VISION-BASED MOTION TRACKING FOR RISK-ASSESSMENT DURING SEISMIC EVENTS

T. C. Hutchinson
F. Kuester
K.-U. Doerr
D. Lim
University of California, Irvine

N. Kurato
H. Ukon
M. Oshio
A. Kondo
Kajima Corporation

CUREE-Kajima Joint Research Program
Phase V

August 2004
This page left intentionally blank.
Vision-Based Motion Tracking for Risk Assessment During Seismic Events

Tara C. Hutchinson¹, Falko Kuester, Kai-Uwe Doerr and David Lim²
University of California, Irvine
August 2004

CUREE-Kajima Phase V Report

¹Assistant Professor, Dept. of Civil and Environmental Engineering
²Assistant Professor, Post-Doctoral Researcher, and Undergraduate Researcher, respectively. Dept. of Electrical Engineering and Computer Science
Executive Summary

Previous experiences during earthquake events emphasize the need for new technologies for real-time monitoring and assessment of facilities with high value nonstructural elements such as equipment or other contents. Moreover, there is substantial limitation in our ability to rapidly evaluation and identify potential hazard zones within a structure, exposing rescue workers, society and the environment to unnecessary risks. A real-time monitoring system, integrated with critical warning systems, would allow for improved channelling of resources. The optimal system would allow for non-intrusive, high-speed data, which is rapidly collected via a fast network and analyzed in a central repository for building owners and rescue workers to make informed decisions. In recognition of these issues, the primary objective of this project is to summarize investigative work on the development, testing, and evaluation of vision-based motion tracking systems for use in monitoring and assessing seismic risk. Two fundamentally different types of vision-based systems are considered: (i) digital red light-based cameras and (ii) digital fully image-based cameras. Shake table experiments are used to test and evaluate the software and hardware infrastructure needed for such a system.

This report is subdivided into chapters detailing the results from investigation of (i) background and literature related to motion tracking using camera hardware and image processing algorithms, (ii) implementation and results of light-based systems and (iii) implementation and results of image-based systems.
Acknowledgements

Support for this work was provided under a joint program between the Consortium of Universities for Research in Earthquake Engineering (CUREE) and Kajima Corporation (CUREE-Kajima Phase V). Helpful comments and suggestions were provided by Dr. Katsuhisa Kanda our program manager, Dr. Narito Kurato, Mr. Hachiro Ukon, Mr. Makoto Oshio, and Mr. Aikido Kondo, each of our project team with Kajima Corporation. We appreciate the support of the U.S. CUREE oversight program managers, Professors Bill Iwan and Haresh Shah. Experimental studies described in Chapter 3 were supported by the Earthquake Engineering Research Centers Program of the National Science Foundation, under Award Number EEC-9701568 through the Pacific Earthquake Engineering Research Center (PEER). Assistance was provided by Mr. Bob Kazanjy, Development Engineer, and other research staff of the Structural Engineering Test Hall (SETH) and in the Visualization and Interactive Systems (VIS) Group at the University of California, Irvine. UC Irvine doctoral students Samit Ray Chaudhuri, Maria-Cruz Villa-Uriol and Mark Phair, contributed to work described in Chapter 3, Appendix A, and Chapter 5, respectively. The above financial and other support is greatly appreciated.
# Contents

Executive Summary ................................................................. i  
Acknowledgements ............................................................... ii  

1 Introduction and Background 1  
1.1 Scope of this Report ......................................................... 2  

2 Related Work and Motivation for using Vision-Based Sensors 5  
2.1 Application of Vision-Based Sensors in Other Fields ..................... 5  
  2.1.1 Highway System Monitoring ........................................... 6  
  2.1.2 Train and Railway Related Systems .................................. 7  
  2.1.3 Environmental Monitoring ............................................ 7  
2.2 Risk Assessment ............................................................... 8  
  2.2.1 Seismic Risk Assessment ............................................... 8  
  2.2.2 Potential for Vision-Based Sensors in Seismic Risk Assessment ..... 9  
  2.2.3 Analysis Example: Before and After .................................. 12  
  2.2.4 Analysis Example: Time-Varying ..................................... 13  
2.3 Summary Remarks ........................................................... 14  

3 Light-Based Vision Systems: Overview and Validation 17  
3.1 Background ........................................................................ 17  
3.2 Application of Light-Based Tracking ...................................... 18  
3.3 Methodology Employed in this Work .................................... 19  
  3.3.1 Camera Calibration ........................................................ 20  
  3.3.2 Feature Detection .......................................................... 21  
3.4 Three-Dimensional (3D) Reconstruction .................................. 21  
3.5 Shake Table Test Program ................................................... 23  
  3.5.1 Experimental Setup ........................................................ 23  
  3.5.2 Selection of Earthquake Input Motions ............................... 25  
  3.5.3 Instrumentation - Conventional and Light-Based ..................... 26  
3.6 Experimental Results .......................................................... 28  
  3.6.1 Translational Motion ....................................................... 28  
  3.6.2 Three-Dimensional Motion ............................................. 29  
  3.6.3 Glassware Motion .......................................................... 32  
3.7 Summary Remarks ............................................................. 35  

4 Image-Based System: Hardware and Software Design 37  
4.1 Image-based System: Hardware Design ................................... 37  
  4.1.1 Camera Array Design ...................................................... 37  
  4.1.1.1 Survey of Available Camera Technologies ....................... 38  
  4.1.2 Computational Platform and Data Flow .............................. 40  
4.2 Image-Based System: Software Design ................................... 43
4.2.1 Image Capture and Acquisition .............................................. 44
  4.2.1.1 Capture Process ....................................................... 44
  4.2.1.2 SceneIdentifier Graphical User Interface (SIGUI) ................. 45
  4.2.1.3 Camera View and File Management in SIGUI ........................ 47

