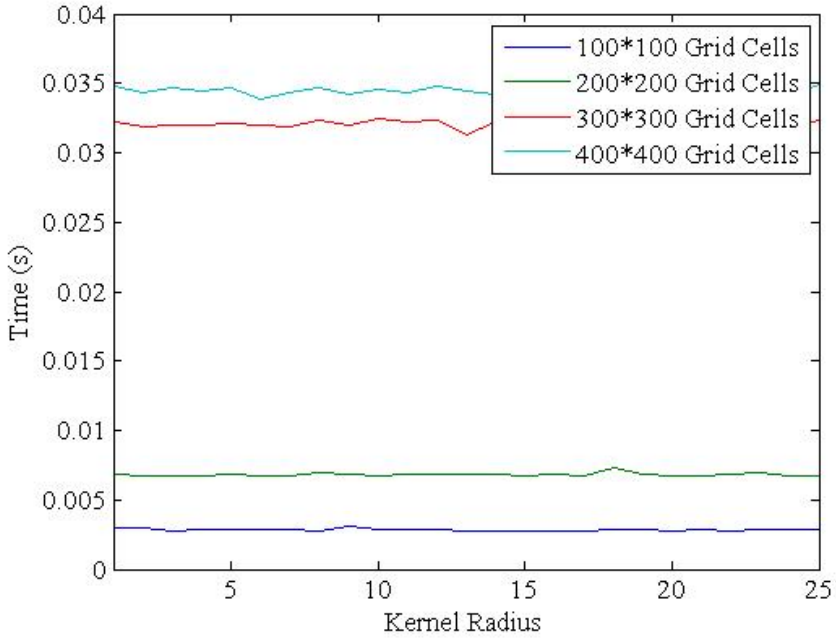


Figure 5.10: Time vs. kernel radius

processes are investigated by varying the size of grid space. Similarly to Test 1, the target grid space and convolution kernel are created artificially. The convolution kernel size is set to be 25 as the practical enough size to capture the motion of a target.

Figure 5.11 and 5.12 show the resulting speedup and the time cost with respect to different grid size, respectively. The time cost for all the prediction, correction and belief fusion processes are low. This verifies the real-time capability of the proposed parallel grid-based RBE and belief fusion techniques. Speedup for the correction and belief fusion processes are observed low compared with that for the prediction process. Since the correction process formulation of the proposed parallel grid-based RBE, Equation 4.11, indicates only partial parallelizable computation, the speedup is expected not as high as the prediction process, which is validated in the speedup result.

5.4.3 Test 3

The last validation test is performed to investigate the effectiveness of the proposed belief fusion technique compared with the conventional observation fusion technique. The time cost required for one iteration of the proposed parallel grid-based RBE using

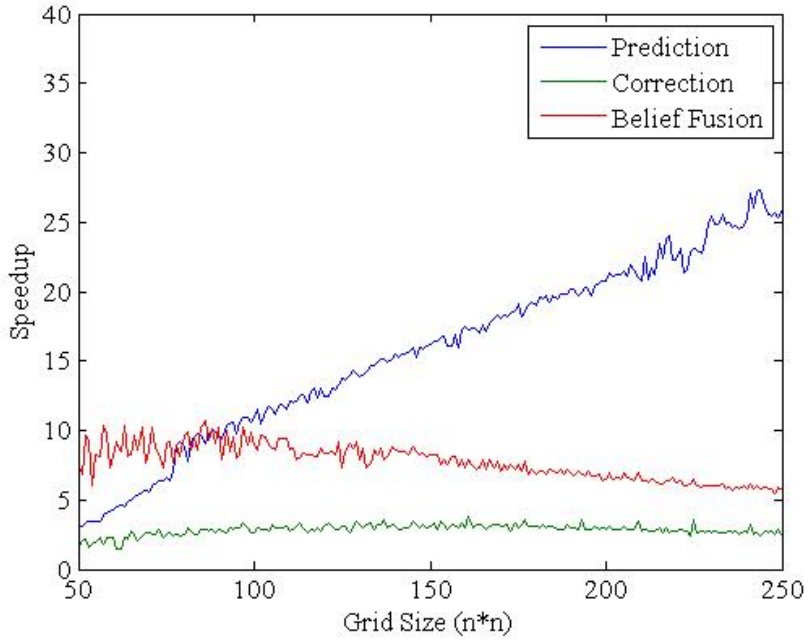


Figure 5.11: Speedup vs. grid size

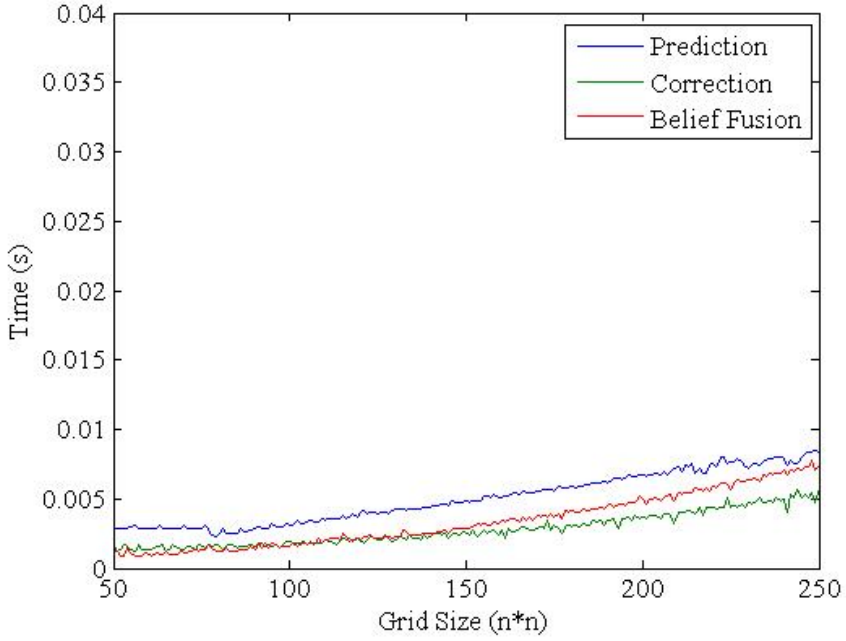


Figure 5.12: Time vs. grid size

the proposed belief fusion technique is first measured and compared with that using the conventional observation fusion technique. Since the beliefs from other sensor platforms or autonomous vehicles carry the past information the belief fusion technique needs not to be performed frequently. The time cost is also investigated with respect to different frequency of the belief fusion.

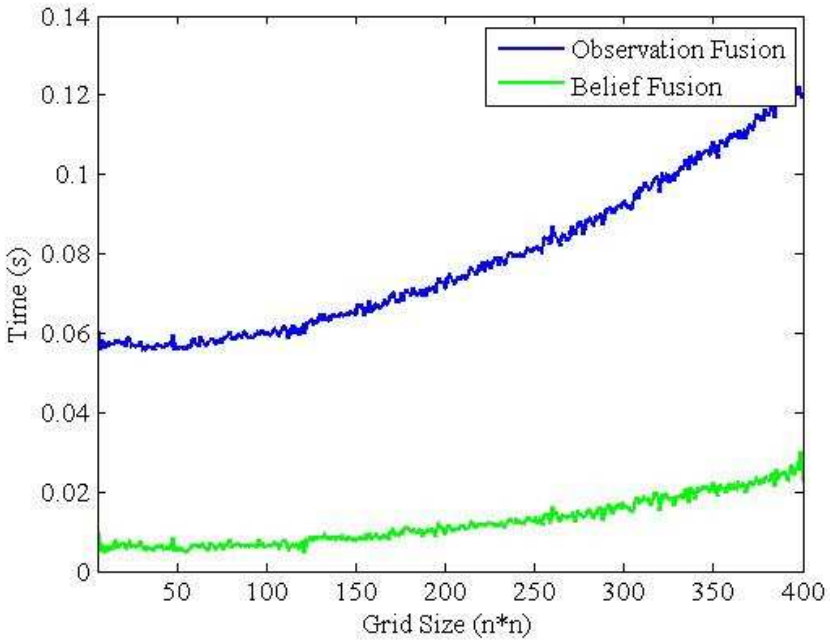


Figure 5.13: Belief fusion (time vs grid size)

Figure 5.13 shows the time cost of one iteration of the proposed parallel grid-based RBE necessary for the proposed belief fusion and the conventional observation fusion techniques. Since the proposed belief fusion technique performs outside the loop of RBE its process is simply consists of the prediction and correction processes. As a consequence, the conventional observation fusion is slower than the proposed belief fusion technique. Since the time cost of the proposed parallel grid-based RBE with observation fusion is nearly 5 times as long as that without observation fusion, the conventional observation fusion technique is equivalent to losing the information of 5 RBEs from the other sensor platforms.

Shown in Figure 5.14 are the time cost required for 100 iteration of the proposed parallel grid-based RBEs with varying intervals of the belief fusion and that with the

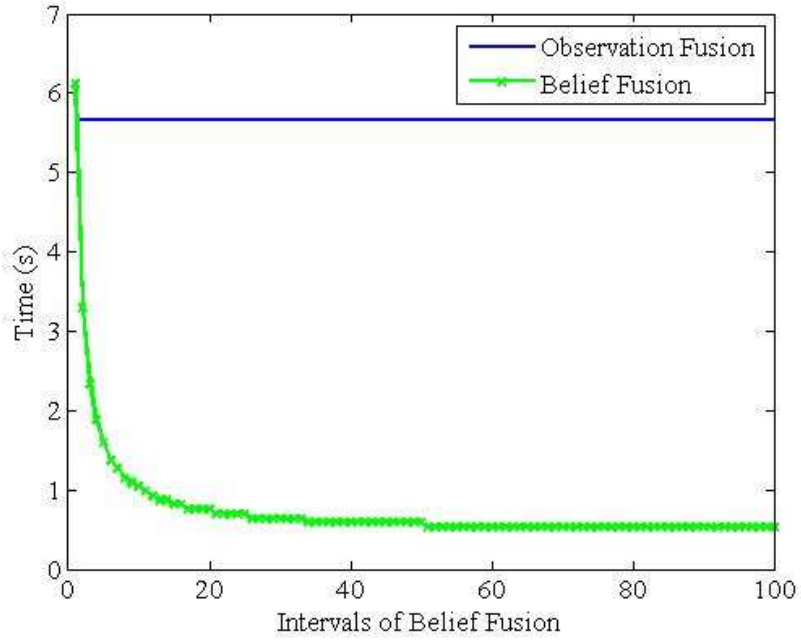


Figure 5.14: Belief fusion (time vs frequency)

conventional observation fusion at every RBE. Estimation with belief fusion at every RBE will take longer than the RBE with the observation fusion since the belief fusion requires communication of the entire belief. However, it is seen that the proposed belief fusion requires half the time cost of conventional observation fusion when the number of intervals is around five. If the accuracy of the belief can be maintained with less belief fusions, the time cost can be reduced by one order. In addition, the conventional observation fusion technique may not become feasible when the communication delay due to the fact that the distance among sensor platforms is introduced to this result. The result strongly justifies the superiority of the proposed belief fusion technique to the conventional observation fusion for the cooperative estimation.

5.4.4 Test 4

The problem described in Test 4 is a simplified marine search and rescue scenario where a life raft with prior belief is drifted by the wind and current and the autonomous rescue helicopters search for and track the life raft to rescue victims. The life raft or target

motion model moves on a horizontal plane and is given by

$$\begin{aligned} x_{k+1}^t &= x_k^t + \Delta t \cdot v_k^t cos\gamma_k^t \\ y_{k+1}^t &= y_k^t + \Delta t \cdot v_k^t sin\gamma_k^t \end{aligned} \tag{5.23}$$

where v^t and γ^t are the velocity and direction of the target motion caused by the wind and current, each subject to a Gaussian noise, and Δt is the time increment. The prior belief on the target is also Gaussian. The sensor platforms or autonomous helicopters are assumed to move on a horizontal plane and given by

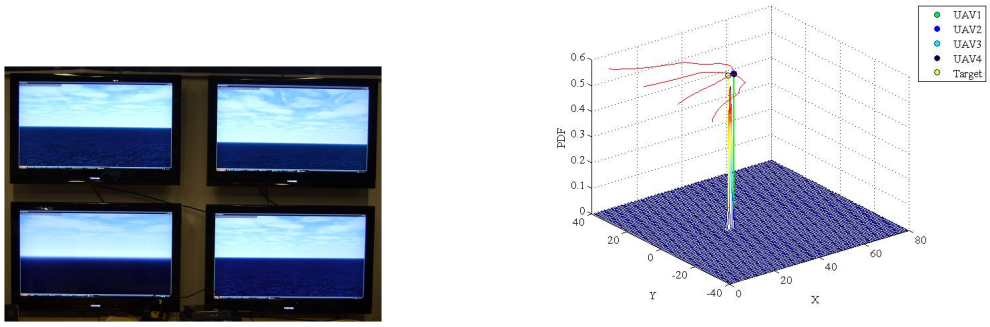
$$\begin{aligned} x_{k+1}^{s_i} &= x_k^{s_i} + \Delta t \cdot v_k^{s_i} cos\gamma_k^{s_i} \\ y_{k+1}^{s_i} &= y_k^{s_i} + \Delta t \cdot v_k^{s_i} sin\gamma_k^{s_i} \\ \theta_{k+1}^{s_i} &= \theta_k^{s_i} + \Delta t \cdot \alpha^{s_i} \gamma_k^{s_i} \end{aligned} \tag{5.24}$$

where v^{s_i} and γ^{s_i} are the velocity and turn of the sensor platform, s_i, and α^{s_i} is a coefficient governing the rate of turn. The probability of detection $P_d(\mathbf{x}_k^t|\mathbf{x}_k^{s_i})$ is given by a Gaussian distribution, whereas the likelihood $l(\mathbf{x}_k^t|^{s_i}\tilde{\mathbf{z}}_k^t, \tilde{\mathbf{x}}_k^{s_i})$ when the target is detected is given by a Gaussian distribution with variances proportional to the distance between the sensor platform s_i and the target. Table 5.5 shows the major parameters of this simulated cooperative search and tracking problem. The communication speed of 70Mbps is a known peak performance of 802.11n in real world.

Table 5.5: Major parameters of simulated cooperative search and rescue

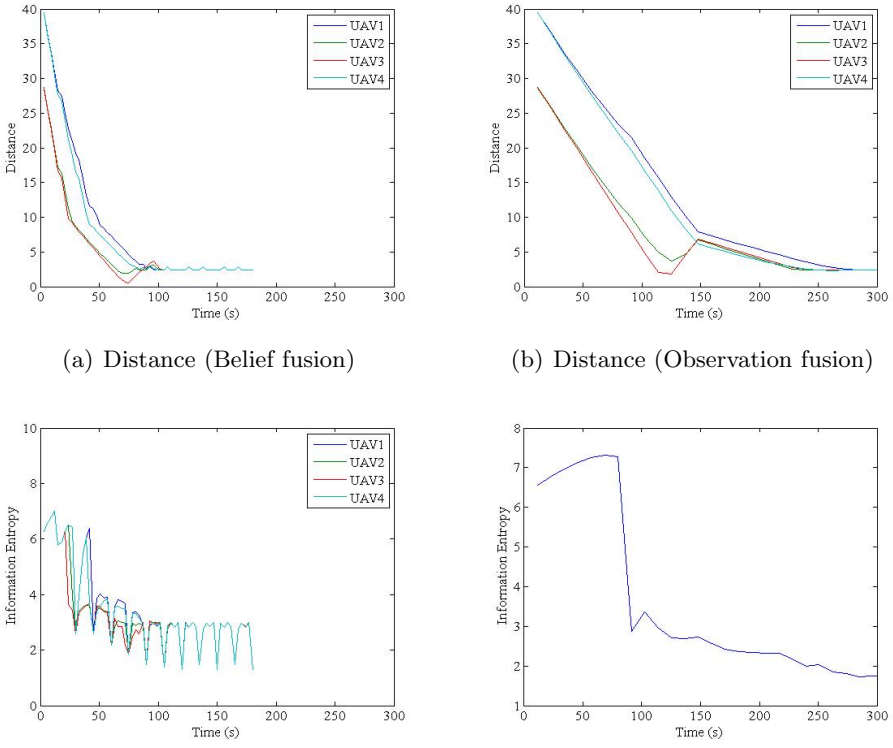
	Parameter	Value
Sensor Platform, s_i	Velocity $v_k^{s_i}$	0.12km/s
	Turn coef. α^{s_i}	0.8
	PoD var.	[0.2km, 0.2km]
	Communication	70Mbps
Target, t	Velocity v_k^t	N(0.1km/s, 0.02km/s)
	Direction γ_k^t	N(0rad, 0.7rad)
	Prior $[x_0^t, y_0^t]$	N([−1km, 1km], diag{0.3km, 0.2km })

Figure 5.15 first shows the snapshot of the cooperate search and rescue and the trajectories of the helicopters with the belief on the target. These show that the successful cooperative search and rescue using the proposed parallel grid-based RBE with



(a) Snapshot of cooperative search and rescue (b) Result of cooperative search and rescue

Figure 5.15: Cooperative search and rescue (Test 4)



(a) Distance (Belief fusion)

(b) Distance (Observation fusion)

(c) Entropy (Belief fusion)

(d) Entropy (Observation fusion)

Figure 5.16: Distance to object and information entropy (Test 4)

the proposed belief fusion. Figure 5.16 then shows the distance of each helicopter to the target and the information entropy with respect to the time by both the proposed belief fusion and conventional observation fusion techniques. The resulting transition of distances shows that the proposed belief fusion outperforms conventional observation fusion technique by finding the target significantly earlier although the belief fusion is performed at every 500 RBEs. The slow performance of the conventional observation fusion is a result of excessive communication with delay. The information entropy of the proposed belief fusion technique is similarly better than that of the conventional observation fusion technique due to the earlier detection of the target. Although infrequent belief fusion in the proposed parallel grid-based RBE makes the information entropy high after a certain period of time, all the helicopters could still keep detecting the target and maintain the information entropy low on average.

5.5 Summary

The novel parallel grid-based RBE technique which derives the new formulations and identifies the parallel computation to accelerate the conventional grid-based RBE has been proposed. By fusing the beliefs, which contain not only the observation information but also the target motion information, from all the sensor platforms or autonomous vehicles the belief fusion technique for the cooperative estimation has been presented. The proposed parallel grid-based RBE technique was implemented in the GPU and further validated the DTFLOP modeling by comparing the estimated time cost with the actual time cost of the parallel grid-based RBE. The superiority of the proposed parallel grid-based RBE technique is investigated via a series of numerical examples in comparison with the conventional grid-based RBE technique.

The results of the validation for the DTFLOP modeling in this Chapter show that the estimated error for the time cost of one iteration of the parallel grid-based RBE technique is less than 6% in average and 11% in maximum value. Compared with the time cost for the computation performed on the CPU and the GPU, the time cost for the data transmission counts nearly 90% of the total time cost. The results of the proposed parallel grid-based RBE technique indicate that the proposed technique accelerates the conventional grid-based RBE technique by at least 10 times and the real-time performance becomes achievable. Moreover, the prediction process of the

proposed parallel grid-based RBE technique shows the most significant speedup, up to 25, because of its complete parallelism whereas the correction and the belief fusion processes show the speedup up to 3 and 10 respectively. The proposed belief fusion technique shows its advantage of the speed as well as the ability to maintain at least 3 times more information of the target compared with the conventional observation fusion technique by the results of the numerical examples.

Chapter 6

Part 2: Full-field Measurements

This Chapter describes the full-field measurement technique for measuring the displacement and strain on a deformed surface of a structure. It has the advantage of nondestructive, field and accurate measurements of a structure. The undeformed surface is first captured as the reference images and the full-field measurement technique measures the displacement and strain on the surface while the structure is deforming. There are two fundamental processes of the full-field measurement technique: the image analysis process and the field estimation process. With the help of the computer vision techniques the image analysis process extracts the features on the captured images and derives the sparse displacement measurements of the deformed surface. In order to provide the smooth field measurements of the displacement and the strain on the deformed surface, the field estimation process takes place by interpolating the sparse displacement measurements into the dense displacement and strain measurements using the shape functions.

This Chapter is organized as follows. Section 6.1, image analysis process, firstly describes an ordinary setup for the full-field measurement technique and then presents the formulations of the feature extraction and the sparse displacement measurements. The field estimation process is presented in Section 6.2 including the interpolation from the sparse displacement measurements to the full-field displacement and strain measurements using the shape functions.

6.1 Image analysis

Figure 6.1 shows a schematic diagram of a typical setup for the full-field measurement experiment. There are a group of n_c cameras, labeled as $\{c_1, c_2, ..., c_{n_c}\}$, and each camera is able to capture the entire surface when the structure is deforming. The pose of each camera is fixed with respect to a reference frame $\{R_0\}$, which is defined on the undeformed surface, and the coordinate frame defined by the camera c_i is $\{R_{c_i}\}$. The pose of the camera c_i can be determined by a camera calibration process and is represented by a transformation matrix ${}^{\{R_{c_i}\}}_{\{R_0\}}P$. The displacement measurement on the deformed surface is obtained by tracking the movements of the n_f features, labeled as $\{f_1, f_2, ..., f_{n_f}\}$, on the captured images from the undeformed reference images to the deformed images.

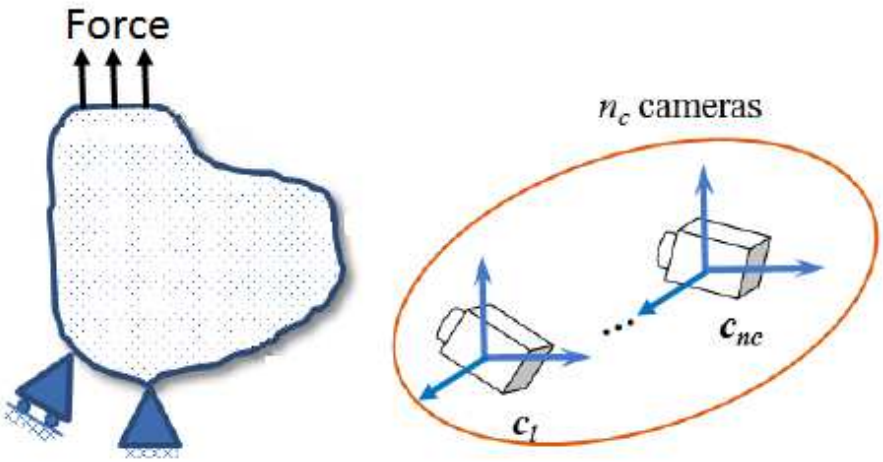


Figure 6.1: Schematic diagram of the full-field measurement experimental setup

There are a number of features, which can be utilized in the full-field measurement technique, and they can be either the manually marked physical features on the deformed surface or the visual features extracted on the captured images. Physical marked features are primary adopted in the full-field measurement technique because of the ease of identification and extraction and invariance from different captured images. On the other hand, although the visual features do not require additional work to mark on the surface they are not robust to be tracked since their sensitivity to the

illumination, viewport of the cameras and large motion. The following two subsections describe the extraction of two typical features, the speckle feature and dot feature.

6.1.1 Speckle feature

For the speckles on the surface it is hard to track each individual speckle on the captured image due to the fact that the size of the speckle is small and each individual speckle does not contain enough information to distinguish itself from other speckles. Instead, the feature is defined in terms of a combination of speckles. The surface is divided into a number of feature blocks, each contains a few speckles, and one can track the movement of each feature block using digital image correlation technique. Figure 6.2 shows a typical captured image of the speckles on the surface (left) and the change in shape of a feature block before and after the deformation (right). The digital

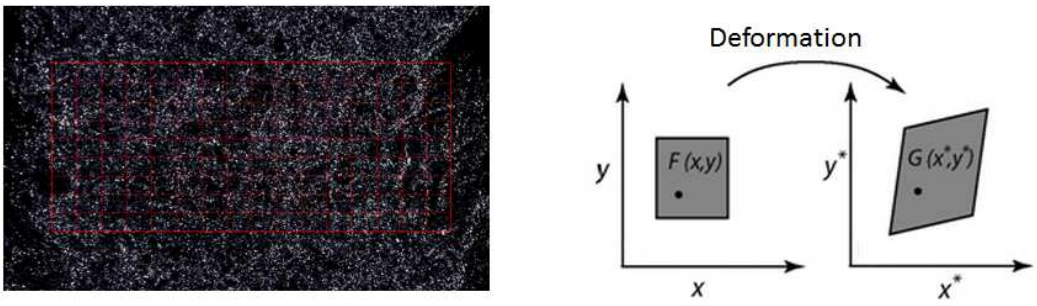


Figure 6.2: Speckle features and digital image correlation (source: google images, under fair use, 2014)

image correlation technique maximizes a correlation coefficient that is determined by examining the grayscale value of a feature block before and after the deformation on the surface to measure the movement of the feature block on the captured image. The formulation of the correlation coefficient is given by

$$r_{ij} = 1 - \frac{\sum_i \sum_j (F(x_i, y_j) - \bar{F})(G(x_i^*, y_j^*) - \bar{G})}{\sqrt{\sum_i \sum_j (F(x_i, y_j) - \bar{F})^2 \sum_i \sum_j (G(x_i^*, y_j^*) - \bar{G})^2}} \tag{6.1}$$

where $F(x_i, y_j)$ is the grayscale value at a point (x_i, y_j) on the undeformed image, $G(x_i^*, y_j^*)$ is the grayscale value at a point (x_i^*, y_j^*) on the deformed image, $\bar{F}$ and $\bar{G}$ are mean values of the grayscale values in F and G, respectively.

6.1.2 Dot feature

The dots marked on the surface appear as the clear dots on the captured image and the size is much larger than that of speckles. Each dot is considered as a unique feature and is tracked on the captured image individually. Since the color of the marked dots is usually chosen to contrast the color of the surface the extraction of those dot features can be achieved by thresholding the captured image in grayscale and then executing the blob extraction algorithm (76). Figure 7.1 shows the process from the captured color image (left) to the thresholded binary image (middle) to the extracted dots on the image (right). The position of the feature f_j on the captured image I_{c_i} is defined

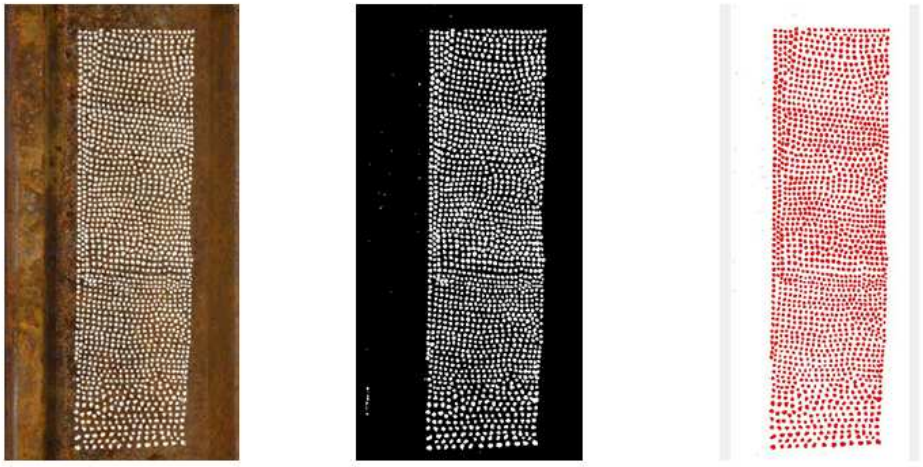


Figure 6.3: Dot features

as ${}^{\{I_{c_i}\}}\mathbf{x}_j$, where $i \in \{1, 2, ..., n_c\}$ and $j \in \{1, 2, ..., n_f\}$ and it is given by

$$ {}^{\{I_{c_i}\}}\mathbf{x}_j = \frac{\sum_{l=1}^{n_l} d_l \, {}^{\{I_{c_i}\}}\mathbf{p}_l}{\sum_{l=1}^{n_l} d_l}, \tag{6.2} $$

where n_l is the number of pixels inside the jth dot feature, d_l is the grayscale value of the lth pixel and ${}^{\{I_{c_i}\}}\mathbf{p}_l$ is its position on the captured image I_{c_i}.

6.2 Field estimation

Applying the multiple view geometry technique (45), which performs the global optimization using the transformation $\{{}^{\{R_{c_1}\}}_{\{R_0\}}P, {}^{\{R_{c_2}\}}_{\{R_0\}}P, ..., {}^{\{R_{c_{n_c}}\}}_{\{R_0\}}P\}$, the position of the feature f_j with respect to the coordinate frame $\{R_0\}$ is obtained as ${}^{\{R_0\}}\mathbf{x}_j$. Define the position of the feature f_j with respect to the coordinate frame $\{R_0\}$ is ${}^{\{R_0\}}\mathbf{x}_{j,u}$ and ${}^{\{R_0\}}\mathbf{x}_{j,d}$ for the undeformed surface and the deformed surface, respectively. The displacement of the feature f_j is given by

$$ {}^{\{R_0\}}\mathbf{u}_j = {}^{\{R_0\}}\mathbf{x}_{j,d} - {}^{\{R_0\}}\mathbf{x}_{j,u}, \tag{6.3} $$

where $j \in \{1, 2, ..., n_f\}$. It is noted that the displacement measurement is a three dimensional vector ${}^{\{R_0\}}\mathbf{u}_j = [{}^{\{R_0\}}(u_x)_j, {}^{\{R_0\}}(u_y)_j, {}^{\{R_0\}}(u_z)_j]$ in metric unit and represents the movement of the feature on the deformed surface.

The field estimation process computes the displacement and strain field by interpolating the measured feature displacements into total n_m interpolated points which cover the entire deformed surface. The displacement at the mth interpolated point, ${}^{\{R_0\}}\mathbf{x}_m$, is given by

$$ {}^{\{R_0\}}\mathbf{u}_m = \sum_{j=1}^{n_t} N_{m,j} c_j {}^{\{R_0\}}\mathbf{u}_j \tag{6.4} $$

and the strain is given by

$$ {}^{\{R_0\}}\boldsymbol{\varepsilon}_m = [\sum_{j=1}^{n_t} \frac{\partial N_{m,j}}{\partial x} {}^{\{R_0\}}(u_x)_j, \sum_{j=1}^{n_t} \frac{\partial N_{m,j}}{\partial y} {}^{\{R_0\}}(u_y)_j, \frac{1}{2}\sum_{j=1}^{n_t} \frac{\partial N_{m,j}}{\partial x} {}^{\{R_0\}}(u_y)_j + \frac{1}{2}\sum_{j=1}^{n_t} \frac{\partial N_{m,j}}{\partial y} {}^{\{R_0\}}(u_x)_j] \tag{6.5} $$

where $N_{m,j} = N_j({}^{\{R_0\}}\mathbf{x}_m)$ is the shape function evaluated at ${}^{\{R_0\}}\mathbf{x} = {}^{\{R_0\}}\mathbf{x}_m$.

Those shape functions are determined by the numerical interpolation techniques. In terms of the requirement of the mesh generation on the surface one can divide the numerical interpolation techniques into two types. Finite element interpolation, the most widely used technique, defines the mesh on the deformed surface and performs the interpolation using the shape function constructed from the vertices, edges and elements. Meshfree interpolation, on the other hand, does not require the mesh generated on the deformed surface but needs more computational power to calculate the shape functions.

6.3 Summary

The two processes, image analysis and field estimation processes, of the full-field measurement techniques were described in this Chapter. In the image analysis process, the speckle feature is extracted using the digital image correlation technique whereas the dot feature is extracted by the pixel grayscale values inside the dot feature. The positions of the extracted features on the captured image are transformed to a unified coordinate frame, when the surface is unformed and the sparse displacement measurements are obtained in the metric unit. The field estimation process applies the shape functions for displacement measurements and interpolates into the field measurement of the displacement and strain on the deformed surface.

Chapter 7

Part 2: Parallel DCT Full-field Measurements

This Chapter presents the novel parallel dot centroid tracking (DCT) full-field measurement technique for measuring the displacement and strain on the deformed surface of a structure. The proposed parallel DCT full-field measurement technique identifies and develops the parallel computation in the image analysis and the field estimation processes and then is implemented into the GPU to accelerate the conventional full-field measurement techniques. In order to accommodate both indoor and outdoor experimental environments a hardware system, which contains two digital cameras, LED lights and adjustable sturdy support, is developed. The software package, which implements the proposed parallel DCT full-field measurement technique, and the corresponding graphic user interface are also presented. In the end, the DTFLOP modeling is applied to estimate the performance of the proposed parallel DCT full-field measurement technique and its performance is validated and investigated by a series of experiments.

This Chapter is organized as follows. Section 7.1 and Section 7.2 presents the parallel dot centroid derivation process and the parallel MLS meshfree interpolation of the proposed parallel DCT full-field measurement technique respectively. The GPU implementation of the proposed parallel DCT full-field measurement technique is presented in Section 7.3. Section 7.4 describes the developed hardware system and graphic user interface. Finally, a series of numerical examples are presented in Section 7.5 and the experiments, in both indoor and outdoor environments, for measuring the displacement

and strain of the rails are presented in Section 7.6.

7.1 Parallel image analysis process

For the DCT full-field measurement technique, the image analysis process first recognize the marked dots on the captured images of the deformed surface. The recognition process is performed by thresholding the grayscale image and applying the connected component labeling technique (103). The connected component labeling technique groups the connected pixels into the marked dots on the captured image and its implementation utilized in this dissertation is a sequential computational implementation on the CPU and the detail of the algorithm is out of the scope. After all the marked dots are recognized it is easy to compute their centroids using the recognized dots, each of which contains the grayscale information inside the dot. A typical marked dot on the captured image is shown in Figure 7.1. The centroid of the marked dot f_j on the

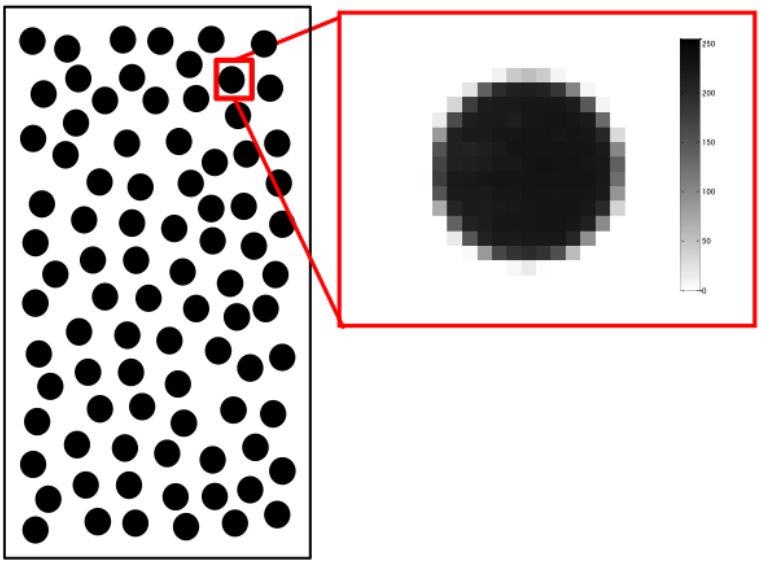


Figure 7.1: A typical marked dot on captured image

captured image I_{c_i} is defined as ${}^{\{I_{c_i}\}}\mathbf{x}_j$, where $i \in \{1, 2, ..., n_c\}$ and $j \in \{1, 2, ..., n_f\}$ and it is given by

$$ {}^{\{I_{c_i}\}}\mathbf{x}_j = \frac{\sum_{l=1}^{n_l} d_l \, {}^{\{I_{c_i}\}}\mathbf{p}_l}{\sum_{l=1}^{n_l} d_l}, \tag{7.1} $$

where n_l is the number of pixels inside the jth marked dot, d_l is the grayscale value of the lth pixel and ${}^{\{I_{c_i}\}}\mathbf{p}_l$ is its position on the captured image I_{c_i}.

Observation the above formulation it is easily seen that the centroid derivation of each marked dot is completely independent. Since each marked dot has its own information the computational parallelism is achievable and the practical number of the marked dots is usually in the order of 100 or 1000. The parallel computational implementation of this centroid derivation process is expected to dramatically accelerate the image process analysis process of the DCT full-field measurement technique.

7.2 Parallel MLS meshfree interpolation

For the field estimation process of the full-field measurement technique the displacement and strain field are interpolated by certain shape functions. The finite element based interpolation requires the construction of the mesh over the measured surface and the interpolation is performed based on the generated mesh, which includes vertices, edges and elements. On the other hand the meshfree interpolation does not have the requirement of the mesh and the interpolation is performed in terms of each interpolated points on the surface and can be implemented in the way of parallel computation. The moving least square (MLS) meshfree interpolation is selected in this dissertation and composes in the proposed parallel DCT full-field measurement technique. As shown in Figure 7.2 the displacement and strain measurement at the interpolated point is computed using the displacement measurement of the marked dots. Given n_f marked dots and n_m interpolated points on the deformed surface, MLS meshfree interpolation computes the displacement and strain field measurements. A circle whose center located at the mth interpolated point is defined as the support of domain and the radius of the circle is ρ_m. The support of domain determines the accuracy of the MLS meshfree interpolation and its computational speed. Suppose that there are l marked dots within the support of domain ρ_m. The following computation is under the coordinate frame of $\{R_0\}$ and to simplify the notation the coordinate superscript is dropped for all the variables. The displacement measurement at the mth interpolated point is given by

$$\mathbf{u}_m = [\Phi_m(\mathbf{U}_x)_m, \Phi_m(\mathbf{U}_y)_m], \tag{7.2}$$

where Φ_m is the shape function for the MLS meshfree interpolation, $(\mathbf{U}_x)_m$ and $(\mathbf{U}_y)_m$ are the vectors which include the displacement measurements of l marked dots within

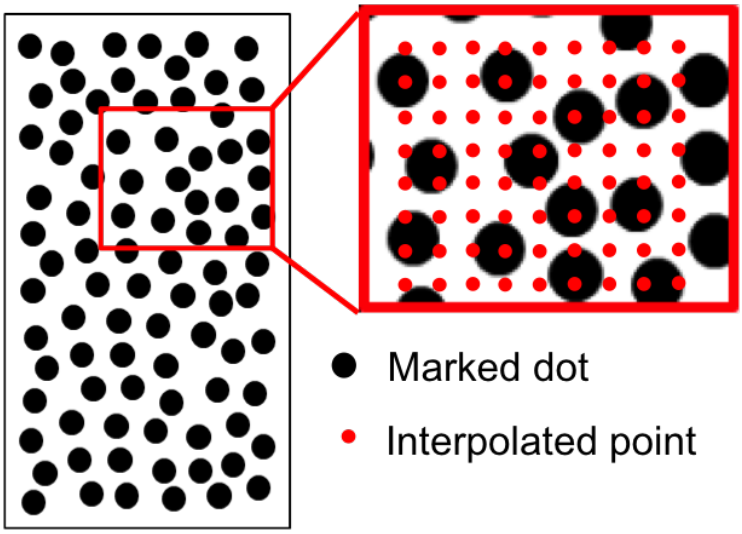


Figure 7.2: MLS meshfree interpolation

the support of domain ρ_m in x and y direction, respectively. The MLS meshfree shape function is defined as

$$\Phi_m = p'(\mathbf{x}_m)(A_m)^{-1}B_m, \tag{7.3}$$

where $p'(\mathbf{x})$ is a row vector which represents the polynomial basis and its transpose vector is $p(x)$. In the scope of this dissertation $p'(\mathbf{x})$ is defined as

$$p'(\mathbf{x}) = [1, x, y, x^2, y^2, xy]. \tag{7.4}$$

A_m and B_m are the two numerical matrices and are given by

$$A_m = \sum_{j=1}^{l} W_{\rho_m}(\mathbf{x}_m, \mathbf{x}_j)p(\mathbf{x}_j)p'(\mathbf{x}_j) \tag{7.5}$$

and

$$B_m = \{W_{\rho_m}(\mathbf{x}_m, \mathbf{x}_1)p(\mathbf{x}_1), W_{\rho_m}(\mathbf{x}_m, \mathbf{x}_2)p(\mathbf{x}_2), ..., W_{\rho_m}(\mathbf{x}_m, \mathbf{x}_l)p(\mathbf{x}_l)\} \tag{7.6}$$

respectively, where $W_{\rho_m}(\mathbf{x}_m, \mathbf{x}_j)$ is a scalar weight function, which associates the interpolated point $\mathbf{x}_m$ to the marked dot $\mathbf{x}_j$ within its support of domain ρ_m. The weight function plays an important role in the performance of the MLS meshfree interpolation. It should be constructed so that it is positive and that a unique solution of the shape

function is guaranteed. Substituting Equations (7.5), (7.6) and (7.3) into the Equation (6.5), the strain at the mth interpolated point is thus given by

$$\varepsilon_m = [\frac{\partial \Phi_m}{\partial x}(\mathbf{U}_x)_m, \frac{\partial \Phi_m}{\partial y}(\mathbf{U}_y)_m, \frac{1}{2}\frac{\partial \Phi_m}{\partial x}(\mathbf{U}_y)_m + \frac{1}{2}\frac{\partial \Phi_m}{\partial y}(\mathbf{U}_x)_m]$$

By observing the above processes, it is shown that the computation of the displacement and strain for all the interpolated points is performed independently. This means that the field estimation phase is completely parallelizable using the proposed MLS meshfree interpolation. However, those equations which derive the shape function also show the computational efficiency depends on the size of support of domain. Larger support of domain indicates more measurements of the marked dots are utilized to derive the shape function, expecting to result in better accuracy but higher computational cost.