4.3 Acquisition System Performance Evaluation ................................. 47
  4.3.1 Results and Discussion .................................................. 49

4.4 Summary Remarks .................................................................. 49

5 Image-Based System: Processing and Data Analysis ......................... 53
  5.1 Pixel-Based Image Processing Chain (PIPC) ............................... 53
    5.1.1 Camera Calibration and Image Correction ............................. 54
    5.1.2 Defining Plausible Features for Tracking using an Object Mask Method .............................. 55
      5.1.2.1 Background Detection and Extraction ......................... 56
      5.1.2.2 Background Detection: Threshold Definition ................ 57
      5.1.2.3 Background Detection: Snap Shot Method .................... 58
      5.1.2.4 Background Detection: Running Average Method ........... 60
      5.1.2.5 Background Detection: Mean-Standard Deviation Method .... 62
    5.1.3 Object Mask Calculation ................................................. 65
    5.1.4 Feature Detection ......................................................... 68
      5.1.4.1 Edge Detection Enhancement ..................................... 69
    5.1.5 2D Feature Tracking ...................................................... 73
      5.1.5.1 2D Tracking Sample Results .................................... 74
  5.2 Resolution of the Processed Data .......................................... 74
    5.2.1 Hardware-Limited Resolution ......................................... 76
    5.2.2 Software (Algorithm)-Limited Resolution ............................ 77
  5.3 Shaking Camera Issue and Correction Patterns ............................ 79
  5.4 Poor Features to Track ....................................................... 81
    5.4.1 Oscillating Lighting on a Feature .................................... 81
    5.4.2 Occlusion of a Feature ................................................ 81
  5.5 Summary Remarks .................................................................. 82

6 Conclusions and Recommendations for Future Work ......................... 91
  6.1 Conclusive Findings .......................................................... 91
  6.2 Recommendations for Future Work ........................................ 92

A Image-Based System: Calibration ............................................... 99
  A.1 Motivation and Related Work ............................................... 99
  A.2 Terminology ................................................................... 100
  A.3 Classification of Calibration Methods .................................. 102
  A.4 Techniques and Steps Involved ............................................ 103
    A.4.1 Image Acquisition System Design .............................. 104
    A.4.2 Selection of Calibration Patterns ................................. 105
  A.5 Common Calibration Packages .............................................. 106
    A.5.1 Static Pattern Calibration ............................................ 106
    A.5.2 Dynamic Calibration - Single Camera ............................ 107
    A.5.3 Multi-Camera Calibration ............................................. 108
    A.5.4 Case Study of a Calibration Tool .................................. 108
  A.6 Summary Remarks ............................................................ 110
List of Figures

1.1 Example of damaged nonstructural contents from the 1994 Northridge Earthquake: (a) Veteran’s Administration Sepulveda hospital and (b) Pharmacy at the Northridge medical arts building (courtesy of (NISEE)) .................................................. 2
1.2 Traditional instrumentation wiring arrangement at a multiplexing board. ..................... 3

2.1 Cars tracked by RoadWatch (image courtesy of David Beymer (Beymer et al., 1997)). ...... 6
2.2 Bridge monitoring using cameras and conventional sensors (Commodore Barry Bridge, monitoring system by Aktan et al. (2002)) (Schematic courtesy of E. Aktan). ......................... 7
2.3 Growing ice at the shore highlighted by an edge detection algorithm (image courtesy of USGS (1991)). ................................................................. 8
2.4 Post-1994 Northridge earthquake photographs: (a) broken sprinkler in the Olive View hospital and (b) sprinkler pipe leakage (after an aftershock)(Courtesy NISEE, University of California, Berkeley). ......................................................... 10
2.5 Ruptured pipe of the supply line to a 3000 liter storage tank in the roof top of mechanical room – 2001 Nisqually earthquake (Filiatrault et al., 2001). ........................................ 11
2.6 Ceiling of the eastern 4-story unit of the Instito Politecnico National – 1957 Mexico City earthquake (Courtesy NISEE, University of California, Berkeley). ...................... 11
2.7 Light fixture and ceiling failure inside the Medical Treatment Building in Sylmar, California after the 1971 San Fernando earthquake (Courtesy NISEE, University of California, Berkeley). 12
2.8 Glass container hitting a bench surface during a shake table experiment: (a) original image and (b) detection of the broken glassware using an edge detection algorithm. ................. 13
2.9 Object tracking and detection from digital streamline images taken during shake table experiments. Current position shown in solid red, tracked (previous) position shown in dashed red. (Digital movie courtesy of K. Mosalam, UC Berkeley). ................................. 14
2.10 Chemicals observed by an vision sensor during a shake table experiment: (a), (b) and (c) original images and (d), (e), and (f) images with only edges identified. The tracked object is circled. Both, single shots of the video stream and detected edges are shown in subsequent order. ................................................................. 15

3.1 Schematic illustrations of image capture concepts: (a) multiple camera correlation and (b) epipolar constraint. ................................................................. 20
3.2 Examples of good and poor ’blob’ detection and correlation: (a) poor spacing selection (too closely spaced), (b) limited resolution, (c) high light intensity and/or poor camera sensitivity, and (d) best correlation. ................................................................. 22
3.3 Spherical marker identification - example of a 3D point cloud in a scene (shake table configuration 1A). ................................................................. 23
3.4 Schematic layout of the mock-laboratory including bench and shelving details: (a) configuration one and (b) configuration two. ........................................ 24
3.5 Photographs showing experimental configuration one - mock-laboratory environment: (a) no scene illumination and (b) scene illuminated. .......................................................... 25