7.3 GPU implementation

7.3.1 Shared buffer & look-up table

Since both the proposed dot centroid derivation and MLS meshfree interpolation are completely parallelizable it is obvious that implementation of the proposed parallel DCT full-field measurement technique into the GPU would accelerate the computational speed. However, it is noted that for each interpolated point multiple marked dots measurements which lays inside the support of domain are required. When it comes to a large and high resolution field, the requirement of a large number of interpolated points leads to a large memory space requirement to store their associated marked dots measurements and make the GPU implementation infeasible due to the limited fast local memory on the GPU.

Shared buffers with a predefined look-up table strategy is developed to overcome this issue. At the GPU initialization, two shared buffers with the size of the number of marked dots, n_f, are allocated on the GPU's shared memory to store derived centroids and displacement measurements. A look-up table is initialized on the GPU's global memory and filled up by computing the indices of the marked dots that associate to each interpolated point given a support of domain. Due to the fact that there is no duplicate index for the element of the look-up table the type of elements can be

represented by a binary bit array to save the unnecessary GPU's memory compared to an integer array.

7.3.2 Implementation

Figure 7.3 shows the schematic diagram of the GPU implementation for the proposed parallel DCT full-field measurement technique. n_f marked dots with their grayscale information are firstly identified by the dot recognition process, which is performed in the CPU on the captured images. Then n_f threads are initialized by the GPU to compute the centroids of n_f marked dots in parallel and pass the results into the shared buffer on the GPU's local memory. With the help of the precomputed look-up table associated marked dots measurements are grouped together as the input of the proposed parallel MLS meshfree interpolation for each interpolated point. Note that each group of associated measurements are independently transferred to the GPU's local memory to be prepared for the proposed parallel MLS meshfree interpolation. In addition, n_m GPU threads are evoked to perform the proposed parallel MLS meshfree interpolation of the displacement and strain fields and the visualization result is outputted in the end.

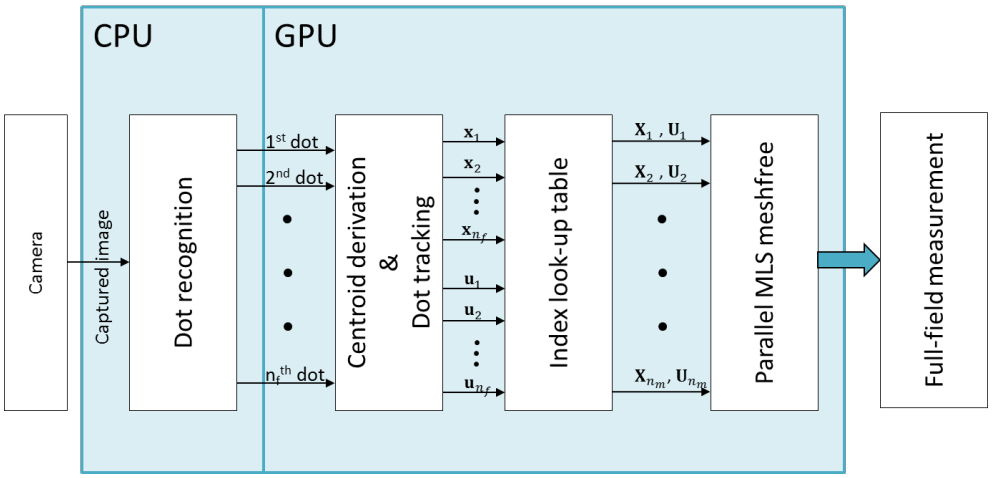


Figure 7.3: GPU implementation for proposed parallel DCT full-field measurement technique

7.4 System development

7.4.1 Hardware system

Figure 7.4 shows the developed hardware system for the proposed parallel DCT full-field measurement technique. Two Nikon D3200 digital single-lens reflex (DSLR) cameras with the resolution of 24 Megapixels are selected to capture the images of the deformed surface. Two cameras are located 50cm away from each other to guarantee the accuracy of the three dimensional measurements. The ball tripod heads provide the smooth movements of the cameras to adapt any shooting angles. The adjustable LED light mounted in the middle is utilized to provide additional lights when the natural lights are not sufficient. The locking level control supports provide the flexibility of installing this system to any ground surface. The entire frame is weighted for 15 lbs and is made by sturdy metal to reduce the effect of the ground vibration during the experiments.

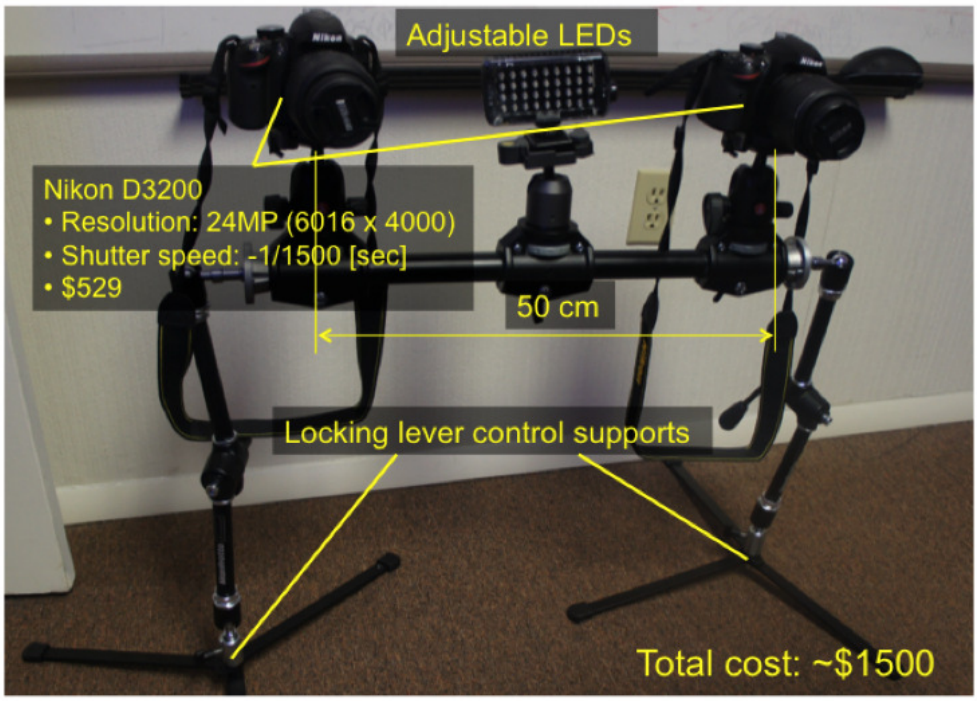


Figure 7.4: Hardware system for parallel DCT full-field measurement

7.4.2 Graphic user interface (GUI)

Figure 7.5 shows the developed graphic user interface (GUI) for the proposed parallel DCT full-field measurement. The developed GUI supports up to 4 DSLR cameras or PointGrey CCD cameras and provides the modes of online or offline full-field measurements. The full-field displacement and strain measurements are visualized using the color map and the three dimensional shape of the deformed surface are shown as well. Two interpolation techniques, finite element interpolation and the proposed parallel MLS meshfree interpolation, are implemented in the GUI. The developed GUI is consist of Main menus, Widget tabs and Plot areas. A user manual for performing the full-field measurement experiment using the developed GUI is described in Appendix A. The following subsections briefly describe the functionalities of Main menus, Widget tabs and Plot areas, respectively:

7.4.2.1 Main menus

Main menus contain the most of the action commands and are illustrated as follows:

- File menu is to create or open a new project, save or retrieve it from hard disk. A predefined configuration file can be loaded to run the proposed parallel DCT full-field measurement technique without adjusting parameters at the widget tabs.
- View menu is to adjust the camera parameters and the position of the measured surface within the field of view from each camera. It also contains commands to toggle the visibility of various items, such as the axis of the coordinate frame, the region of interest, the images from each camera and so on.
- Pre-processing menu is to select the region of interest from the captured image for each camera, initialize the computer vision parameters and adjust the camera parameters.
- Tools menu is to switch the interpolation methods, change the three dimensional surface resolution and export the results or plots.

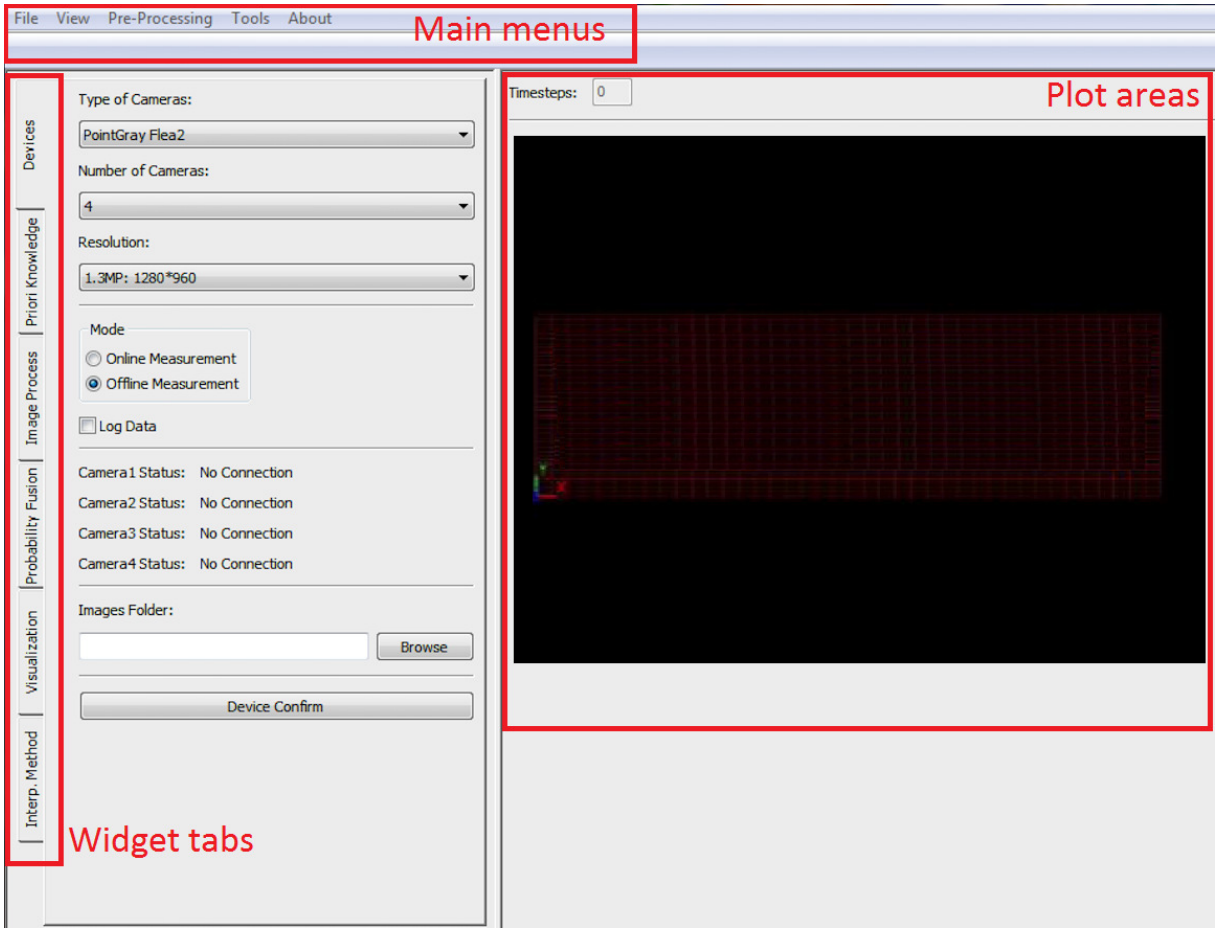


Figure 7.5: GUI for parallel DCT full-field measurement

7.4.2.2 Widget tabs

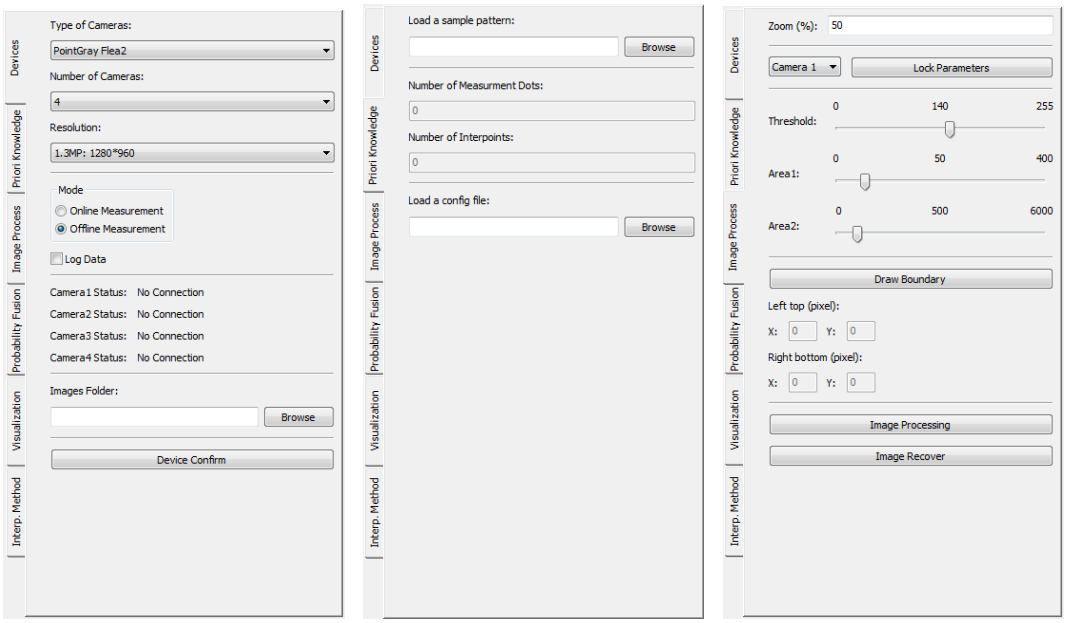
All of the functionality of the proposed parallel DCT full-field measurement technique are in Widget tabs. There are total 6 widget tabs, Device tab, Prior knowledge tab, Image process tab, Probabilistic tab, Visualization tab and Interpolation tab, method, and their usages are described as follows:

- Device tab (Figure 7.6(a)) is to choose the camera system, number of cameras, image resolution and to determine the online or offline measurement mode. The directory of all the captured images need to be provided if the user wants to log the captured images or the offline measurement mode is selected. After the "Device Confirm" button is clicked the device information is passed in the system and the status of each camera is shown on the tab. Current version supports Nikon DSLR cameras, PointGray cameras and webcams compatible with the Windows built-in driver.

- Prior knowledge tab (Figure 7.6(b)) is to load a predefined dot pattern file for the measured surface and a configuration file. The configuration file is adopted the XML format and includes the predefined parameters for the image processing, visualization and other options for the following tabs. Directly loading from a configuration file avoids choosing all the options manually and unnecessary repeated manipulations for the image processing parameters and thus improve the efficiency of the experiments.

- Image process tab (Figure 7.6(c)) is to draw a region of interest (ROI) on the captured image and to adjust the image processing parameters for each camera. After determining the ROI for the captured image of each camera, three image processing parameters need to be properly adjusted by manually manipulating the slidebars. Threshold represents the grayscale threshold value to filter out the clear dots on the captured image. Area1 and Area2 represents the biggest and smallest area threshold to filter out the boundary noise and white noise, respectively. User has to properly adjust those image processing parameters for each camera to utilize multiple cameras.

- Probability fusion tab (Figure 7.7(a)) is to select the probability data fusion techniques. Either multi-frame data fusion or multi-camera data fusion technique can



(a) Devics tab (b) Prior knowledge tab (c) Image processing tab

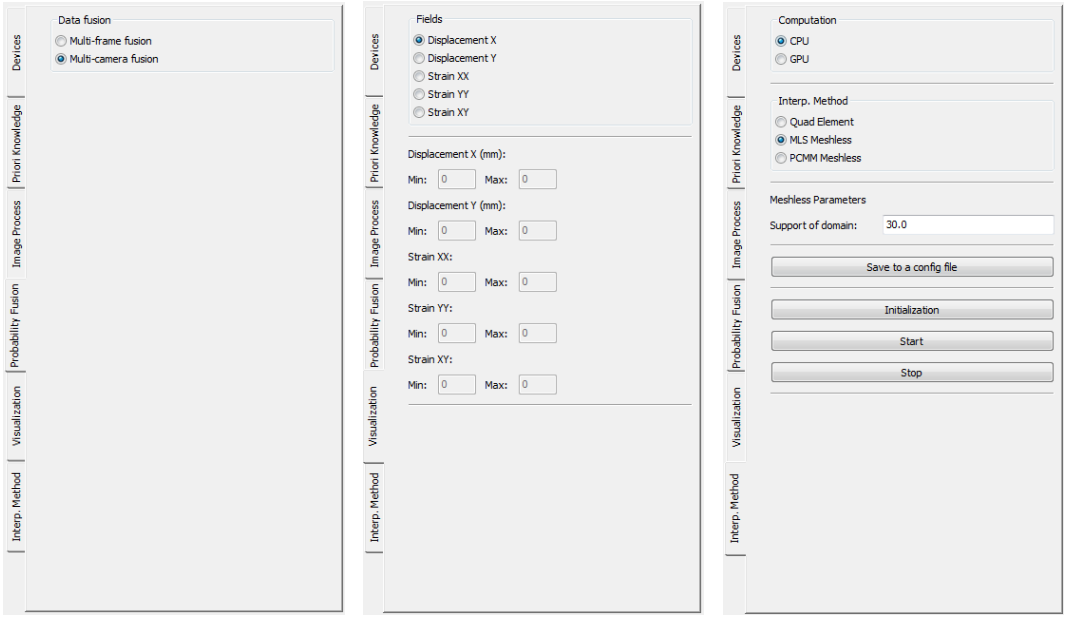
Figure 7.6: Widget tabs I

be selected. It is noted that the data fusion is expected to improve the accuracy of the measurements but to decelerate the proposed parallel DCT technique.

- Visualization tab (Figure 7.7(b)) is to select the visualization options and show the quantitative results of the measured displacement or strain fields. The options include the displacements (u_x, u_y, u_z) in three dimension, the plain strains (ε_{xx}, ε_{yy})on the deformed surface and the shear strain (ε_{xy}). The three dimensional surface visualization with color map is shown based on the selected option in the tab.

- Interpolation method tab (Figure 7.7(c)) is to select the interpolation method, the computation mode and define the required parameters for the chosen interpolation method. There are two interpolation methods options: the finite element interpolation and the MLS meshfree interpolation. When the MLS meshfree interpolation is selected the support of domain needs to be specified. The proposed parallel DCT full-field measurement technique is implemented into both the CPU and GPU and the user can select either one based on the existed computer ca-

pability. The "Save to a config file" button needs to be clicked when the user completes adjusting all the parameters of other tabs. A configuration file, which contains all the adjusted parameters, is saved to avoid repeating the same process for other experiments if the camera setting and the light conditions keep unchanged.



(a) Probability fusion tab (b) Visualization tab (c) Interpolation method tab

Figure 7.7: Widget tabs II

7.4.2.3 Plot areas

Plot areas is the canvas of the visualization. It shows the displacement or strain field measurement in terms of the color map on the deformed surface based on the visualization options selected on Visualization tab. The three dimensional change in shape of the deformed surface is also on Plot areas. The viewport and the zoom can be adjusted by dragging the visualized surface and rolling the mouse wheel respectively. The left, right, up and down arrow buttons on the keyboard control the position of the visualized surface.

7.5 Numerical studies

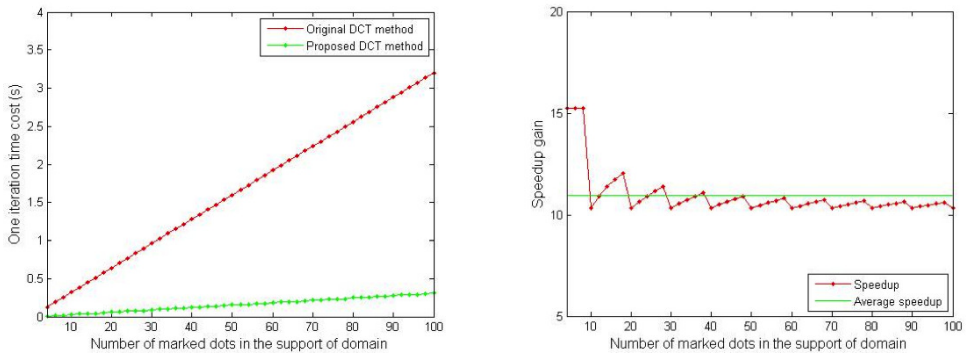
This section presents a series of numerical examples to demonstrate the performance of the proposed parallel DCT full-field measurement technique. At the beginning the performance of the proposed parallel DCT full-field measurement technique is estimated by the proposed DTFLOP modeling. The proposed GPU implementation is then validated and demonstrated its acceleration. In addition the proposed parallel DCT full-field measurement technique is compared with the finite element analysis and traditional sensors to demonstrate its capability for high accuracy measurements. In the end a tensile experiment is performed to demonstrate the proposed parallel DCT full-field measurement technique in comparison with the theoretical analysis.

7.5.1 Performance estimation by DTFLOP modeling

This test is aimed to estimate the performance of the proposed parallel DCT full-field measurement technique by applying the proposed DTFLOP modeling in Chapter 3. The proposed DTFLOP identifies the sequential and the parallel computation of the proposed parallel DCT full-field measurement technique respectively and estimates its time cost. Since the support of domain determines the smoothness and the computational speed of the proposed parallel DCT full-field measurement technique it is chosen as the varying parameters for the application of the proposed DTFLOP modeling. The varying support of domains is represented by the number of marked dots inside it, increasing from 10 to 100 in this test. Figure 7.8(a) and 7.8(b) shows the estimated time cost of one iteration of the proposed parallel DCT full-field measurement technique and its speedup compared with the sequential implementation on the CPU, respectively, from the proposed DTFLOP modeling. The number of the marked dots and the interpolated points are 369 and 1024 respectively. As shown in those two figures the time cost linearly increases with respect to the increase of the support of domain and the average speedup gain is about 11.

7.5.2 Theoretical validation

This set of tests are aimed to investigate the validity of the proposed parallel DCT full-field measurement technique. The computer setup for those tests is shown in Table 7.1. The test data are generated from the ANSYS, a benchmarking software for the



(a) Time cost comparison (b) Speedup gain

Figure 7.8: Estimated performance of proposed parallel DCT full-field measurement technique by DTFLOP modeling

finite element analysis (FEA). FEA is widely used as a benchmarking technique for the analysis of the solid mechanics and is a desired option to validate the proposed DCT full-field measurement technique. The simulated deformed surface is a thin plate specimen with an open hole at its centroid under a tensile loading in the vertical direction. The marked dots are generated for the captured images following the mesh generation process of ANSYS and a total 400 images are generated. There are 369 marked dots and 1024 interpolated points to show the full-field displacement and strain field measurements.

Table 7.1: Computer configuration for theoretical validation

CPU	Intel Dual-Core,2.70GHz
GPU	Nvidia GeForce GT220
Memory	4.0GB
OS	Windows 7 Pro

7.5.2.1 Correctness Test

This test is aimed to validate the correctness of the displacement and strain field measurements of the proposed parallel DCT full-field measurement technique implemented parallel into the GPU in comparison with that implemented sequentially into the CPU. The captured images above are utilized for this correctness test.

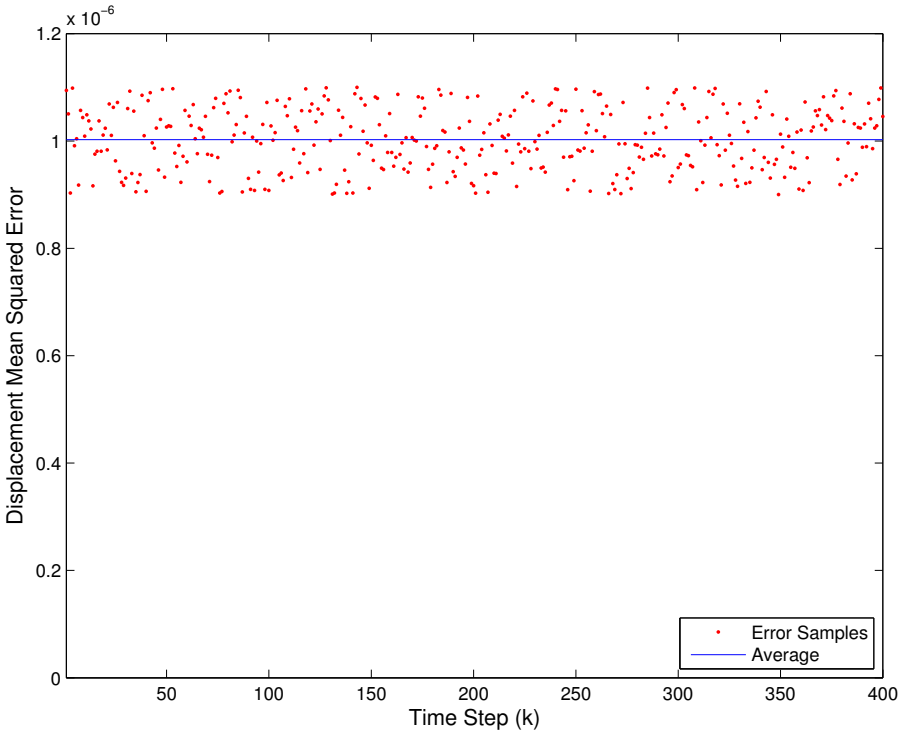


Figure 7.9: Mean square error results

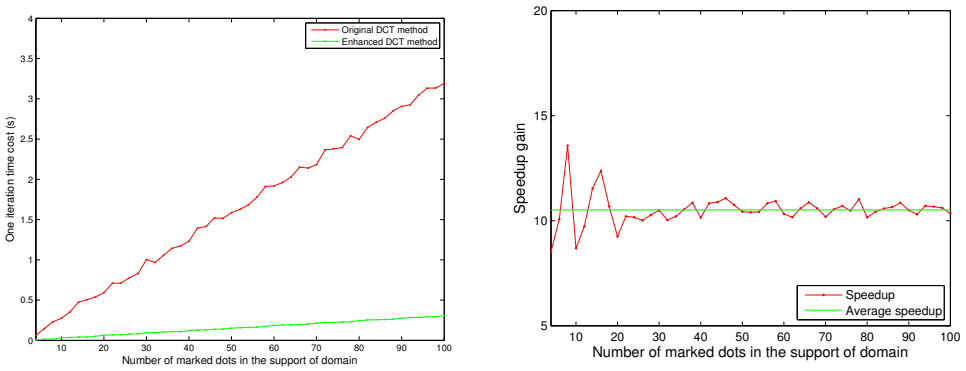
The result for the mean squared errors of the displacement field is shown in Figure 7.9. It indicates that there is a small random error between the sequential implementation into the CPU and the parallel implementation into the GPU of the proposed parallel full-field measurement technique with a mean value of 1.0×10^{-6}. This is of the correct order of magnitude to indicate the usage of the single precision floating point operations on the GPU as opposed to the double precision floating point operations on the CPU.

7.5.2.2 Speedup Test

The performance of the proposed parallel DCT full-field measurement technique has been estimated using the proposed DTFLOP modeling in the subsection above. This test is to perform the same test and further validate the speedup gained from the proposed parallel DCT full-field measurement technique.

Figure 7.10(a) and 7.10(b) shows the actual time cost of one iteration of the proposed parallel DCT full-field measurement technique and its speedup compared with the sequential implementation on the CPU, respectively. The varying support of domains is represented by the number of marked dots inside it as before. On average across the range of supports of domain tested, the proposed parallel DCT full-field measurement technique has been found to gain at least 10 times speedup. When the number of marked dots within the support of domain is less than 10, the time cost of the GPU's initialization contributes the most, resulting in a small speedup gain. With a reasonable support of domain, that is the number of marked dots inside it from 10 to 100, the proposed parallel DCT full-field measurement technique implemented into the GPU achieves a stable 1 order speedup compared with its sequential CPU implementation and the results also confirm the prediction from the DTFLOP modeling.

The number of interpolated points to estimate the displacement and strain measurements on the deformed surface is the other parameter for the proposed parallel DCT full-field measurement technique. Let the number of interpolated points vary from 384 to 2048, integer power of 2 to maximize the performance of the GPU capability. The speedup gain of the proposed parallel DCT full-field measurement technique implemented into the CPU to that into the CPU is shown in Figure 7.11. The result indicates that the speedup gain firstly increases with respect to the increase of the number of the interpolated points since the GPU is able to fully parallelize all the



(a) Time cost comparison

(b) Speedup gain

Figure 7.10: Performance of proposed parallel DCT full-field measurement technique

computation whereas the speedup gain further decreases because the GPU's parallel computational capability limit is reached.

7.5.2.3 Comparison with finite element analysis

This test is to show the displacement and strain full-field measurement results of the proposed parallel DCT full-field measurement technique in comparison with the results from the finite element analysis (FEA). Figure 7.12(a) and 7.12(b) show the comparison results of the full-field displacement field while Figure 7.13(a), 7.13(b) and 7.13(c) show that of the full-field strain field. The displacement field indicates the shrink in the horizontal direction and the elongation in the vertical direction and the result of the proposed parallel DCT full-field measurement technique aligns with the result of the FEA. In addition, the strain field shows the strain concentration around the open hole and its position from the proposed parallel DCT full-field measurement technique coincides with that of the FEA as well. Those results demonstrate the capability of the proposed parallel DCT full-field measurement technique by the benchmarking FEA.

7.5.3 Accuracy evaluation

7.5.3.1 Dot size

The direct measurements of the proposed parallel DCT full-field measurement technique are the marked dots. It is obvious that the accuracy of the proposed parallel DCT full-

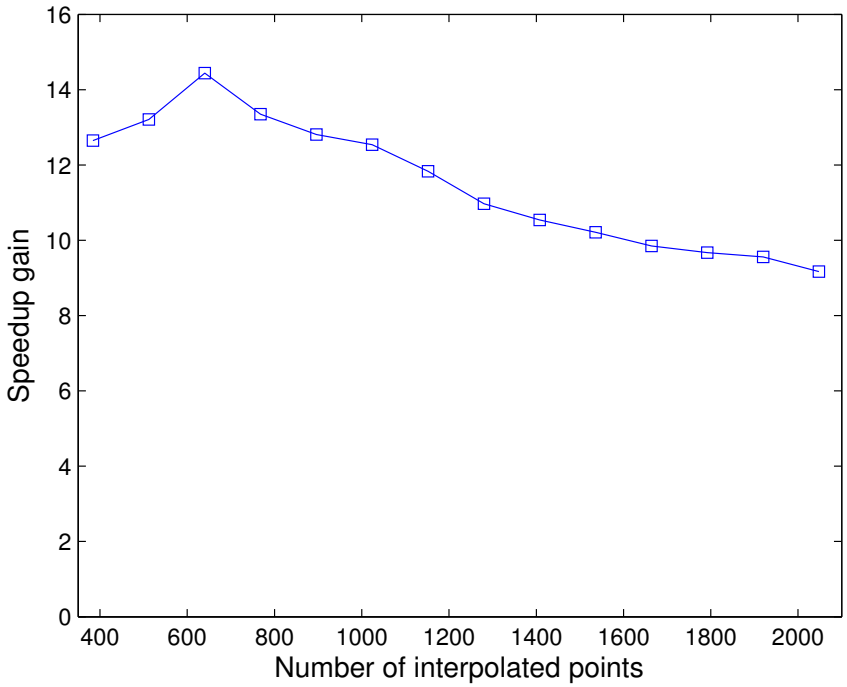
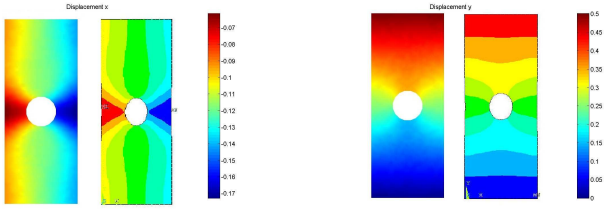
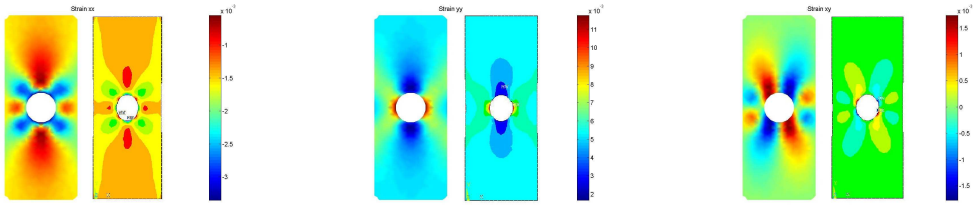


Figure 7.11: Speedup gain vs number of interpolated points



(a) Displacement u_x (b) Displacement u_y

Figure 7.12: Full-field displacement measurement: proposed DCT (left) vs FEA (right)



(a) Strain ε_{xx} (b) Strain ε_{yy} (c) Strain ε_{xy}

Figure 7.13: Full-field strain measurement: proposed DCT (left) vs FEA (right)

field measurement technique relies on the accurate measurement of the marked dots in the image analysis process. Since the centroids of the marked dots are derived from the greyscale information on the captured image (Equation 7.1) this test is aimed to evaluate the accuracy of the measurement with varying size of the marked dots. Figure 7.14 shows an illustration of a marked dots with the radius of 3 pixels on the left and 4 measurements on the captured images on the right in this test. The marked dot is printed and located at the center of the red square on the captured images. The measured radius of the marked dots varies from the radius of 10 pixels to 200 pixels in this test and the standard deviation of the centroid of the marked dot is shown in Figure 7.15. It is seen that there is no obvious difference of the standard deviation in

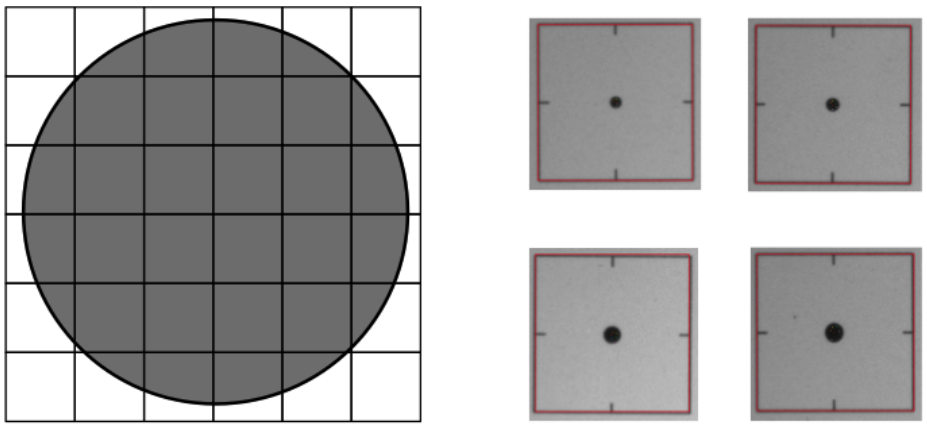


Figure 7.14: Marked dots

the horizontal and the vertical direction and the reason is that DSLR cameras usually

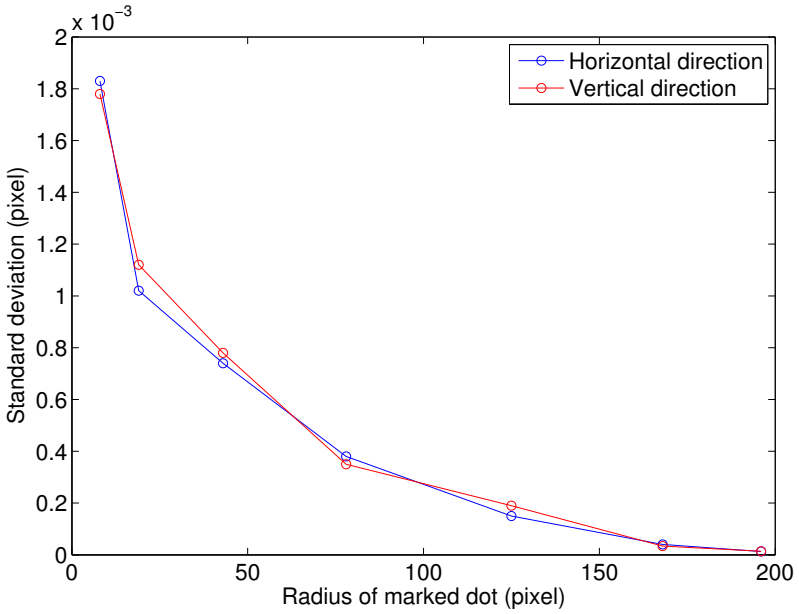


Figure 7.15: Standard deviation vs dot size

have the squared pixels of the captured image. The results show that the standard deviation decreases from the order of 10^{-3} pixel to the order of 10^{-5} pixel and indicate that the large size of the marked dots results in more accurate centroid measurement of the marked dot and is expected to eventually benefit the accuracy of the full-field displacement and strain measurement.

7.5.3.2 Dot density

It is obvious that the other factor which can affect the accuracy of the full-field measurement is the density of the marked dots. As shown in the MLS meshfree interpolation of the proposed parallel DCT full-field measurement technique the support of domain is the primary parameter, which can affect the accuracy of the field estimation results. If the support of domain is fixed increasing the density of marked dots means that there are more marked dots or centroid measurements inside the support of domain. Thus the density of the marked dots can be represented in terms of the number of marked dots inside a support of domain. In this test the MLS meshfree interpolation is executed on a simulated rectangular shape specimen (Figure 7.2) with 0.1 percent elongation, which means a vertical strain value of 0.001. The number of interpolated

points is fixed at 2048 and the density of marked dots, number of marked dots inside the support of domain, varies from 13 to 193. Figure 7.16 shows the strain value of the MLS meshfree interpolation and it is seen that both small and large number of the marked dots inside the support of domain result in an inaccurate measurement. The reasonable number is in the range of 40 to 130.

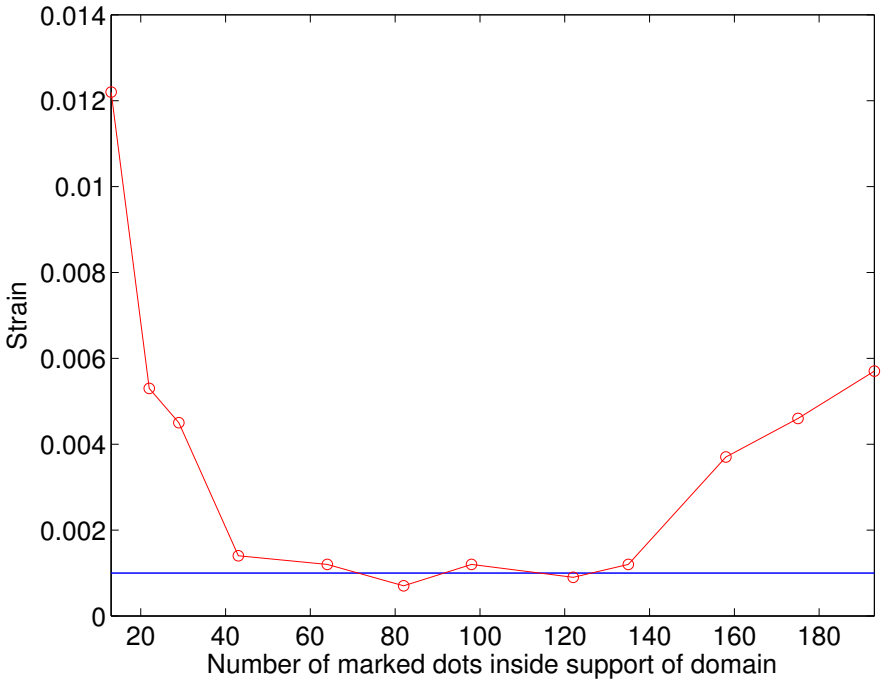


Figure 7.16: Accuracy vs dot density

However, a reasonable large dot density or number of marked dots inside the support of domain also indicate the size of marked dots has to be small and the computational speed is slow. Therefore a tradeoff has to be considered between the dot size and the dot density when the proposed parallel DCT full-field measurement technique is performed. A possible solution would be that applying small dense marked dots in the location which is expected to show more strain concentration and applying large sparse marked dots in other location.

7.5.4 Experimental validation

7.5.4.1 Comparison with traditional sensors

This subsection experimentally validate the proposed parallel DCT full-field measurement technique by comparing the result with the traditional sensors, including strain gauges and extensometers. The experiment is consist of a uniaxial material testing machine and a rectangular composite specimen. The specimen is under a tensile loading and is expected to be elongated. There are 560 black marked dots on the specimen for the proposed parallel DCT full-field measurement technique. The digital cameras used for this experiment are PointGrey Flea2 5MP CCD cameras and the testing machine is SHIMADZU AG-IC 100KN uniaxial testing machine. The displacement is controlled to that is elongate specimen at a 0.2mm/min loading rate. The equipment models and the specification are summarized in Table 7.2. The experimental setup is shown in Figure 7.17.