3.6 Photographs showing detail of retro-reflective passive markers used at shelf level to monitor chemical glassware and textbook movements: (a) no scene illumination and (b) scene illuminated (G#1 denotes glassware #1). .................................................. 26

3.7 Isometric view of three-dimensional geometric model of the mock-laboratory environment (configuration one) - showing layout of high-speed CCD cameras surrounding viewing volume. .................................................. 28

3.8 Comparison of LVDT and the light-based motion measurements in the x-direction at the base of the shake table for the 1989 Loma Prieta, Corralitos input motion (GM6). (top: comparison with mean of multiple markers, bottom: comparison with mean \( \pm \sigma \) response). .................................................. 30

3.9 Characteristics of the input time histories GM-6 (1989 Loma Prieta, Corralitos) and GM-10 (2000 Tottori, Japan, Kofu): (a,d) acceleration, (b,e) velocity and (c,f) displacement. .................................................. 32

3.10 Three-dimensional response of a small bifocal microscope - subjected to the 2000 Tottori, Japan, Kofu input motion (GM-10) in configuration one: (a) location of passive markers, (b) photograph of final position at end of motion, (c) displacement of bench and relative x-direction displacement time history of the three passive markers placed on the microscope, and (d) resulting rotational time history about the z-axis. .................................................. 33

3.11 Response of a small 38 cm (15" Standard) diagonal CRT computer monitor - subjected to two repeated trials of the 1989 Loma Prieta Corralitos input motion (GM-6) in configuration one: (a) photograph of toppled monitor (Trial #2) at end of motion, (b) rotational time history about the y-axis and (c) rotational time history about the z-axis for Trial #1 and #2. .................................................. 34

3.12 Response of large chemical glassware elements mounted on shelf - subjected to the 2000 Tottori, Japan, Kofu input motion (GM-10) in configuration one: (a) photograph at end of simulation of G#2 and G#3 and (b) relative x-direction displacement time history (Glassware #1, #2, #3, and #4 are placed per the illustration in Figure 3.6). .................................................. 35

4.1 Interline transfer of charge packets of photodiodes laterally and then vertically (into register) on the CCD chip (schematic courtesy of the Eastman Kodak Company (2003)). .................................................. 38

4.2 Photographs of the Basler A301fc camera: (a) with pen for scale and (b) with mounting wings ready for assembly. .................................................. 40

4.3 Color sensitivity on the CCD chip (upon placement of the Bayer filter) for the Basler A301fc cameras. .................................................. 41

4.4 Capturing and storing data using the image-based system: (left) photographs of cameras located on the shake table and (right) data flow inside the computer system after capturing. .................................................. 42

4.5 SceneIdentifier’s unique multi-threaded capture and storing architecture. .................................................. 45

4.6 Snapshot of the SceneIdentifier graphical user interface (SIGUI): (a) overall view and (b) camera controls. .................................................. 46

4.7 Snapshot of the SceneIdentifier data view mode from SIGUI. .................................................. 48

4.8 Sample performance data for one camera with a shutterValue = 250. .................................................. 50

4.9 Performance data summary plots for shutterValue = 250 for: (a) one camera, (b) two cameras, (c) three cameras, and (d) four cameras. .................................................. 51

4.10 Performance data summary plots for shutterValue = 500 for: (a) one camera, (b) two cameras, (c) three cameras, and (d) four cameras. .................................................. 52