Table 7.2: Equipment specification

Equipment	Model	Specifications
Digital Camera	PointGrey Flea 2	5MPixels
Testing Machine	Shimadzu AG-IC	100KN capacity
Computer	Dell Vostro 420	Pentium Core i5, 2GB RAM
GPU	Nvidia	GeForce GT220
Extensometer	Shimadzu SSG50-10H	
Strain Gauge	CEA-XX-250UW-350	Resolution: 6-10 microstrain

Figure 7.18 shows the comparison of the strain result among the strain gauge, the extensometer and the proposed parallel DCT full-field measurement technique. It is shown that the proposed parallel DCT full-field measurement technique is able to achieve the same accuracy (less than 20 microstrains) as the strain gauge in the controlled environment.

7.5.4.2 Comparison with theoretical results

This subsection presents a tensile experiment on a open-hole specimen to examine the experimental performance of the proposed parallel DCT full-field measurement technique. The specimen is a thin-plate aluminum specimen with an open hole in

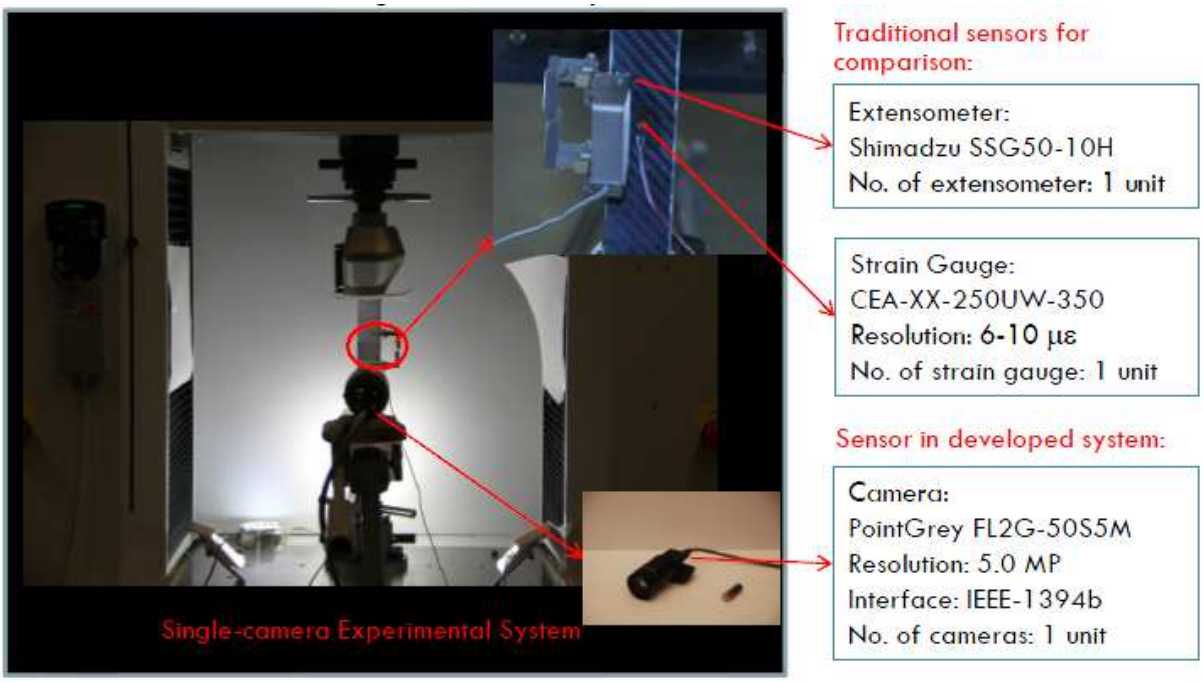


Figure 7.17: Experimental setup for experimental validation

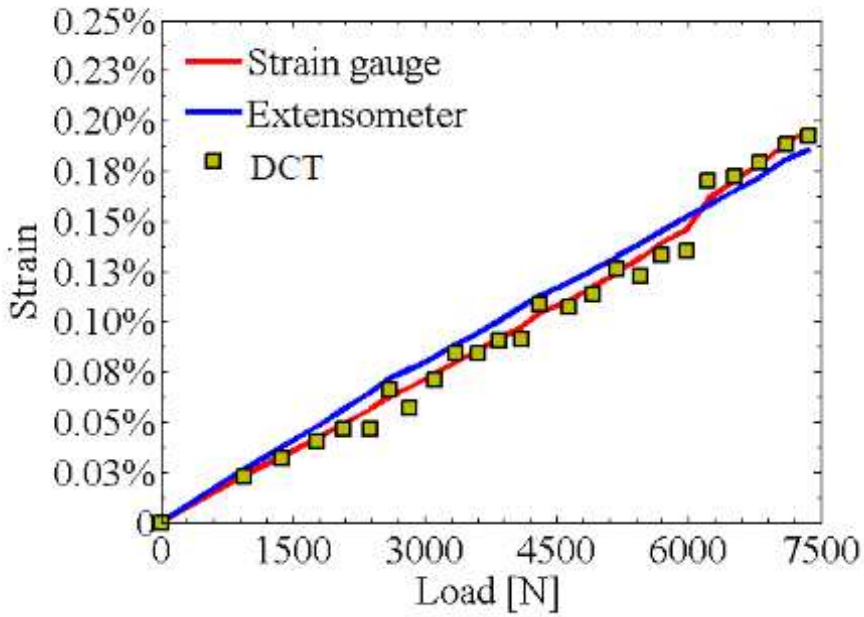
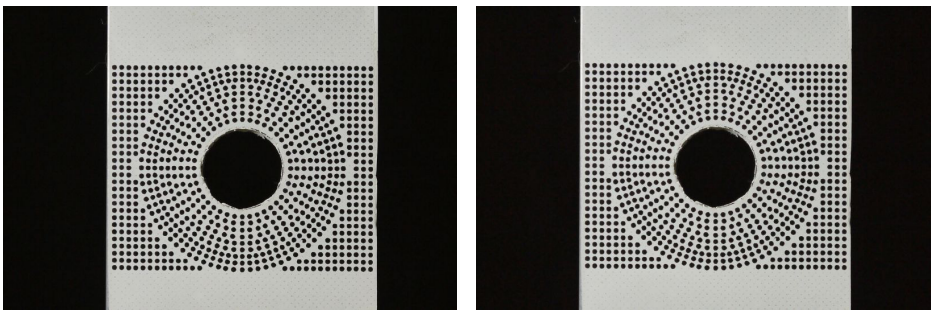


Figure 7.18: Strain comparison among strain gauge, extensometer and proposed DCT technique

its geometric center. Since the strain concentration is expected to appear around the hole boundary 898 black dots are marked around the open hole area. The typical captured images are shown in Figure 7.19 under the controlled light condition, which optimizes the contrast of the black dots with the white background. The specimen is elongated around 2mm in the vertical direction using the same material testing machine, Shimadzu SSG50-10H, as the previous subsection. The full-field measurement results of the displacement are shown in Figure 7.20(a) and 7.20(b). The results show the same field distribution pattern in the theoretical results but tilt in a small angle to lead the fields asymmetric. The reason for this slight tilting may be from the shutter vibration when the images are captured. Figure 7.21(a), 7.21(b) and 7.21(c) show the strain field results, which match the theoretical results of the tensile loading (Figure 7.13(a), 7.13(b) and 7.13(c)) as well. The strain concentration is seen clearly around the boundary of the open hole. Therefore, the proposed parallel DCT full-field measurement technique is experimental validated.

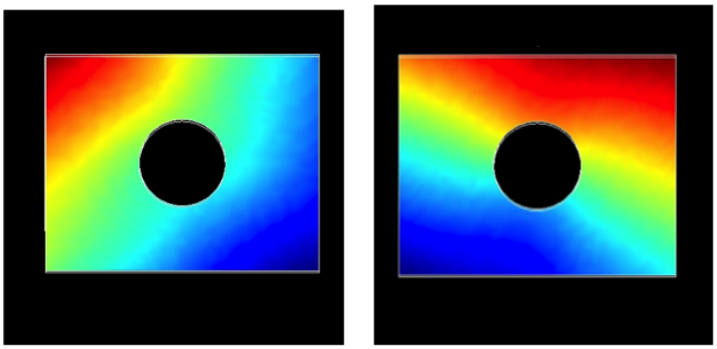


(a) Undeformed (b) Deformed

Figure 7.19: Captured images for proposed parallel DCT full-field measurement technique

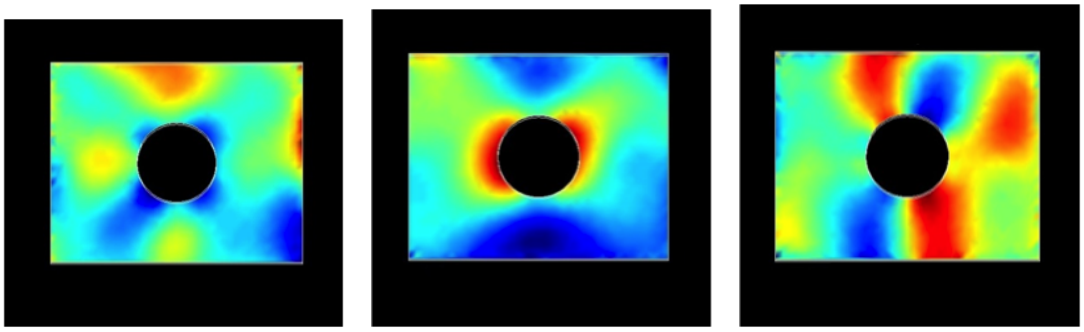
7.6 Railway experiments

This section presents an application of the proposed parallel DCT full-field measurement technique. The application is to measure the deformation or displacement and strain fields on the rail surface when the train is passing. Traditionally the deformation and strain are measured by the strain gauge and its installation and calibration are tedious and labor intensive. The proposed parallel DCT full-field measurement has the advantage of the easy setup, the fast measurements and mostly important the



(a) Displacement u_x (b) Displacement u_y

Figure 7.20: Full-field displacement measurements



(a) Strain ε_{xx} (b) Strain ε_{yy} (c) Strain ε_{xy}

Figure 7.21: Full-field strain measurements

field measurements in comparison with the traditional strain gauge measurements. The proposed parallel DCT full-field measurement technique and the developed hardware system are first applied in the measurements of the rail in the indoor laboratories environment and then applied in that in the outdoor environment when the train is passing on the rail. As shown in the Figure 7.22, the coordinate frame system for the rail is defined as x for the longitudinal direction, z for the vertical direction and y for the lateral direction.

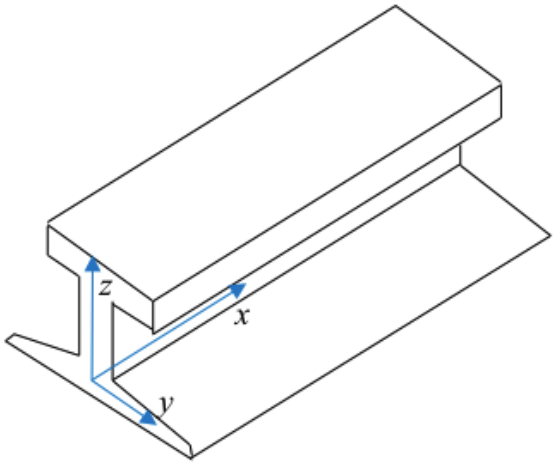
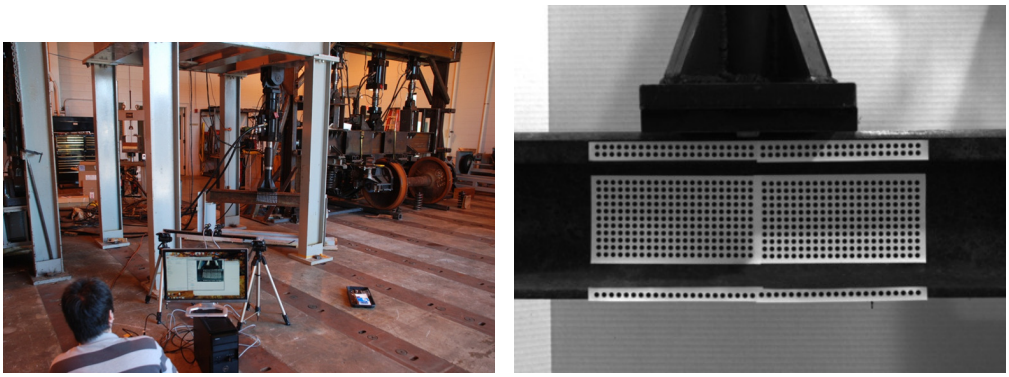


Figure 7.22: Coordinate frame defined on the rail

7.6.1 Indoor laboratorial experiments

7.6.1.1 Rail bending experiement

A three point bending experiment of a rail under a vertical loading downward is performed using the proposed parallel DCT full-field measurement technique. The rail is supported at the two sides and the loading applies for the rail at its middle position. The experimental setup and a typical measurement image, which includes 616 marked dots, are shown in Figure 7.23. The vertical loading during the time of the experiment is shown in Figure 7.24 with the maximum value of $50,000$lbs. A digital camera with the resolution of 5 megapixels is utilized to capture the images for the experiment.



(a) Experimental setup (b) A measurement image of the rail

Figure 7.23: Rail bending experiment

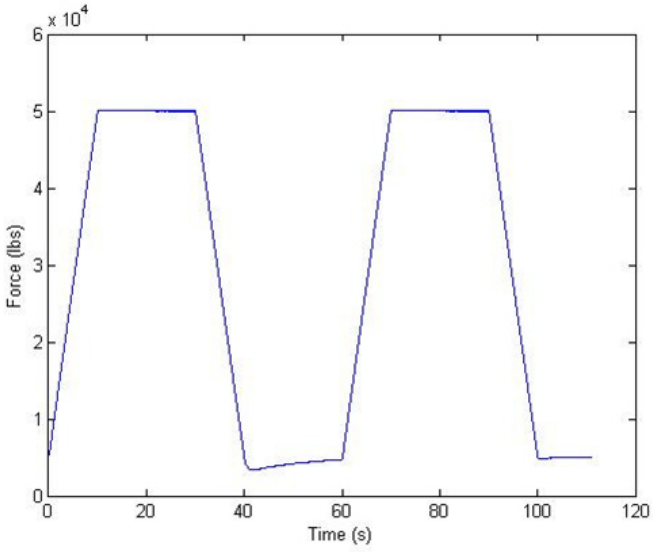
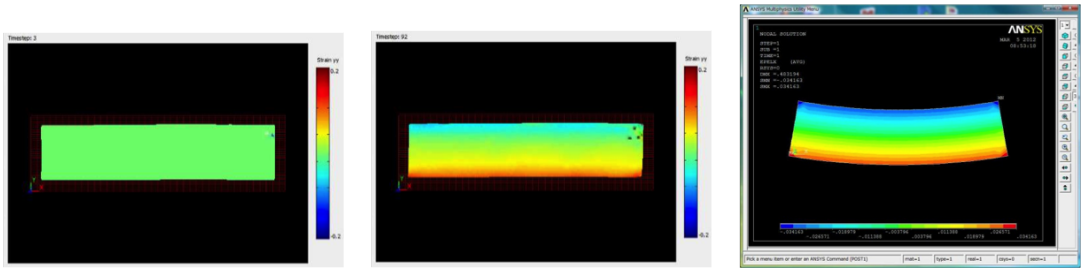


Figure 7.24: Vertical loading for rail bending experiment

Figure 7.25(a) and 7.25(b) shows the longitudinal full-field strain measurement of the undeformed and the deformed rail under the maximum loading respectively. The phenomenon is observed twice during the experiment with respect to the two cycles of the vertical loading. It is seen that the compression appears at the top of the rail and the tension appears at the bottom of the rail. The result matches a three point bending theoretical expectation, which is also shown in Figure 7.25(c) using the FEA.



(a) Undeformed state (proposed DCT) (b) Deformed state (proposed DCT) (c) Deformed state (FEA)

Figure 7.25: Longitudinal strain ε_{xx} in rail bending experiment

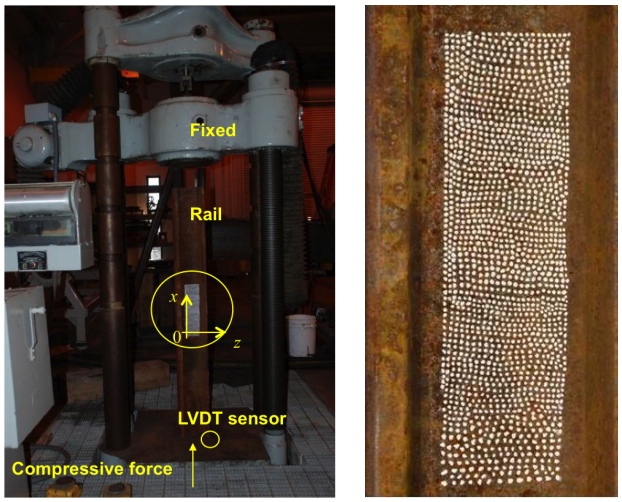
7.6.1.2 Rail compression experiment

This subsection presents a compression experiment of a standard rail. The dimension of the rail is shown in Table 7.3. The loading machine has a fixed top panel and applies the compressive force at the bottom. The rail is lifted and loaded in its longitudinal direction with the compressive force manually applied from 0 to 100,000lbs. A linear variable differential transformer (LVDT) sensor is attached at the bottom of the rail to measure the deformation during the compressive loading. There are 1611 white marked dots with the approximated diameter of 5mm marked on the surface of the rail. The image is captured at every 20,000lbs increment. The experimental setup and a typical captured image is shown in Figure 7.26(a) and 7.26(b), respectively.

The displacement fields in the vertical direction and the longitudinal direction is shown in Figure 7.27(a) and 7.27(b) respectively. In the vertical direction it is seen that the rail is expanded to the two sides of the rail, which aligns with the correct compression field distribution in theory. The reason of the imperfect symmetry in the vertical displacement field may be from the uneven top surface of the rail. The displacement

Table 7.3: Rail dimension

Item	Quantity
Length	147.4cm
Width	17.5cm
Dot pattern length	30.48cm
Dot pattern height	7.62cm

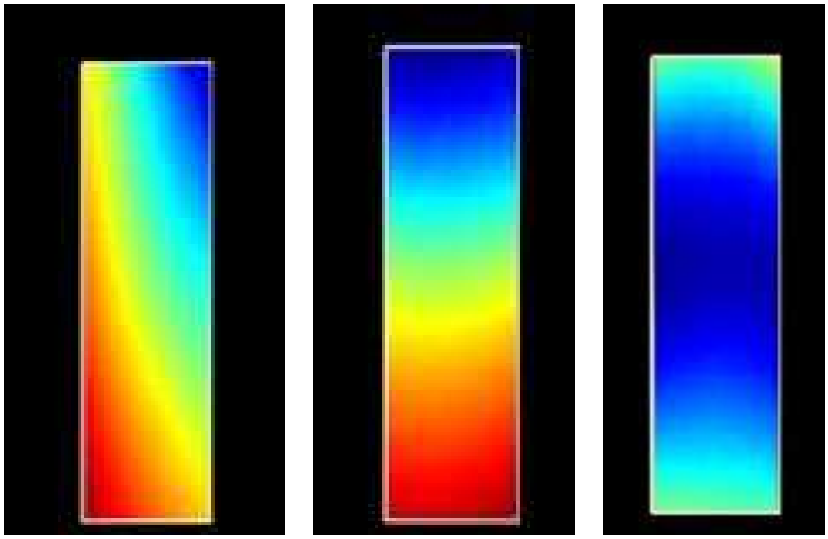


(a) Experimental setup (b) A measurement image of the rail

Figure 7.26: Rail compression test

field show a linear distribution, which also matches the theoretical compression distribution, of the rail for the longitudinal direction and the larger deformation is observed at the bottom than that at the top. Figure 7.27(c) shows the strain field in the longitudinal direction. The longitudinal strain is expected to be a constant value because of the linear displacement deformation. It is seen that the longitudinal strain is a constant value in the majority of the rail with the smaller value on the top and the bottom of the rail. This result verifies the theoretical expectation as well and the discrepancy may be from the imperfect boundary conditions. The longitudinal displacement measurements from the LVDT sensor and the proposed parallel DCT full-field measurement technique are shown in Table 7.4. It is seen that the longitudinal displacement measurements of the proposed parallel DCT full-field measurement technique well match that of the

LVDT sensor and the error is less than 0.2mm.



(a) Displacement u_z (b) Displacement u_x (c) Strain ε_{xx}

Figure 7.27: Rail compression results

Table 7.4: Longitudinal displacement measurements

Force (lbs)	LVDT (mm)	Proposed DCT (mm)
0	0.0000	0.0000
20,000	0.6096	0.6182
40,000	0.9525	0.9685
60,000	1.2573	1.2650
80,000	1.5494	1.5571
100,000	1.8288	1.8361
0	0.0254	0.0340

7.6.2 Outdoor field experiments

This subsection presents an outdoor field experiment for measuring the displacement and strain field of the rail when a train is passing using the proposed parallel DCT full-field measurement technique and the developed hardware system. The location is the Hardy experimental site own by Norfolk Southern, Inc at the Roanoke, VA. Hardy site

has a strain gauge installed on the rail to measure the vertical force with the frequency of 512Hz. The track has a 5.7^o curvature and the high rail is measured by the proposed parallel DCT full-field measurement technique. The developed hardware system is located on the ballast and about 2m away from the rail. The detail experimental parameters are summarized in Table 7.5. Two cameras are utilized to measure the rail in order to show the three dimensional measurement results. Figure 7.28(a) and 7.28(b) shows the experimental setup and the typical captured images (top image from left camera and bottom image from right camera), respectively.