5.1 Pixel-based image processing chain (PIPC) flowchart. .................................................. 54

5.2 Camera Model .................................................. 55

5.3 Image correction example: (a) image with lens distortion, (b) corrected image, and (c) correction model. .................................................. 56
5.4 Baseline image sequence for evaluating background detection methods (picture on a wooden desk, white background with holes): (a) Case I: static test environment, (b) Case II: moving background (note directional vectors at holes behind object), and (c) Case III: lighting/shadow effects (note shadow at right upper corner). ........................................ 57
5.5 Example of sensitivity to threshold settings: (a) image of interest (tracking frame with two objects), (b) identified object mask \((\text{thr}=60)\), and (c) identified object mask \((\text{thr}=30)\). .......................... 58
5.6 Background detection using the snap shot method, Case I, images with \(\text{thr}=60\): (a) mask, (b) edges defined by Canny, and (c) contours. ........................................ 59
5.7 Background detection using the snap shot method, Case II, images with \(\text{thr}=60\): (a) mask, (b) edges defined by Canny, and (c) contours. ........................................ 59
5.8 Background detection using the snap shot method, Case III, images with \(\text{thr}=60\): (a) mask, (b) edges defined by Canny, and (c) contours. ........................................ 60
5.9 Background detection using the running average method, Case II, images with \(\text{thr}=60\), frames = 10, \(\alpha = 0.05\): (a) mask, (b) edges defined by Canny, and (c) contours. ........................................ 61
5.10 Background detection using the running average method, Case II, images with \(\text{thr}=60\), frames = 30, \(\alpha = 0.05\): (a) mask, (b) edges defined by Canny, and (c) contours. ........................................ 61
5.11 Background detection using the running average method, Case II, images with \(\text{thr}=60\), frames = 10, \(\alpha = 0.50\): (a) mask, (b) edges defined by Canny, and (c) contours. ........................................ 62
5.12 Image capturing under neon light conditions (60 Hz lighting): (a) image acquisition at 40 Hz (shutter at 500) and (b) image acquisition at 80 Hz (shutter continuous at 584). .......................... 63
5.13 Background detection using the running average method, Case III, images with \(\text{thr}=30\), frames = 30, \(\alpha = 0.05\): (a) mask, (b) edges defined by Canny, and (c) contours. ........................................ 64
5.14 Background detection using the mean-standard deviation method, Case I, images with \(\text{thr}=60\), frames = 10, \(3\sigma\): (a) mask, (b) edges defined by Canny, and (c) contours. ........................................ 64
5.15 Background detection using the mean-standard deviation method, Case I, images with \(\text{thr}=60\), frames = 10, \(3\sigma\), \(\text{co} = 15\): (a) mask, (b) edges defined by Canny, and (c) contours. ........................................ 65
5.16 Background detection using the mean-standard deviation method, Case II, images with \(\text{thr}=60\), frames = 10, \(3\sigma\), \(\text{co} = 15\): (a) mask, (b) edges defined by Canny, and (c) contours. ........................................ 66
5.17 Background detection using the mean-standard deviation method, Case II (vertical movement), images with \(\text{thr}=60\), frames = 30, \(6\sigma\), \(\text{co} = 15\): (a) mask, (b) edges defined by Canny, and (c) contours. ........................................ 66
5.18 Background detection using the mean-standard deviation method, Case III, images with \(\text{thr}=30\), frames = 30, \(3\sigma\), \(\text{co} = 15\): (a) mask, (b) edges defined by Canny, and (c) contours. ........................................ 66
5.19 Mask calculation flowchart. ........................................ 67
5.20 Example of an applied contour filter (sliding computer): (a) input image and (b) image after contour filter is applied. ........................................ 68
5.21 Feature detection flowchart. ........................................ 68
5.22 Example of the good feature algorithm by Shi and Tomasi (1994): (a) all features (identified with crosses), (b) computed object mask, and (c) selected features (based on object mask). ........................................ 69
5.23 Example of the manual ROI selections and masking applied to these ROI to identify good features of interest: (a) original image, (b) all features (identified with crosses), (c) multiple ROI selected by user (identified with red boxes), and (d) selected features within object mask. ........................................ 70
5.24 Canny algorithm: (a) without and (b) with background subtraction \((T_{\text{low}}=30\) and \(T_{\text{high}}=100\) in both cases). ........................................ 72
5.25 Canny algorithm used with background subtraction thresholds set at: (a) \(\text{thr}=60\) and (b) \(\text{thr}=200\) \((T_{\text{low}}=10\) and \(T_{\text{high}}=30\) in both cases). ........................................ 72
5.26 Geometric basis for sub-pixel accuracy calculation implemented. ........................................ 74
5.27 Optical flow calculation, results for: (a) frame 2 and (b) frame 50. ........................................ 75
5.28 Displacement time history of identified features: (a) x-direction and (b) y-direction. ........................................ 75
5.29 2-dimensional (x-y) history of identified features. ........................................ 76
5.30 Simple camera-object setup to determine viewing angle. ........................................ 77
5.31 Theoretical resolution as a function of object distance. ........................................ 78
5.32 Photograph of selected pattern feature for evaluating tracking accuracy of the optical flow algorithm. ............................................................... 78
5.33 Plan and elevation schematic of scene used to evaluate the tracking accuracy of the optical flow algorithm. ......................................................... 83
5.34 Software-limited resolution sensitivity study: time history of (a) x-direction and (b) y-direction pixel data, during static collection. ...................... 84
5.35 Evaluated test patterns. .......................................................................................... 84
5.36 Pattern layout recommended for shaking camera correction. ............................... 85
5.37 Example of poor features to track (glassware in changing lighting conditions): (a) at \( t=0 \) seconds and (b) \( t=10 \) seconds. ...................................................... 86
5.38 Example of poor features to track (glassware in changing lighting conditions): (a) x-direction and (b) y-direction time histories. ................................. 87
5.39 Feature tracking result using four cameras at: (row 1) frame 0, (row 2) frame 400, (row 3) frame 800 and (row 4) frame 1200. .......................................... 88
5.40 Four camera comparison over 1200 frames: (a) x-displacement and (b) y-displacement. .................. 89

A.1 Fundamental terminology in world and camera coordinate systems. ....................... 101
A.2 Classification of calibration methods. ................................................................. 102
A.3 Calibration steps for single and multiple camera configurations. ......................... 104
A.4 Marker-based (a, b) and Feature (c, d) tracking. ............................................. 105
A.5 Pattern-based calibration using cubes as reference objects. ................................ 106
A.6 Different views of the spatial configuration for an \( n \) cameras setup. ...................... 112
A.7 Computed errors for a 12 cameras setup. ....................................................... 113
A.8 Computed mean errors for all tested setups. .................................................... 113
List of Tables

3.1 Details of the equipment and contents tested in configurations one and two. .................. 26
3.2 Summary of earthquake motions used for base excitation input to the different bench-shelf systems. .......................................................... 27
3.3 Summary statistics comparing LVDT and light-based measurements - earthquake input motions at the base of the shake table. ........................................ 31

4.1 Comparison of specifications and cost of different available camera models. ................. 39
4.2 Common PCI buses and their associated speed and bandwidth. ................................. 42
4.3 Summary of capture performance experiments considering multiple camera configurations and different shutter exposure times. ................................. 50

5.1 Parameters to calculate theoretical resolution \( r_{estheor} \). ........................................ 76
5.2 Resolution for camera lens combination used in this work. ...................................... 77
5.3 Software-limited resolution sensitivity study: parameters used for optical flow accuracy calculation. .......................................................... 79
5.4 Software-limited resolution sensitivity study: calculated resolution in the orthogonal plane. 79
Chapter 1