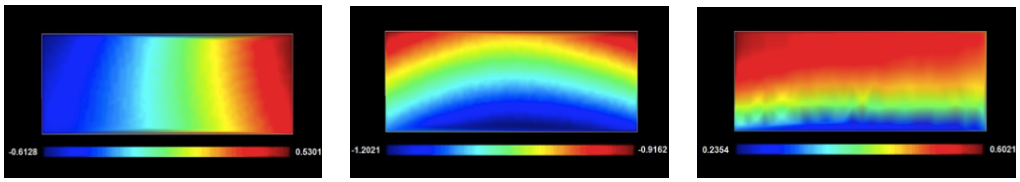
Table 7.5: Experimental parameters for outdoor field experiment

Dot pattern length	24.13cm
Dot pattern height	10.67cm
Distance from rail	203.2cm
Still image resolution	24MP
Video resolution	1080p (2MP)
Video frames per second	30FPS
Train passing speed	30mph

(a) Experimental setup (b) Captured images of the rail

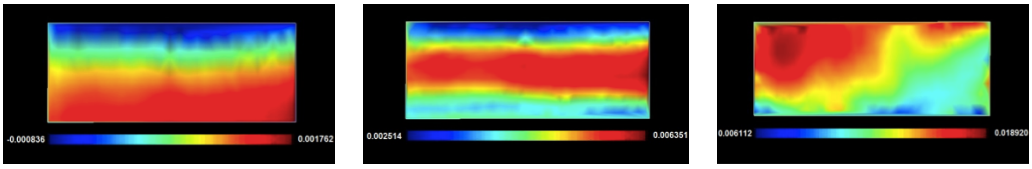
Figure 7.28: Outdoor field test of rail

A set of results are shown in Figure 7.29 and 7.30 for the displacement and strain field measurement respectively and the results corresponds to the captured images



(a) Displacement u_x (b) Displacement u_z (c) Displacement u_y

Figure 7.29: Displacement results of outdoor rail field test



(a) Strain ε_{xx} (b) Strain ε_{zz} (c) Strain ε_{xz}

Figure 7.30: Strain results of outdoor rail field test

(Figure 7.28(b) when the wheel contacts with the rail on the top and slightly left of the marked dot pattern. The longitudinal displacement measurement (Figure 7.29(a)) is seen that the rail is expanded to two sides, which matches the compression distribution in theory. The vertical displacement measurement (Figure 7.29(b)) shows a bending distribution distribution and a vertical layer, the displacement at the bottom is larger than that at the top of the rail. The reason for this vertical layer may be from the torsion led by the lateral force and it compensates some of the deformation at the top of the rail. By observing the lateral displacement measurement (Figure 7.29(c)) it is seen that the rail is pushed out in the lateral direction and the displacement at the top is larger than that at the bottom. Since the bottom of the rail is fixed on the track the above observation is expected. Also the displacement at the left side is seen to be larger than that at the right side and it is because that the wheel contacts with the rail at the location, which is slightly left of the middle of the marked dot pattern.

The longitudinal strain field measurement (Figure 7.30(a)) shows a bending distribution pattern, in which compression and tension appears at the top and the bottom of the rail respectively. The vertical strain field measurement (Figure 7.30(b)) indicates a bending distribution pattern as well with the maximum strain in the middle layer of the rail. The reason for that the maximum strain appears in the middle layer may be

from the torsion of the rail and it compensates the strains on the top as described in the analysis of the vertical displacement measurement. It is seen that the shear strain on the left side is larger than that on the right side (Figure 7.30(c)) and it is expected since the wheel contacts with the rail slight on the left of the marked dot pattern. The displacement and strain field measurements from the proposed parallel DCT full-field measurement technique match the physics with reasonable explanation.

To demonstrate the performance of the proposed parallel DCT full-field measurement technique quantitatively the comparison with the strain gauge measurement attached on the same location of the rail is performed. The strain gauge on the rail measures the vertical force when the train is passing. Figure 7.31 shows a measurement of the vertical forces when a train passes the measured rail. The total time period is 194 seconds and the time period in the red bounding box is measured by the proposed parallel DCT full-field measurement technique. Since the vertical force is expected to have a linear relationship with the vertical displacement. The proposed parallel DCT full-field measurement technique measures the vertical displacement and shows the comparison with the vertical force measurement of the strain gauge in Figure 7.32. The measurements in the purple circles are bad measurements from the blurred captured images due to the fact that the dynamic motion of the rail exceeds the capturing speed of the camera. It is seen that the measurements of the proposed parallel DCT full-field measurement technique in the red circles partially match that of the strain gauge and the total match is about 70 percent.

7.7 Summary

A novel parallel DCT full-field measurement technique for measuring the displacement and strain the deformed surface of a structure has been proposed. The proposed parallel DCT full-field measurement technique identifies and develops the parallel computation in the image analysis and the field estimation processes and then is implemented into the GPU to accelerate the conventional full-field measurement techniques. A detail GPU implementation strategy is also presented. To accommodate indoor or outdoor experimental environments a hardware system, which contains two digital cameras, LED lights and adjustable support legs, is developed. A software package, which implements the proposed parallel DCT full-field measurement technique, and a graphic

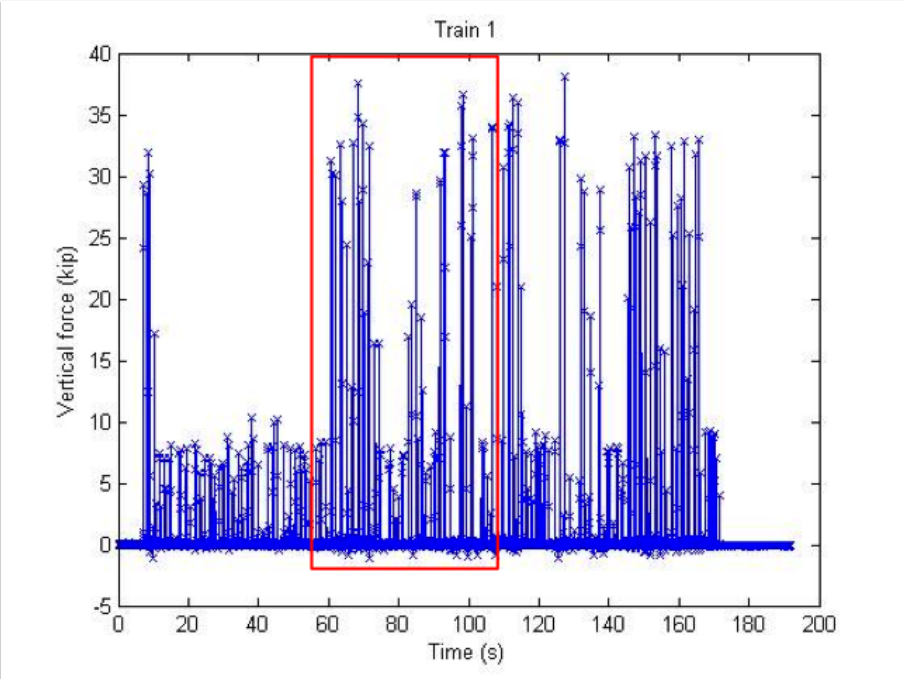


Figure 7.31: Vertical force during train passing

user interface are also developed. The performance and validity of the proposed parallel DCT full-field measurement technique is demonstrated into a series of experiments in the end.

The results of the acceleration of the proposed parallel DCT full-field measurement technique show that the speedup gained is at least 10 and it matches the estimation by the proposed DTFLOP modeling as well. The full-field measurement results of the proposed parallel DCT full-field measurement technique meet the expectation of the physical explanation and match the results of the FEA, the benchmarking analysis for solid mechanics. The experiment of a tensile test on a open-hole aluminum specimen validates the capability of the proposed parallel DCT full-field measurement technique by comparing the experimental results with the theoretical results from the previous analysis. The proposed parallel DCT full-field measurement technique is applied to measure the displacement and strain field in a rail compression test. The results align with the expectation and can be explained reasonable by the physics. The comparison with the LVDT sensor shows error between the longitudinal displacement measurement of the proposed parallel DCT full-field measurement technique and that of the LVDT

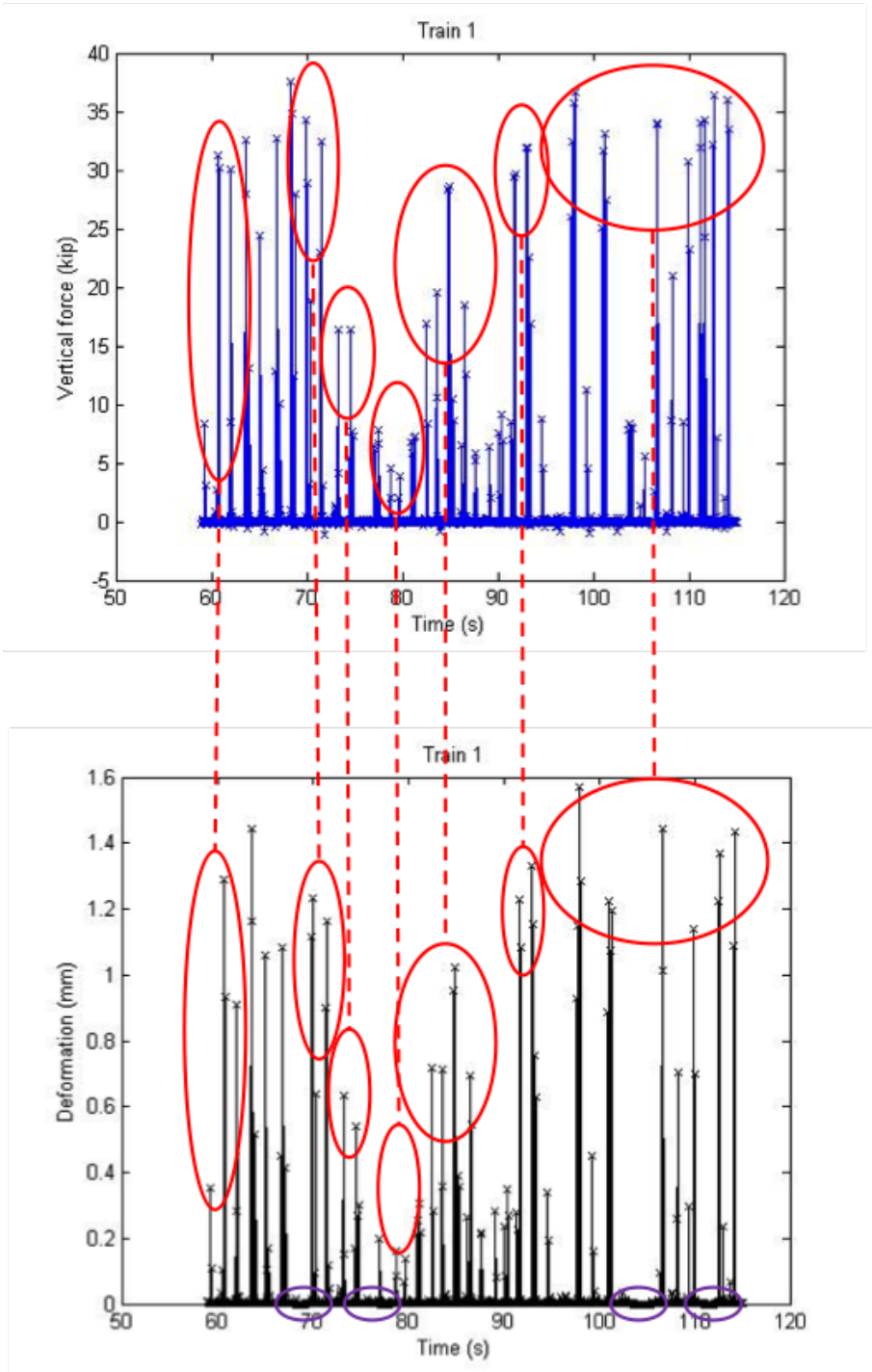


Figure 7.32: Deformation results measured by proposed parallel DCT full-field measurement technique

sensor is less than 0.2mm. The results of the outdoor field experiment for measuring the displacement and strain of the rail when a train is passing on the track also show the explainable results from the physics and the vertical displacement measurements show a 70 percent match with the measurements of the strain gauge, which is attached at the same location as the marked dots on the rail.

Chapter 8

Conclusions and Future Work

8.1 Conclusions

The proposed DTFLOP modeling, which identifies the sequential and the parallel computation, model the data transmission process and the floating point operation process on the CPU and the GPU and derives the formulation to predict the real-time behavior of a dynamic system, has successfully achieved the objective stated at the beginning of this dissertation. The proposed DTFLOP modeling classifies the computation into the sequential computation, which is conducted on the CPU, and the parallel computation, which is performed on the GPU. The proposed DTFLOP modeling formulates the data transmission between the CPU and the GPU using the parameters of the memory access speed and the floating point operations to be carried out on the CPU and the GPU by relating the calculation rate respectively. It is possible to estimate the time cost for computing the model that represents a dynamic system given a certain computer. The proposed DTFLOP modeling can be utilized as a general method to analyze the computation of a model related to a dynamic system. Two real life systems are selected to demonstrate the performance of the proposed DTFLOP modeling, the cooperative autonomous vehicle system and the full-field measurement system, and the related contributions are summarized in the following two sections.

8.1.1 Part 1: Cooperative autonomous vehicle system

The parallel grid-based RBE technique which derives the new formulations and identifies the parallel computation to accelerate the conventional grid-based RBE has been

developed and presented. The belief fusion technique, which fuses not only the observation information but also the target motion information, for the cooperative estimation has been proposed. The proposed DTFLOP modeling is validated using the proposed parallel grid-based RBE technique with the GPU implementation by comparing the estimated time cost with the actual time cost of the parallel grid-based RBE. The superiority of the proposed parallel grid-based RBE technique is investigated by a number of numerical examples in comparison with the conventional grid-based RBE technique.

The numerical example to validate the proposed DTFLOP modeling shows that the estimated error for the time cost of one iteration of the parallel grid-based RBE technique is low in both average and maximum value. The investigation of the time cost of each component modeled in the proposed DTFLOP modeling yields that the time cost for the data transmission dominates the total time cost. The proposed parallel grid-based RBE technique dramatically accelerates the conventional grid-based RBE technique and the real-time performance becomes achievable. Moreover, the speedups gained by each process, the prediction, correction and belief fusion process, of the proposed parallel grid-based RBE technique are evaluated and it is seen that the prediction process achieves the best acceleration performance because of its completely parallelism. The belief fusion technique is examined by a simulated search and rescue test and it is observed to maintain more information of the target compared with the conventional observation fusion technique and eventually leads to the better performance of the target search and rescue.

8.1.2 Part 2: Full-field measurement system

The parallel DCT full-field measurement technique to achieve the full-field measurement of the displacement and strain on the deformed surface of a structure has been presented. The proposed parallel DCT full-field measurement technique measures the displacement and strain field by tracking the centroids of the marked dots on the deformed surface. It identifies and develops the parallel computation in the image analysis and the field estimation processes and then is implemented into the GPU to accelerate the conventional full-field measurement techniques. An efficient way to implement the proposed parallel DCT full-field measurement technique into the GPU is then presented. The corresponding software package, which also includes a graphic user

interface, is developed and described. In order to accommodate indoor or outdoor experimental environments a hardware system, which contains two digital cameras, LED lights and adjustable support legs, has been presented as well. A number of both simulated and real experiments are performed to validate and demonstrate the proposed parallel DCT full-field measurement technique.

The proposed parallel DCT full-field measurement technique accelerates the conventional full-field measurement by an order and the experimental result matches the estimation by the proposed DTFLOP modeling. The theoretical validation experiments of the proposed parallel DCT full-field measurement technique is able meet the expectation of the physical explanation and also match the results of the FEA. To further validate the proposed parallel DCT full-field measurement technique a tensile experiment on an aluminum specimen is performed and its result matches the theoretical analysis. Along with the developed software package and the hardware system, the proposed parallel DCT full-field measurement technique is applied for measuring the displacement and strain field on the surface of the rail. The results of the rail compression test shows a correction compression measurements and the accuracy is demonstrated by comparing with the measurement of the LVDT sensor. In the outdoor field experiment, which measures the displacement and strain fields of the deformed surface of the rail when a train is passing, the proposed parallel DCT full-field measurement technique is able to show the correct results aligned with the physics and the vertical displacement measurement matches the vertical force measurement of the strain gauge attached at the same location of the marked dots on the rail.

8.2 Future work

This dissertation has focused on developing the computer modeling of a typical computer, which contains one CPU and one GPU as the primary computational units. Although the proposed computer modeling is demonstrated by the two systems considered in this dissertation, its validity and performance still needs more experiments to be demonstrated. Also the developed computer modeling can be generalized and extended to include all the processors, e.g. embedded chips on the mobile platform, or consider the processors in a homogeneous way and only discriminate the sequential and parallel computational processes. A more complicated modeling is expected to model the CPU

for its partial parallel computational capability since the modern CPU possibly has up to 8 physical cores. As the trend of the future computing technique more sophisticated parallel techniques are expected to be developed and be implemented based on different processors and the real-time behavior or performance is expected to be well predicted by the generalized modeling.

References

[1] A Ajovalasit, S Barone, and G9 Petrucci. **Towards RGB photoelasticity: full-field automated photoelasticity in white light**. *Experimental Mechanics*, **35**(3):193–200, 1995.

[2] M Akiba, KP Chan, and N Tanno. **Full-field optical coherence tomography by two-dimensional heterodyne detection with a pair of CCD cameras**. *Optics Letters*, **28**(10):816–818, 2003.

[3] and others. **Method and apparatus for non-contact measuring of the deflection of roads or rails**, September 19 2000. US Patent 6,119,353.

[4] K Andersen and R Helsch. **Calculation of grating coordinates using correlation filter techniques**. *Optik*, **80**(76):9, 1988.

[5] NP Andrianopoulos. **Full-Field Displacement Measurement of a Speckle Grid by using a Mesh-Free Deformation Function**. *Strain*, **42**(4):265–271, 2006.

[6] Josef Andrỳsek. **Approximate recursive Bayesian estimation of dynamic probabilistic mixtures**. *Multiple Participant Decision Making*, pages 39–54, 2004.

[7] Amit Apte, Martin Hairer, AM Stuart, and Jochen Voss. **Sampling the posterior: An approach to non-Gaussian data assimilation**. *Physica D: Nonlinear Phenomena*, **230**(1):50–64, 2007. 13

[8] Stéphane Avril, Marc Bonnet, Anne-Sophie Bretelle, Michel Grédiac, François Hild, Patrick Ienny, Félix Latourte, Didier Lemosse, Stéphane Pagano, Emmanuel Pagnacco, et al. **Overview**

of identification methods of mechanical parameters based on full-field measurements. *Experimental Mechanics*, **48**(4):381–402, 2008.

[9] Daniel Balageas, Claus-Peter Fritzen, and Alfredo Güemes. *Structural health monitoring*, **493**. Wiley Online Library, 2006. 14

[10] S Barone, M Berghini, and L Bertini. **Grid pattern for in-plane strain measurements by digital image processing**. *The Journal of Strain Analysis for Engineering Design*, **36**(1):51–59, 2001.

[11] Brian K Bay. **Texture correlation: a method for the measurement of detailed strain distributions within trabecular bone**. *Journal of Orthopaedic Research*, **13**(2):258–267, 1995. 15

[12] Niclas Bergman. **Recursive Bayesian estimation: Navigation and tracking applications. Dissertations no 579**. *Linköping Studies in Science and Technology, SE-581*, **83**, 1999. 13

[13] M Bocciolone, A Caprioli, A Cigada, and A Collina. **A measurement system for quick rail inspection and effective track maintenance strategy**. *Mechanical Systems and Signal Processing*, **21**(3):1242–1254, 2007.

[14] D Bowness, AC Lock, W Powrie, JA Priest, and DJ Richards. **Monitoring the dynamic displacements of railway track**. *Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit*, **221**(1):13–22, 2007.

[15] Timothy Brockett and Yahya Rahmat-Samii. **A novel portable bipolar near-field measurement system for millimeter-wave antennas: construction, development, and verification**. *Antennas and Propagation Magazine, IEEE*, **50**(5):121–130, 2008.