Introduction and Background

The overall performance of buildings, bridges, and other civil infrastructure subjected to natural hazards such as earthquakes, wind, etc., has dramatically improved over the years due to an increased understanding of both component and system behavior. Although the primary components of a structure may perform fairly well, it is now well recognized that extensive nonstructural damage may occur, resulting in potential life safety threats and significant economic losses. For equipment and building contents, the primary economic losses may be accrued due damage and subsequent need for repair or cleanup, resulting in prolonged downtime. Of particular concern are nonstructural elements within buildings that contain hazardous materials. During post-earthquake response, damage to these elements can create difficulties with assessing the status of the interior of the building structure. For example, during the 1994 Northridge earthquake, 387 hazardous material incidents were identified (EERI, 1995) and nearly 60% of these incidents occurred inside laboratories, resulting in an estimated $1.5 million in clean-up expenses. During this event, these incidents caused subsequent interior building damage before rescue crews could identify hazardous areas within the building. At Cal-State University, Northridge, for example, three separate fires ignited due to hazardous material spills and subsequently destroyed nine science laboratories. Extensive nonstructural and associated secondary damage were observed as a result of this event. The Northridge earthquake claimed 33 lives with estimated economic losses totalling $50 billion (CUREE, 1998). Photographs from this event illustrate the types of interior building damage, which contributed to business disruptions and downtime, resulting in substantial secondary damage (Figure 1.1).

These and many other past experiences during earthquake events have emphasized the need for new technologies for real-time monitoring and assessment of facilities with high value nonstructural elements such as equipment or other contents. Moreover, there is substantial limitation in our ability to rapidly evaluate and identify potential hazard zones within a structure, exposing rescue workers, society and the
environment to unnecessary risks. A real-time monitoring system, integrated with critical warning systems, would allow for improved channelling of resources. In recognition of these issues, we have been investigating vision-based systems, specifically digital red-light cameras and high speed charged-couple-device (CCD) cameras, and developing a methodology for using these advanced monitoring techniques for risk assessment during earthquake events. We describe these as two fundamentally different types of systems: (i) light-based systems and (ii) image-based systems. Our approach consists of developing a non-intrusive, high-resolution, high-speed network of these two types of vision-based systems and corresponding detection, tracking and analysis algorithms for monitoring equipment and contents in buildings. Shake table experiments are being used evaluate the approach. These experiments also use conventional (wired) transducers, thereby providing a unique reference dataset for evaluating and refining the vision-based approach.

A distinct advantage of using such an approach may be clearly seen by observing Figure 1.2. In the laboratory or in the field, traditional sensors must be discretely attached (and most often cable-based as shown in Figure 1.2), require long set-up times, and for small/light elements, modify the properties of the physical system (e.g. change the mass or stiffness). In contrast, using images as the sensor adds no mass or stiffness to the system, offers high-resolution and speeds, while directly providing digital information.

1.1 Scope of this Report

This report is organized into the following chapters: (i) introduction and motivation (ii) related work and motivation, (iii) light-based system: overview and validation, (iv) image-based system: hardware and software
1.1. SCOPE OF THIS REPORT

Figure 1.2: Traditional instrumentation wiring arrangement at a multiplexing board.

design, (v) image-based system: processing and data analysis and (vi) conclusions and recommendations. Appendix A summarizes calibration studies conducted for this study.
Chapter 2

Related Work and Motivation for using Vision-Based Sensors

Vision (camera-based) technologies can greatly assist in risk assessment in many realms of science and engineering. These sensors have great potential when monitoring the movement of different types of systems at a variety of dimensional scales. A challenge put forth by vision-based sensor networks is that large amounts of data are being acquired at high rates, posing a need for efficient processing, analysis, and data storage algorithms as well as hardware to support these computational demands. Due to the readily available high-speed computational platforms today, significant amounts of data can be processed in near-real time. These computational advantages, combined with an increased availability of high-quality vision-based sensors have promoted their use in many practical field applications.

In this chapter, an overview of vision-based sensors used in various field applications is provided, with a focus in the area of civil infrastructure monitoring. Since vision-based systems can be particularly powerful for monitoring both local and global movements within an environment, hazards such as earthquakes are a natural application area for this new technology. Based on examples from past earthquakes, potential benefits of these systems are described and simple examples of coarse image analysis approaches presented in support of possible deployment scenarios.

2.1 Application of Vision-Based Sensors in Other Fields

The increasing speed and resolution, combined with dramatically decreasing cost of vision-based sensors in recent years has resulted in an increased usage of these systems in practical field applications. Common field examples can be broadly categorized as either (i) observation or (ii) surveillance. Application areas include observing sensitive public areas for security reasons, monitoring traffic on bridges and highways, inspecting
critical systems, or detecting changes in the environment over time. Select examples from each of these are provided in the following sections.

2.1.1 Highway System Monitoring

Highway systems adopted vision-based sensors for monitoring traffic in the early 1990’s. Work in this area has focused on the detection of unauthorized use of roadways and the recognition of special traffic situations, such as excessive congestion. Several publications are available describing vehicle tracking implementations (e.g. Bell et al. (1999); Irani and Anandan (1998); Koller et al. (1994)). One example is the RoadWatch project described by Beymer et al. (1997), which can count the number of vehicles on a highway and detect volume and location of traffic congestions. Figure 2.1 shows an example of several cars being tracked by this system.

Similarly, vision-based sensors have been used to monitor bridge structures. For example, in the work by Aktan et al. (2002), streamed digital video images provide a mechanism to monitor the traffic moving over critical areas of the bridge. Combined with other types of measurement devices such as accelerometers, mechanical displacement and force transducers, each strategically located on the structure, it is possible to identify the cause and effect relationship for significant loading situations, such as trucks travelling over the structure. The bottom half of Figure 2.2 illustrates the synchronization of images and vibration data used to identify the response of a heavy truck crossing the bridge.

Figure 2.1: Cars tracked by RoadWatch (image courtesy of David Beymer (Beymer et al., 1997)).
2.1. APPLICATION OF VISION-BASED SENSORS IN OTHER FIELDS

2.1.2 Train and Railway Related Systems

Vision sensors have also been used in areas such as inspection of various train and railway related systems (e.g. Mair and Fararooy (1998)). Applications exist that inspect the rail profile (e.g. Magnus (1995); Bachinsky (1995)), rail gap (Sasama et al., 1991), contact wire positions (Ostermeyer, 1983), or wear (Van Gigch et al., 1991). Additionally, the train itself may be monitored or inspected, including the inspection of wheels or the thickness of the brake pads. During operation, vision sensors may also detect locking wheels or overheating brake systems. To observe the maximum occupancy inside a train, vision sensors have also been used to count the number of passengers (Zhang and Sexton, 1995; Khoudaour et al., 1996; Ottonello et al., 1992).

2.1.3 Environmental Monitoring

Environmental monitoring includes applications in shoreline and coastal bluff observation, based on images collected from continuously streaming cameras monitoring, for example, the water-beach interface (USGS, 1991). In this example, a system was developed that is able to monitor breaking waves, alongshore currents, rip currents, and beach-face profiling, especially in stormy weather. Figure 2.3 illustrates an example of...
image data captured during a cold winter season, where the red line outlines the ice-shoreline interface.

![Image of ice-shoreline interface](image.png)

**Figure 2.3:** Growing ice at the shore highlighted by an edge detection algorithm (image courtesy of USGS (1991)).

### 2.2 Risk Assessment

Many definitions are available to quantify the term risk assessment. Generally, risk is defined as the uncertainty of a specific occurrence times the consequence resulting from this occurrence. Uncertainty must be determined for each risk individually. This can be done using mathematical models based on probability estimates. Consequence is most often defined as the financial measure of the effect of the occurrence. One must distinguish between risks that affect people and risks that affect only material (objects). Financial consequence affecting material (objects) tend to be fairly challenging to estimate, whereas risk associated with people, such as fatalities, can be defined with a greater level of certainty. With humans involvement, one can more accurately define risk as the hazard severity level times the likelihood of occurrence times the number of persons exposed to the risk (Campbell, 1998).

#### 2.2.1 Seismic Risk Assessment

In the context of seismic risk assessment, it is often thought of as the identification of the risk of damage incurred during a particular earthquake event. An alternative measure may be the number of people injured or fatalities caused by the specific event. To assess this type of risk, the likelihood of occurrence and the hazard severity level must be determined in the same fashion as conventional risk assessment. In the context of earthquake hazards, direct impacts, such as damage to the primary structure have to be considered in combination with post-earthquake hazards, such as chemical spills or broken pipes. Post-event hazards may
often result in substantial damage to the primary structure, which could be avoided using the appropriate assessment tools.

2.2.2 Potential for Vision-Based Sensors in Seismic Risk Assessment

In this section, we identify specific areas and issues where vision-based technologies and associated image processing algorithms can assist in assessing the risk due to earthquake hazards, specifically secondary (post-earthquake) hazards created within the interior of building structures. In this case, one can differentiate between damage that is associated with elements directly attached to the structure, such as connected piping networks or attached equipment, and elements not attached to the structure, such as building contents or unattached equipment. The following discussion provides specific examples from past earthquakes with reference to both attached and unattached elements.

After analyzing the damage resulting from the 1971 San Fernando earthquake, the State of California enacted the California Hospital Act. The objective of this act was to provide an enhanced level of design and construction to improve the resistance of hospitals in California. The 1994 Northridge earthquake illustrated that structural systems used for the construction of hospitals built after 1971, considering the new California Hospital Act, performed very well. However, the piping and air handling systems in many hospitals suffered fractures along individual pipes or at joints, resulting in the temporarily closure of many hospital buildings (Todd et al., 1994). Figure 2.4(a) and (b) show examples of sprinkler pipe leakages. The photograph in part (a), shows a sprinkler inside the Olive View hospital at the ceiling level, while part (b) shows leakage on the outside of a building near Olive View Hospital resulting from a sprinkler system damaged during an aftershock. To ensure that the fire prevention system of a building is still functional after an earthquake, guidelines provided by the Federal Emergency Management Agency (FEMA, 2000) point to improved standards and the need for flexibly mounted piping. However, older structures may not conform to current understanding of earthquake demands, resulting in failure of sprinkler systems. Newer structures being built according to modern design standards, such as the Uniform Building Code (UBC, 1997) or International Building Code (IBC, 2003), meet these requirements.

During the 1994 Northridge earthquake, 2500 water heaters were damaged and subsequently introduced wide spread natural gas leaks. By optically monitoring and correlating measured movements to tolerable limits, the potential for post-earthquake fires in this situation would be greatly reduced. Similarly, the 2001 Nisqually earthquake also resulted in widespread nonstructural damage. In one example, a water pipe ruptured in the mechanical room on the roof of a hotel causing 3000 liters of water in a storage tank to
flood several floors (Filiatrault et al., 2001). Figure 2.5 shows the ruptured pipe and the storage tank. The proposed system of vision-based sensors would have been able to detect the spilling water and could have triggered the shut down of the piping system to prevent further damage to the building.

Vision-based sensors may also reveal the occupancy in public areas and subsequently aid in determining if large equipment has toppled or slid and buried occupants. This would be particularly helpful in guiding earthquake reconnaissance teams to areas of distress while removing the need to survey unoccupied areas. Figure 2.6 shows an example of a failed ceiling element that may have buried several occupants. This photograph was taken at the Instituto Politecnico National in Mexico City after the Mexico earthquake in 1957. A vision-based sensor could have provided information about the number of occupants in the area now covered by the ceiling element.