[16] HA Bruck, SR McNeill, M Ae Sutton, and WH Peters Iii. **Digital image correlation using Newton-Raphson method of partial differential correction**. *Experimental Mechanics*, **29**(3):261–267, 1989.

[17] Fu Chang, Chun-Jen Chen, and Chi-Jen Lu. **A linear-time component-labeling algorithm using contour tracing technique**. *computer vision and image understanding*, **93**(2):206–220, 2004.

[18] Peter C Chang, Alison Flatau, and SC Liu. **Review paper: health monitoring of civil infrastructure**. *Structural health monitoring*, **2**(3):257–267, 2003. 14

[19] L Chapman, JE Thornes, and SP White. **Thermal imaging of railways to identify track sections prone to buckling**. *Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit*, **220**(3):317–327, 2006.

[20] DJ Chen and FP Chiang. **Computer-aided speckle interferometry using spectral amplitude fringes**. *Applied optics*, **32**(2):225–236, 1993. 15

[21] Chia Chen Ciang, Jung-Ryul Lee, and Hyung-Joon Bang. **Structural health monitoring for a wind turbine system: a review of damage detection methods**. *Measurement Science and Technology*, **19**(12):122001, 2008. 14

[22] K Diamanti and C Soutis. **Structural health monitoring techniques for aircraft composite structures**. *Progress in Aerospace Sciences*, **46**(8):342–352, 2010. 14

[23] K Dobney, CJ Baker, L Chapman, and AD Quinn. **The future cost to the United Kingdom's railway network of heat-related delays and buckles caused by the predicted increase in high summer temperatures owing to climate change**. *Proceedings of the institution of mechanical engineers, Part F: Journal of rail and rapid transit*, **224**(1):25–34, 2010.

[24] K Dobney, CJ Baker, AD Quinn, and L Chapman. **Quantifying the effects of high summer temperatures due to climate change on buckling and rail related delays in south-east United Kingdom**. *Meteorological Applications*, **16**(2):245–251, 2009.

[25] Prashant Doshi and Piotr J Gmytrasiewicz. **Monte Carlo sampling methods for approximating interactive POMDPs**. *Journal of Artificial Intelligence Research*, **34**(1):297, 2009. 13

[26] Coenraad Esveld. **Modern railway track**. 2001.

[27] Ladislav Fryba. *Dynamics of railway bridges*, **1**. Thomas Telford London, 1996.

[28] Qing Wang Fu-Pen Chiang. **New developments in full field strain measurements using speckles**. *Nontraditional Methods of Sensing Stress, Strain, and Damage in Materials and Structures*, **1318**:156, 1997.

[29] T Furukawa, Xianqiao Tong, G Dissanayake, and HF Durrant-Whyte. **Parallel grid-based method and belief fusionReal-time cooperative non-Gaussian estimation**. In *Industrial and Information Systems (ICIIS), 2011 6th IEEE International Conference on*, pages 370–375. IEEE, 2011.

[30] Tomonari Furukawa, Hugh F Durrant-Whyte, and Benjamin Lavis. **The element-based method-theory and its application to Bayesian search and tracking**. In *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, pages 2807–2812. IEEE, 2007. 13

[31] Tomonari Furukawa, Benjamin Lavis, and Hugh F Durrant-Whyte. **Parallel grid-based recursive Bayesian estimation using GPU for real-time autonomous navigation**. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 316–321. IEEE, 2010. 13

[32] Tomonari Furukawa, Lin Chi Mak, Kunjin Ryu, and Xianqiao Tong. **The platform-and hardware-in-the-loop simulator for multi-robot cooperation**. In *Proceedings of the 10th Performance Metrics for Intelligent Systems Workshop*, pages 347–354. ACM, 2010.

[33] Tomonari Furukawa and John G Michopoulos. **Computational design of multiaxial tests for anisotropic material characterization**. *International journal for numerical methods in engineering*, **74**(12):1872–1895, 2008.

[34] Tomonari Furukawa and John G Michopoulos. **Online planning of multiaxial loading path for elastic material identification**. *Computer Methods in Applied Mechanics and Engineering*, **197**(9):885–901, 2008.

[35] Tomonari Furukawa, John G Michopoulos, and Donald W Kelly. **Elastic characterization of laminated composites based on multiaxial tests**. *Composite Structures*, **86**(1):269–278, 2008.

[36] Tomonari Furukawa, Yoshitaka Wada, John G Michopoulos, and Athanasios Iliopoulos. **Probabilistic Vision-Based Full-Field Displacement and Strain Measurement via Uncertainty Propagation**. In *ASME 2012 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, pages 981–987. American Society of Mechanical Engineers, 2012.

[37] Neil J Gordon, David J Salmond, and Adrian FM Smith. **Novel approach to nonlinear/non-Gaussian Bayesian state estimation**. In *IEE Proceedings F (Radar and Signal Processing)*, **140**, pages 107–113. IET, 1993.

[38] M Grédiac. **Stress analysis and identification with full-field measurements**. *Applied Mechanics and Materials*, **3**:9–16, 2005.

[39] Michel Grediac. **The use of full-field measurement methods in composite material characterization: interest and limitations**. *Composites Part A: applied science and manufacturing*, **35**(7):751–761, 2004.

[40] S Grime and Hugh F Durrant-Whyte. **Data fusion in decentralized sensor networks**. *Control engineering practice*, **2**(5):849–863, 1994. 30

[41] Dong Guo and Xiaodong Wang. **Quasi-Monte Carlo filtering in nonlinear dynamic systems**. *Signal Processing, IEEE Transactions on*, **54**(6):2087–2098, 2006.

[42] Kevin G Harding and James S Harris. **Projection moiré interferometer for vibration analysis**. *Applied Optics*, **22**(6):856–861, 1983. 15

[43] John Harlim and Brian R Hunt. **A non-Gaussian Ensemble Filter for Assimilating Infrequent Noisy Observations**. *Tellus A*, **59**(2):225–237, 2007. 13

[44] H Harrison, T McCanney, and J Cotter. **Recent developments in coefficient of friction measurements at the rail/wheel interface**. *Wear*, **253**(1):114–123, 2002.

[45] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003. 59

[46] Dieter W Heermann. *Computer-Simulation Methods*. Springer, 1990. 11

[47] Mark N Helfrick, Christopher Niezrecki, Peter Avitabile, and Timothy Schmidt. **3D digital image correlation methods for full-field vibration measurement**. *Mechanical Systems and Signal Processing*, **25**(3):917–927, 2011.

[48] Roger W Hockney and James W Eastwood. *Computer simulation using particles*. CRC Press, 1988. 11

[49] Dongliang Huang and Henry Leung. **Maximum likelihood state estimation of semi-Markovian switching system in non-Gaussian measurement noise**. *Aerospace and Electronic Systems, IEEE Transactions on*, **46**(1):133–146, 2010. 13

[50] YY Hung, L Lin, HM Shang, and BG Park. **Practical three-dimensional computer vision techniques for full-field surface measurement**. *Optical Engineering*, **39**(1):143–149, 2000.

[51] AP Iliopoulos and JG Michopoulos. **Effects of anisotropy on the performance sensitivity of the Mesh-Free random grid method for whole field strain measurement**. In *ASME 2009 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, pages 65–74. American Society of Mechanical Engineers, 2009. 15

[52] AP Iliopoulos, JG Michopoulos, and NP Andrianopoulos. **Performance sensitivity analysis of the Mesh-Free Random Grid method for whole field strain measurements**. In *ASME 2008 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, pages 545–555. American Society of Mechanical Engineers, 2008. 15

[53] A Jaffer and S Gupta. **Recursive Bayesian estimation with uncertain observation (Corresp.)**. *Information Theory, IEEE Transactions on*, **17**(5):614–616, 1971.

[54] David V Jáuregui, Kenneth R White, Clinton B Woodward, and Kenneth R Leitch. **Noncontact photogrammetric measurement of vertical bridge deflection**. *Journal of Bridge Engineering*, **8**(4):212–222, 2003.

[55] Robert Jones and Catherine Wykes. *Holographic and speckle interferometry*, **6**. Cambridge university press, 1989. 15

[56] Abdelfateh Kerrouche, J Leighton, WJO Boyle, YM Gebremichael, Tong Sun, Kenneth TV Grattan, and B Taljsten. **Strain measurement on a rail bridge loaded to failure using a fiber Bragg grating-based distributed sensor system**. *Sensors Journal, IEEE*, **8**(12):2059–2065, 2008.

[57] Tariq Khan and Pradeep Ramuhalli. **A recursive Bayesian estimation method for solving electromagnetic nondestructive evaluation inverse problems**. *Magnetics, IEEE Transactions on*, **44**(7):1845–1855, 2008.

[58] J-H Kim, F Pierron, M Grédiac, and MR Wisnom. **A Procedure for Producing Reflective Coatings on Plates to be Used for Full-Field Slope Measurements by a Deflectometry Technique**. *Strain*, **43**(2):138–144, 2007.

[59] Andrew Kish and Dwight W Clark. **Track buckling derailment prevention through risk-based train speed reductions**. In *AREMA annual conference*, pages 20–23, 2009.

[60] KL Knothe and SL Grassie. **Modelling of railway track and vehicle/track interaction at high frequencies**. *Vehicle system dynamics*, **22**(3-4):209–262, 1993.

[61] Vipin Kumar, Ananth Grama, Anshul Gupta, and George Karypis. *Introduction to parallel computing*, **110**. Benjamin/Cummings Redwood City, 1994.

[62] Benjamin Lavis and Tomonari Furukawa. **HyPE: hybrid particle-element approach for recursive bayesian searching-and-tracking**. *Robotics: Science and Systems IV*, page 135, 2009. 13

[63] Benjamin Lavis, Tomonari Furukawa, and Hugh F Durrant Whyte. **Dynamic space reconfiguration for Bayesian search and tracking with moving targets**. *Autonomous Robots*, **24**(4):387–399, 2008. 13

[64] Jun S Liu and Rong Chen. **Sequential Monte Carlo methods for dynamic systems**. *Journal of the American statistical association*, **93**(443):1032–1044, 1998. 11

[65] Rui LIU, Yin-guan WANG, Zhen-yu CHEN, and Yong-pan LI. **Rail stress measurement with critically refracted longitudinal waves [J]**. *Technical Acoustics*, **4**, 2004. 14

[66] Xiao-yong Liu, Qing-chang Tan, and Rong-li Li. **Study on digital image correlation using artificial neural networks for subpixel displacement measurement**. In *Advances in Neural Network Research and Applications*, pages 405–412. Springer, 2010.

[67] YY Lu, T Belytschko, and Lu Gu. **A new implementation of the element free Galerkin method**. *Computer methods in applied mechanics and engineering*, **113**(3):397–414, 1994.

[68] Y Luo. **A model for predicting the effect of temperature force of continuous welded rail track**. *Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit*, **213**(2):117–124, 1999.

[69] Derek LYON. **Dynamic measurements in the research and development of rail vehicles**. *Vehicle System Dynamics*, **16**(3):149–165, 1987.

[70] Alexei Makarenko and Hugh Durrant-Whyte. **Decentralized data fusion and control in active sensor networks**. In *Proceedings of the Seventh International Conference on Information Fusion*, **1**, pages 479–486, 2004. 30

[71] Elias N Malamas, Euripides GM Petrakis, Michalis Zervakis, Laurent Petit, and Jean-Didier Legat. **A survey on industrial vision systems, applications and tools**. *Image and vision computing*, **21**(2):171–188, 2003.

[72] Jan Mandel and Jonathan D Beezley. **An ensemble Kalman-particle predictor-corrector filter for non-Gaussian data assimilation**. In *Computational Science–ICCS 2009*, pages 470–478. Springer, 2009. 13

[73] Jean-Denis Mathias, Xavier Balandraud, and Michel Grédiac. **Experimental investigation of composite patches with a full-field measurement method**. *Composites Part A: Applied Science and Manufacturing*, **37**(2):177–190, 2006.

[74] A McDonach, J McKelvie, P MacKenzie, and CA Walker. **Improved moire interferometry and applications in fracture mechanics, residual stress and damaged composites**. *Experimental Techniques*, **7**(6):20–24, 1983. 15

[75] John G Michopoulos, John C Hermanson, and Tomonari Furukawa. **Towards the robotic characterization of the constitutive response of composite materials**. *Composite Structures*, **86**(1):154–164, 2008. 16

[76] Lewis G Minor and Jack Sklansky. **The detection and segmentation of blobs in infrared images**. *Systems, Man and Cybernetics, IEEE Transactions on*, **11**(3):194–201, 1981. 58

[77] T Nguyen-Thoi, HC Vu-Do, T Rabczuk, and H Nguyen-Xuan. **A node-based smoothed finite element method (NS-FEM) for upper bound solution to visco-elastoplastic analyses of solids using triangular and tetrahedral meshes**. *Computer Methods in Applied Mechanics and Engineering*, **199**(45):3005–3027, 2010.

[78] J-J Orteu, Y Rotrou, T Sentenac, and L Robert. **An innovative method for 3-D shape, strain and temperature full-field measurement using a single type of camera: principle and preliminary results**. *Experimental mechanics*, **48**(2):163–179, 2008.

[79] Joern Pachl. *Railway operation and control*. 2002.

[80] Bing Pan, Anand Asundi, Huimin Xie, and Jianxin Gao. **Digital image correlation using iterative least squares and pointwise least squares for displacement field and strain field measurements**. *Optics and Lasers in Engineering*, **47**(7):865–874, 2009.

[81] Bing Pan, Kemao Qian, Huimin Xie, and Anand Asundi. **Two-dimensional digital image correlation for in-plane displacement and strain measurement: a review**. *Measurement science and technology*, **20**(6):062001, 2009.

[82] Jan Wei Pan, Jin Quan Cheng, and Tomonari Furukawa. **Data fusion of probabilistic full-field measurements for material characterization**. *Key Engineering Materials*, **462**:686–691, 2011. 16

[83] Jaime Peraire, Morgan Vahdati, Ken Morgan, and Olgierd C Zienkiewicz. **Adaptive remeshing for compressible flow computations**. *Journal of computational physics*, **72**(2):449–466, 1987.

[84] Daniel M Popper and Paul B Etzel. **Photometric orbits of seven detached eclipsing binaries**. *The Astronomical Journal*, **86**:102–120, 1981. 2

[85] Michael J Quinn. *Parallel computing: theory and practice*. McGraw-Hill, Inc., 1994.

[86] Nancy Roberts, David F Andersen, Ralph M Deal, Michael S Garet, William A Shaffer, et al. *Introduction to computer simulation: the system dynamics approach*. Addison-Wesley Publishing Company, 1983. 11

[87] Mendel Rosenblum, Stephen A Herrod, Emmett Witchel, and Anoop Gupta. **Complete computer system simulation: The SimOS approach**.

Parallel & Distributed Technology: Systems & Applications, IEEE, **3**(4):34–43, 1995. 12

[88] JF Sanders, LM Smith, FA Spelman, and DJ Warren. **A portable measurement system for prosthetic triaxial force transducers**. *Rehabilitation Engineering, IEEE Transactions on*, **3**(4):366–373, 1995.

[89] M Sanjeev Arulampalam, Simon Maskell, Neil Gordon, and Tim Clapp. **A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking**. *Signal Processing, IEEE Transactions on*, **50**(2):174–188, 2002.

[90] E Savio, Leonardo De Chiffre, and R Schmitt. **Metrology of freeform shaped parts**. *CIRP Annals-Manufacturing Technology*, **56**(2):810–835, 2007. 15

[91] Tyson Schmidt, John Tyson, and Konstantin Galanulis. **Full-field dynamic displacement and strain measurement using advanced 3d image correlation photogrammetry: part 1**. *Experimental Techniques*, **27**(3):47–50, 2003.

[92] Donald S Searle. **Dynamic rail longitudinal stress measuring system**, February 7 1995. US Patent 5,386,727.

[93] J Sevenhuijsen. **Two simple methods for deformation demonstration and measurement**. *Strain*, **17**(1):20–24, 1981. 15

[94] JS Sirkis and TJ Lim. **Displacement and strain measurement with automated grid methods**. *Experimental Mechanics*, **31**(4):382–388, 1991. 15

[95] M Sjödahl and LR Benckert. **Electronic speckle photography: analysis of an algorithm giving the displacement with subpixel accuracy**. *Applied Optics*, **32**(13):2278–2284, 1993. 15

[96] Correlated Solutions. **VIC-3D user manual**. *Columbia, SC: Correlated Solutions*, 2005. 15

[97] Harold W Sorenson. *Kalman filtering: theory and application*, **38**. IEEE press New York, 1985.

[98] John A Stankovic and Krithi Ramamritham. **What is predictability for real-time systems?** *Real-Time Systems*, **2**(4):247–254, 1990.

[99] WJ Staszewski, BC Lee, L Mallet, and F Scarpa. **Structural health monitoring using scanning laser vibrometry: I. Lamb wave sensing**. *Smart Materials and Structures*, **13**(2):251, 2004. 14

[100] Bjoern Stenger, Arasanathan Thayananthan, Philip HS Torr, and Roberto Cipolla. **Filtering using a tree-based estimator**. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 1063–1070. IEEE, 2003. 13

[101] TR Sussman, W Ebersöhn, and ET Selig. **Fundamental nonlinear track load-deflection behavior for condition evaluation**. *Transportation Research Record: Journal of the Transportation Research Board*, **1742**(1):61–67, 2001.

[102] MA Sutton, JL Turner, HA Bruck, and TA Chae. **Full-field representation of discretely sampled surface deformation for displacement and strain analysis**. *Experimental Mechanics*, **31**(2):168–177, 1991.

[103] Kenji Suzuki, Isao Horiba, and Noboru Sugie. **Linear-time connected-component labeling based on sequential local operations**. *Computer Vision and Image Understanding*, **89**(1):1–23, 2003. 62

[104] Albert Tarantola. *Inverse problem theory and methods for model parameter estimation*. siam, 2005. 12

[105] Vikrant Tiwari, Michael A Sutton, SR McNeill, Shaowen Xu, Xiaomin Deng, William L Fourney, and Damien Bretall. **Application of 3D image correlation for full-field transient plate deformation measurements during blast loading**. *International Journal of Impact Engineering*, **36**(6):862–874, 2009.

[106] Wei Tong. **An evaluation of digital image correlation criteria for strain mapping applications**. *Strain*, **41**(4):167–175, 2005.

[107] Xianqiao Tong, Tomonari Furukawa, and Hugh F Durrant-Whyte. **Computational Modeling for Parallel Grid-based Recursive Bayesian Estimation–Parallel Computation Using Graphics Processing Unit–**. *Journal of Uncertainty Analysis and Applications*, **1**(1):15, 2013.

[108] Xianqiao Tong, Tomonari Furukawa, and Saied Taheri. **Speed Enhancement of Displacement and Strain Field Measurement Using Graphics Processing Unit**. In *ASME 2012 Rail Transportation Division Fall Technical Conference*, pages 37–44. American Society of Mechanical Engineers, 2012.

[109] E Toth, A Brath, and A Montanari. **Comparison of short-term rainfall prediction models for real-time flood forecasting**. *Journal of Hydrology*, **239**(1):132–147, 2000. 2

[110] J Tyson, T Schmidt, and K Galanulis. **Biomechanics deformation and strain measurements with 3D image correlation photogrammetry**. *Experimental Techniques*, **26**(5):39–42, 2002. 15

[111] Charles M Vest. **Holographic interferometry**. *New York, John Wiley and Sons, Inc., 1979. 476 p.*, **1**, 1979. 15

[112] Qi Wang and Vincent Hayward. **Compact, portable, modular, high-performance, distributed tactile transducer device based on lateral skin deformation**. In *Haptic Interfaces for Virtual Environment and Teleoperator Systems, 2006 14th Symposium on*, pages 67–72. IEEE, 2006.

[113] Yu Wang and Alberto M Cuitiño. **Full-field measurements of heterogeneous deformation patterns on polymeric foams using digital image correlation**. *International Journal of Solids and Structures*, **39**(13):3777–3796, 2002.

[114] EM Weissman and D Post. **Full-field displacement and strain rosettes by moire interferometry**. *Experimental Mechanics*, **22**(9):324–328, 1982.

[115] Zhengguo Xu, Yindong Ji, and Donghua Zhou. **A new real-time reliability prediction method for dynamic systems based on on-line fault prediction**. *Reliability, IEEE Transactions on*, **58**(3):523–538, 2009.