Using vision-based sensors, a blocked escape route can be detected and occupants advised to use another escape route. Although the Federal Emergency Management Agency (FEMA, 2000), for instance, recommends that certain materials such as hollow clay or unreinforced masonry should not be used around stairs,
2.2. RISK ASSESSMENT

Figure 2.5: Ruptured pipe of the supply line to a 3000 liter storage tank in the roof top of mechanical room – 2001 Nisqually earthquake (Filiatrault et al., 2001).

Figure 2.6: Ceiling of the eastern 4-story unit of the Instito Politecnico National – 1957 Mexico City earthquake (Courtesy NISEE, University of California, Berkeley).

elevators, and corridors to keep escape routes clear, it is well known that these materials are used in such areas. Figure 2.7 shows an example for a hallway, which, due to cracked walls, broken light fixtures and ceiling elements, may not be usable as an escape route.

Telecommunication is another important area where vision-based sensors may assist in evaluating and minimizing seismic risk. Due to the strong linkage with today’s economy, a loss of communication services can be extremely costly on both a local and national scale. Consequently, backup systems are generally provided for added redundancy and, depending on the type of services provided, the backup system is either (i) activated after failure of the main system is detected or (ii) operated in parallel. Vision-based sensor networks can assist in identifying physical failures within the infrastructure of the telecommunication system, and this information may then be provided to an early warning system to prevent cascading affects that may disable the entire system. For example, if one monitors a communications rack with hundreds of cables.
connected to it, the movement of this rack can be traced and compared against a predefined threshold. If movement exceeds predetermined limits, a warning can be issued to maintenance staff and cabling and connectors surveyed for potential damage.

Another important example is related to potentially hazardous areas, such as biological or chemical science laboratories. These laboratories often contain hazardous chemicals unrestrained and mounted on shelves or bench tops. Spillage of these chemicals may result in subsequent hazardous areas within the structure. To further exasperate this, fire may also result where these chemicals are exposed to each other or combined with leaking gas or other fluids. Optical sensors can assist in detecting these types of situations and thus reduce hazards by turning off gas or other piping systems or warning people to prevent them from entering such hazardous areas. This provides a means to inform emergency personnel about critical situations, allowing them to be more attentive to primary hazard zones.

### 2.2.3 Analysis Example: Before and After

Perhaps the simplest technique for using image data is based on an overall change detection approach, whereby images from before and after an event (called bi-temporal images) are obtained and used to identify elements that have changed within the original image (before). As an example, Figure 2.8(a) shows one frame of a video stream captured during a shake table experiment, where a glass container that was initially sliding towards the front of the shelf, topples and finally shatters on the table top. Using a simple edge algorithm applied to this image, and comparing it with pre-event images, the remains of the container can be clearly seen in Figure 2.8(b). When combined with additional pre- and post-processing techniques, additional details can be extracted. For example, by subtracting a reference image depicting the initial arrangement from the
2.2.3 Analysis Example: Time-Varying

A continuous monitoring approach using edge detection algorithms is also able to reveal the situation described in Section 2.2.3. In this case, the path of the container can be tracked and its changing physical shape observed. Tracing the object in time, the individual parts of the broken glass container can still be identified [Figure 2.8(b)]. Consequently either using a pair of before-after images, or an image sequence, it is possible to determine that this container is broken and if the type of material stored in this container is known, a hazard warning can be issued by the system automatically pinpointing that a potential risk exists.

Figure 2.9(a)–(d) shows a sequence of images captured during a shake table experiment at the University of California, Berkeley. In this example, a retail store is reconstructed on a shake table and the contents of the scene observed. During simulated earthquake motions, many components, including a heavy box, fell from one of the shelves onto the floor. These changes in the environment can be detected and highlighted. In this case, rectangles show object movement over time, annotated with a vector indicating the primary direction of movement.

In contrast to Figure 2.9, Figure 2.10(a)–(c) depicts only a selected region-of-interest (ROI) within an environment. The vision sensor is directed towards a portion of a shelf where several glass containers, which
may contain potentially hazardous chemicals, are resting. During shaking, one of the containers topples and spills some of its contents. The upper images show single shots of the captured video stream, and the monitored container is circled in the images. To track the container, an edge detection algorithm was used, resulting in images such as the ones shown in Figure 2.10(d)–(f). Similar to the example shown in Figure 2.8, the algorithm by Canny (1986) is used to extract object features (feature edges) that are easier to identify.

If at least two cameras are capturing the same viewing area, a three-dimensional model can be computed from the images if no occlusion occurs. This assumes that the vision-based sensors are properly synchronized, resulting in images being captured at exactly the same time or that precise time stamps are available for each image to enable post-recording synchronization.

### 2.3 Summary Remarks

In this chapter, we provide a survey of related work in the area of vision-based sensors used for (i) observation, (ii) surveillance, and (iii) monitoring of civil infrastructure. The motivation regarding vision-based sensors and their use in reducing earthquake hazards (particularly secondary hazards) by increasing our assessment ability are discussed. Select simple examples of coarse image analysis before-after and time-varying approaches are presented in support of possible deployments in the area of earthquake risk assessment. This chapter presents our motivation to develop tools for earthquake risk assessment using vision-based sensors.
2.3. SUMMARY REMARKS

Figure 2.10: Chemicals observed by an vision sensor during a shake table experiment: (a), (b) and (c) original images and (d), (e), and (f) images with only edges identified. The tracked object is circled. Both, single shots of the video stream and detected edges are shown in subsequent order.
Chapter 3

Light-Based Vision Systems: Overview and Validation

In this chapter, the methodology of light-based motion tracking is applied to the measurement of the three-dimensional motions of various types of equipment and building contents commonly found in biological and chemical science laboratories. The system is comprised of six high-speed, high-resolution charge-coupled-device (CCD) cameras outfitted with a cluster of red-light emitting diodes (LEDs). Retro-reflective (passive) spherical markers discretely located in a scene are tracked in time and used to describe the behavior of various types of equipment and contents subjected to a range of earthquake motions. Results from this study show that the non-intrusive, light-based approach is extremely promising in terms of its ability to capture relative displacements in three orthogonal directions and complementary rotations.