[116] XF Yao, LB Meng, JC Jin, and HY Yeh. **Full-field deformation measurement of fiber composite pressure vessel using digital speckle correlation method**. *Polymer testing*, **24**(2):245–251, 2005. 15

[117] Bernard P Zeigler, Herbert Praehofer, and Tag Gon Kim. *Theory of modeling and simulation: integrating discrete event and continuous complex dynamic systems*. Academic press, 2000. 11, 12

[118] Peng Zhou and Kenneth E Goodson. **Subpixel displacement and deformation gradient measurement using digital image/speckle correlation (DISC)**. *Optical Engineering*, **40**(8):1613–1620, 2001. 15

Appendix A

User Manual for Proposed Parallel DCT Full-field Measurement Technique

This Appendix describes the procedures to perform a full-field displacement and strain measurement experiment using the proposed parallel DCT full-field measurement technique. A typical example for using the develop GUI to perform the experiment is first described in Section A.1 and then the detail preparation procedures for the proposed parallel DCT full-field measurement technique is explained in Section A.2

A.1 A typical example

The developed GUI for the proposed parallel DCT full-field measurement technique is able to connect up to 4 cameras and visualize the full-field displacement and strain on the deformed surface in real time. Especially for the low-quality CPU computer system, GPU mode is more desired to achieve real-time performance. A typical example is described in the following:

1. Select the desired devices type, image resolution and number of cameras;
2. Choose between online or offline measurement mode;
3. Load a predefined dot pattern for the testing specimen;
4. Load a predefined configuration file if applicable and then skip Step 5 and 6;

5. Choose ROI on the captured images and adjust the proper image processing parameters for each camera;

6. Choose the desired probabilistic data fusion method, computation mode (CPU/GPU), desired interpolation method and save all the parameters to a configuration file;

7. Click the initialization button to pass the user-defined parameters and options to the system and then click start button to start real-time full-field measurement and corresponding three dimensional visualization is shown in the Plot areas.

8. After the analysis is complete, results can be exported for future analysis.

It is noted that the user is able to observe the real-time full-field measurement results through the three dimensional visualization in the Plot areas when software is running. At the same time the user is also free to switch between the different visualization options.

A.2 Preparation procedures

This section describes the preparation procedures to perform a full-field measurement experiments using the proposed parallel DCT full-field measurement technique and the developed software. It includes the specimen preparation, lamps and lamp setting and camera setting and calibration.

A.2.1 Specimen Preparation

A.2.1.1 Specimen Marking

The specimen needs to be marked so that the full-field displacement and strain can be measured. Suggested for a planar specimen is to print marks on an adhesive sheet such as an adhesive label and stick it to the specimen. As far as it is made of a material which elongates well, this approach can capture the linear and nonlinear material behavior until a crack starts to grow. Remarks to be noted in this approach are:

- The adhesive sheet must be white in color;

- The surface of the adhesive sheet should not be shiny to avoid light reflection;

- The adhesive sheet must not be a paper but be plastic or plastic-like such that it elongates well;
- The adhesive sheet must be able to print well;
- The adhesive should be able to firmly attach the sheet to the specimen.

If the surface of the specimen is uneven, e.g. the surface of the rail, it is desired to directly paint the dots on the measured surface. It can be either a permanent marker pens or spray paints. The color needs to be as contrast as possible with the color of the measured surface.

A.2.1.2 Marked dots

Figure A.1 shows an example of the marked dots on an open-hole specimen. As shown in the figure, the marked dots may have different size and be distributed with different density. The rules to follow are:

- With no overlap, make the marked dots as large as possible. The larger the dots, the more accurate the measurement;
- Increate the density of dots in the areas that may see large change of strain or strain concentration.

Obviously, the size of dots in high density areas becomes smaller than that of dots in low density areas. Since the smallest dot exhibits the worst accuracy, the dot size needs to be controlled in case that the minimum accuracy is specified. The printer can be laser, ink-jet or anything else. Resolution with 600 dpi or higher is desirable and sufficient for full-field strain measurement.

A.2.1.3 Sticking Adhesive Sheet

Careless sticking may result in creating an uneven surface with bubbles and/or wrinkles. Sticking may start at the center and then gradually and slowly continue outwards.

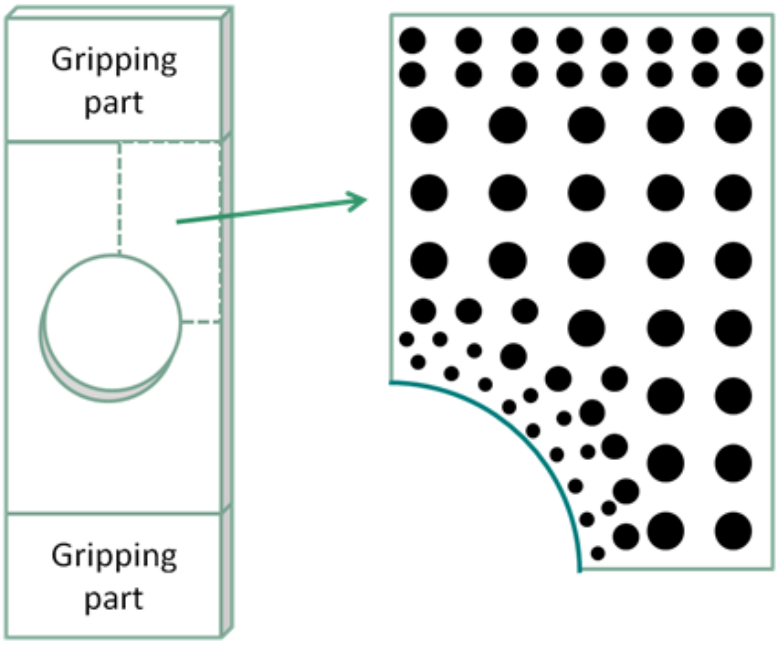


Figure A.1: Marked dots on an open-hole specimen

A.2.2 Lamps and Lamp Settings

A.2.2.1 Lamps to Select

The lamp suggested for the proposed parallel DCT full-field measurement technique is that with light-emitting diode (LED). Light with AC waveform flickers and thus makes the measurement accuracy inconsistent and bad, so it is not suited for the full-field displacement and strain measurements. Multiple lamps are necessary if a single lamp cannot provide bright and uniformly distributed light to the specimen. The lamp system is thus to be made such that it provides light:

1. As bright as possible;
2. As equally distributed as possible.

A.2.2.2 Lamps Settings

Since the light should not flicker the lamp stands should be heavy and rigid enough to provide light uniformly. Lamps should also be attached to the lamp stand firmly. The procedure for setting lamps is as follows:

1. Fix the specimen with white surface to the testing machine;
2. Make the light of the lamps brightest (Figure A.2);
3. Start the camera software;
4. Control the camera aperture mechanically such that the surface of the specimen exhibits some darkness with the brightness around 128 out of 255 (Figure A.3);
5. Relocate the lamps until the surface sees the uniform darkness.

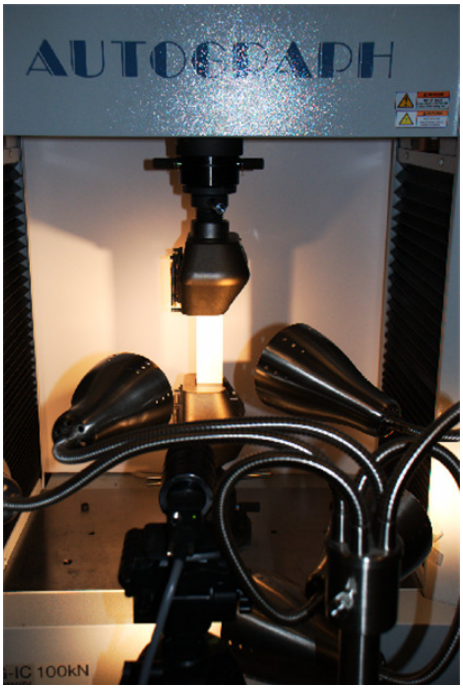


Figure A.2: Specimen with brightest light

A.2.3 Cameras, Camera Settings and Calibration

A.2.3.1 Cameras to Select

The type of cameras is most popularly classified in terms of the type of image sensor; either CMOS or CCD. Most generally, CMOS image sensors are the technology of the choice for high-volume, space-constrained applications where image quality requirements are low. CCD image sensors, on the other hand, offer superior image quality

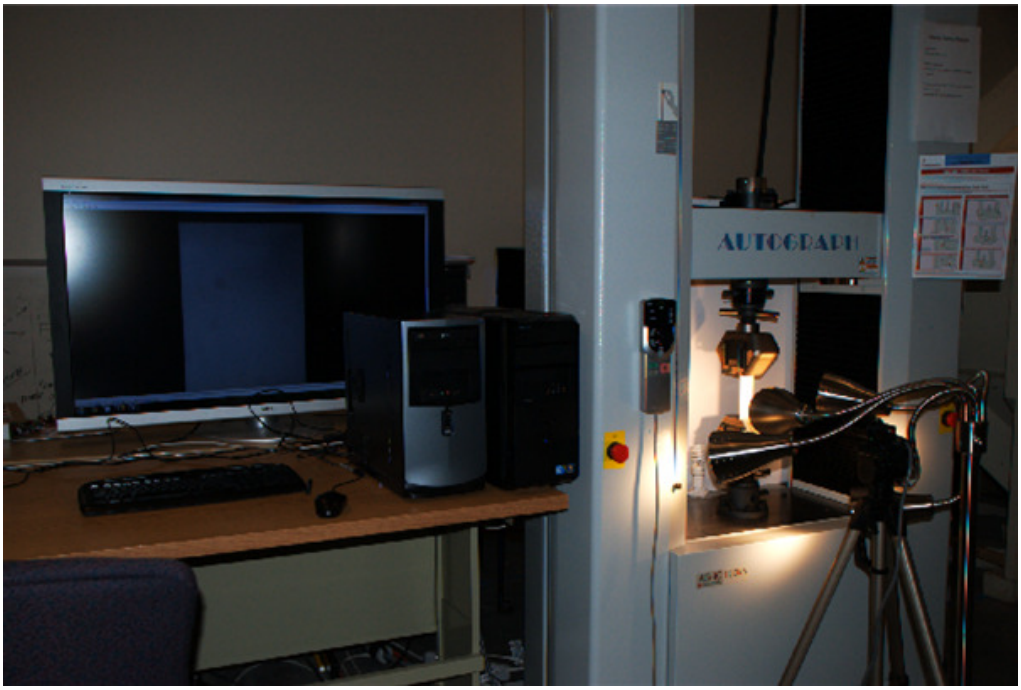


Figure A.3: Specimen with some darkness

and flexibility at the expense of system size. As a result, CMOS image sensors see a natural fit for security cameras, PC videoconferencing, wireless handheld device, bar-code scanners, fax machines, consumer scanners, toys, biometrics and some automotive vehicle uses whereas CCD image sensors remain the most suitable technology for high-end imaging applications, such as digital photography, broadcast television, high-performance industrial imaging, and most scientific and medical applications. Recent years have however started to observe smaller differences, and our comparative studies experimentally held indeed support this. Camera suggestions are not therefore made based on the type of the image sensor but on the other elements.

Two types of camera suggested for the proposed parallel DCT full-field measurement technique are are the medium-level digital cameras for industrial use and single-lens digital cameras for personal use. While industrial cameras and personal cameras use CCD and CMOS image sensors respectively for the above reasons, suggestions do not consider the type of image sensor but take the following advantages of each camera into account:

- Industrial cameras

1. No force of motion in shuttering;
2. Regular shape in compact size;
3. Fast data transmission.

- Personal cameras

1. High resolution (up to 15MP);
2. Low cost (less than $600).

These advantages contribute to either the measurement accuracy or the cost effectiveness. As seen in the list of advantages both types of camera possess advantages that enhance measurement accuracy. As a result, single-lens personal cameras may be chosen if the budget is tight. However, the disadvantages exist conversely:

- Industrial cameras

1. Low resolution (up to 5MP);
2. Cost (more than $2000).

- Personal cameras

1. Considerable force of inertia in shuttering;
2. Irregular shape in large size;
3. Low data transmission.

Personal cameras mechanically close a shutter, and the force of inertia caused by this may lead to the movement of their positions. In order to use personal cameras it is important that the cameras are firmly fixed to the fixture and that the fixture is rigid and firmly fixed to the ground.

A.2.3.2 Camera Settings

Camera settings should take the following procedures:

1. Locate cameras to make sure the area of the marked dots is inside the field of view of all the cameras;
2. Set the aperture mechanically as low as possible;

3. On the camera software, set the shutter speed such that

   (a) The white background has a brightness of 255 (In Figure A.4, it is shown that the RGB values on white background is [255, 255, 255]);

   (b) The black dot has the minimum brightness (In Figure A.5, it is shown that the RGB values on a black dot is [32, 32, 32]).

Setting the aperture of the camera to the lowest value makes the specimen subject to the brightest light. This primarily has the advantage of removing random noise created by the image sensor, but setting a high shutter speed resultantly by this further allows the process fast. Figure A.6 shows the effect of the camera setting as well as the lamp

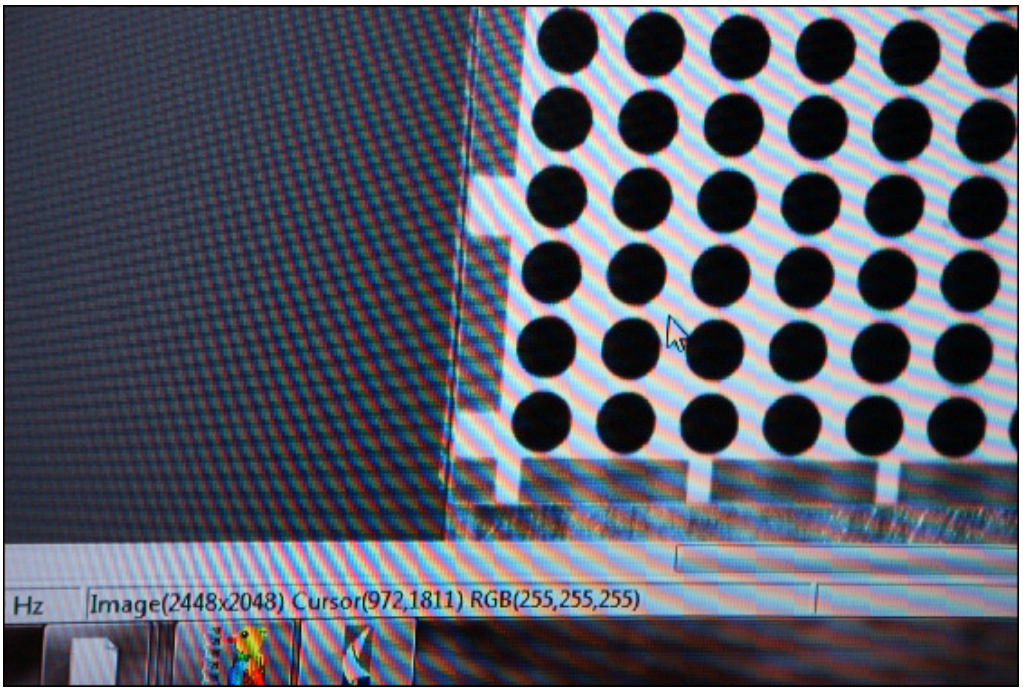


Figure A.4: Shutter speed adjustment (white background)

setting where the settings create the finest image for the proposed parallel DCT full-field measurement technique. Although pixels in the black dot and the white background ideally should have the brightness of 0 and 255, the actual captured images do not achieve this regardless of the light condition. The excessive light reduces the darkness of the black dot whereas the insufficient light makes the white background dark. By

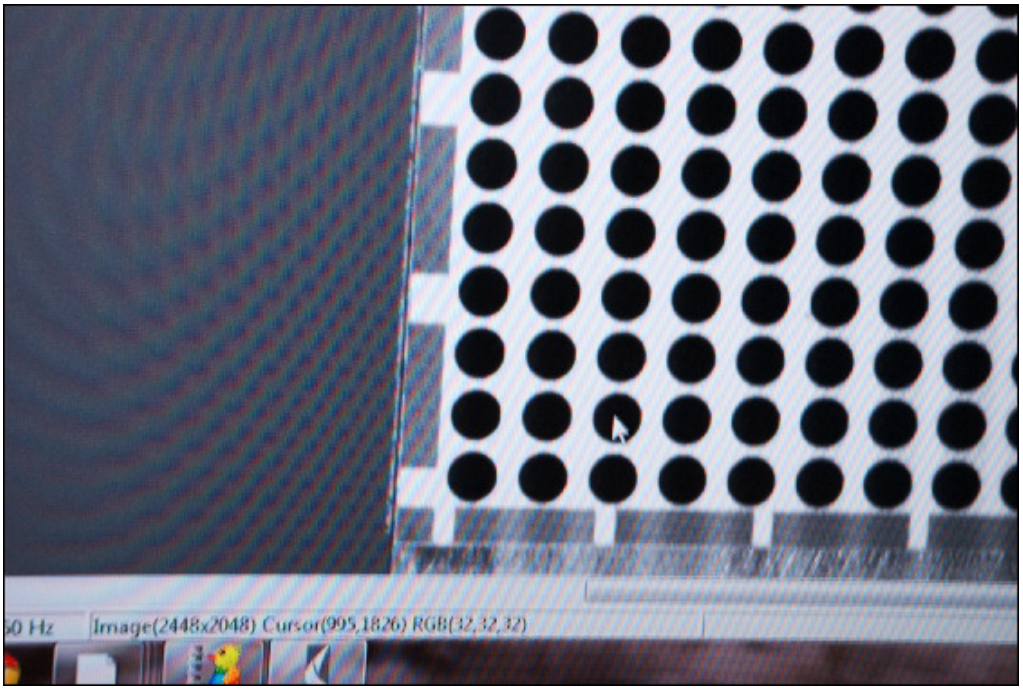


Figure A.5: Shutter speed adjustment (black dots)

setting the shutter speed as instructed above, the histogram of the captured image most resembles that of the true image.

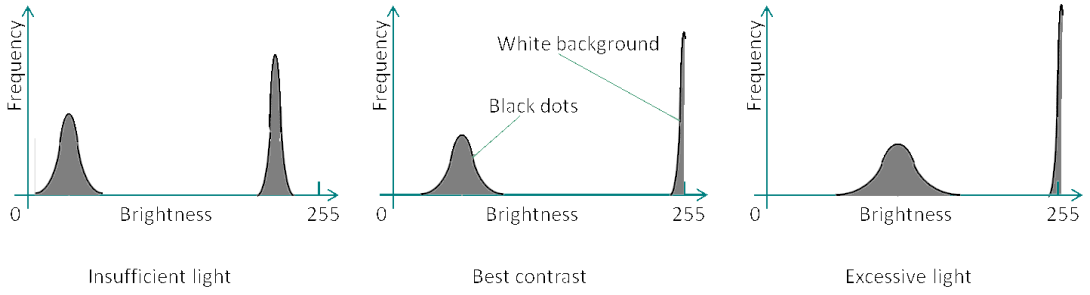


Figure A.6: Histogram

The experimental studies have shown that the best condition is when the distribution of the white background is about to reach zero in the 8bit level. In this condition, none of the darkness information is lost while the measured white background is close to the true white color. Achieving the condition by controlling the camera parameters is found to be twice as good as achieving the condition by controlling the light brightness

in the measurement accuracy.

A.2.3.3 Camera Calibration

Cameras must be calibrated in order to extract the intrinsic parameters of the camera, including the parameters for focal length, principal point, skew coefficient and distortions. Since the proposed parallel DCT full-field displacement and strain measurement technique is highly dependent on the surrounding environment of the experiments, the camera calibration needs to be thus performed at a static position where the camera is fixed in the experiments. Such a process is performed either using a checkerboard or a dot pattern calibration board with an open-source calibration toolbox. The developed software in this dissertation utilizes the openCV library to perform the camera calibration process.

Camera calibration should take the following procedure:

1. Prepare a checkerboard or a dot pattern calibration board as shown in Figure A.7;
2. Use the fixed camera to capture at least 10 images with different rotations and positions;
3. Extract the all the features from all the images;
4. Identify the camera intrinsic parameters through nonlinear optimization.

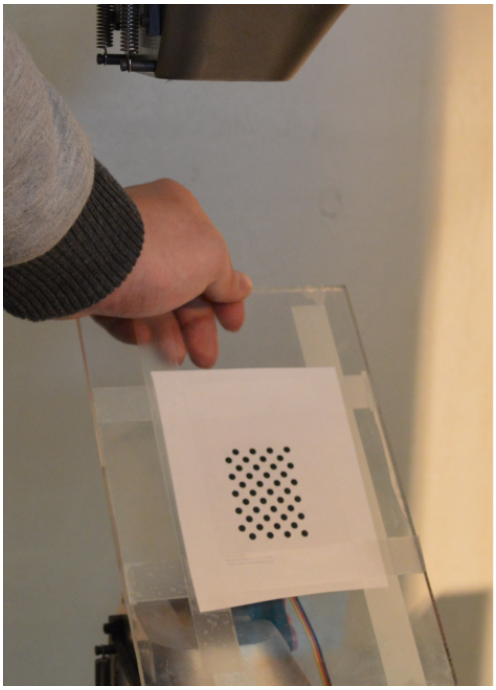


Figure A.7: A typical dot pattern calibration board