3.1 Background

Alternative methods of tracking seismic motions are desirable, particularly in the laboratory setting where scale models are often used to study earthquake response. Many fields, besides earthquake engineering, require precise high-speed motion tracking and thus new technologies are rapidly becoming available. Biomechanics, human gait analysis, robotics, virtual reality (VR), gaming and even entertainment have successfully employed a variety of new motion tracking techniques. Typically these applications require accurate six degree-of-freedom tracking, expressed by position \((x, y, z)\) and orientation \((\text{yaw}, \text{pitch}, \text{roll})\). In earthquake engineering, generally the three positional degrees-of-freedom are the most important motions to track and often only one of these dominates depending upon the problem. However, in some cases, oddly shaped elements, with non-uniform mass distribution will tend to rotate even upon uni-axial input motion. Depending

\(^1\)Results described here are presented in the citation Hutchinson et al. (2004)
on the particular environment constraints and the configuration of the system to be tracked, the required accuracy can range from millimeters to fractions thereof. Moreover, high spatial and temporal resolution (high sampling rates), large working volumes, and limited instrumentation time are also desirable. Perhaps the most difficult problem faced in earthquake engineering experimental research is the cumbersome physical attachment and associated lengthy set-up times of most conventional motion sensors. For reduced scale experiments, these sensors generally add substantial mass or stiffness and therefore change the response characteristics of the system.

3.2 Application of Light-Based Tracking

Early and very extensive usage of optical tracking technologies can be traced to robotics and subsequently to the medical field for biomechanics and human motion studies with a focus on gait analysis (e.g. Cappozzo (1984); Heyn et al. (1996); Kidder et al. (1996); Sampath et al. (1998); Hansen et al. (2002)). The ability to carefully analyze and characterize a patient’s manner and rate of movement (gait) has greatly aided in the development of treatment options for the physically handicapped. For these applications, requirements of very precise measurements, extended workspaces and limited physical constraints are similar to those faced in many civil engineering applications.

Image-based systems, using analog or digital camera technologies, have been successfully developed and used for a variety of civil engineering applications. For example, Moon and Bernold (1997) used an image-based approach for controlling a robotic paint removal system applied to bridge structures. This system required robot joint positional information, which could be collected from images captured using a moderate speed analog video camera and frame grabber. The pronounced physical constraints and associated difficult field working conditions makes the image-based approach in this context extremely appealing. Papanicolaou et al. (1999) describe an image-based system for use in detecting the movement of sediments in a stream. This system was used in conjunction with a laser Doppler velocimeter to study the effects of turbulence on sediment transport. In a similar application, Gustafsson and Gustafsson (1995) use image analysis techniques to study particle movement by introducing and monitoring passive markers traveling through a flow pattern. In a recent application to structural measurements, Fu and Moosa (2002) use a single monochrome CCD camera for detailed monitoring of the external deformations of a simply supported beam.

Successful application to damage detection and structural health monitoring was demonstrated using this single CCD camera approach. In the context of structural vibration, Shinozuka et al. (2001) developed a
proof-of-concept system using a single color CCD camera operating at 30 fps (frames per second). These and other image-based approaches applied to civil engineering applications use single cameras coupled with frame grabbers and digital image processing techniques to identify 2-dimensional scene anomalies specific to the application problem. Without reducing the spectrum of captured light or changing the light characteristics (e.g. using structured light), intensive manual labor or sophisticated edge detection techniques must be applied on a per image basis. Furthermore, capturing the full spectrum of light at resolutions valuable for structural monitoring coupled with high frequency motions (such as earthquake loading) results in extremely large volumes of data generated.

3.3 Methodology Employed in this Work

Reducing the spectrum of light captured by a camera system is often termed light-based capture. Light-based capture systems are based on an emitter/receiver scenario, whereby a scene is illuminated and only a narrow band of wavelengths of light reflected or emitted from the scene is captured. Elements (markers) within the scene can be either active or passive, that is, emit or reflect a bounded wavelength of light. Active markers generally utilize light emitting diodes (LEDs), mounted statically at specific locations in the test space and a set of CCD cameras that are responsible for recording the created reference pattern. Given the exact spacing and position of a set of reference LEDs, the relative position of the CCDs can then be computed and this information subsequently used to track arbitrary markers within the scene. The LEDs in this case require local cabling to provide power for the diode. Alternatively, externally illuminated retro-reflective markers (passive markers) can be used. This eliminates external power source requirements and associated cabling in the scene of interest. The idea of bounding the wavelength of capture has significant implications with respect to the amount of data that has to be acquired. For example, in an image-based system consisting of mega-pixel (1024x1024) cameras with 24bit color resolution per pixel, operating at 120 frames per second, three gigabit of data would be acquired per second per camera. When data is only recorded for a specific wavelength or a range thereof, data complexity and volume can be reduced significantly.

Computational Approach to Multi-Camera Capture

In this work, multiple high-resolution, high-speed CCD cameras are used in conjunction with: (i) camera calibration, (ii) feature identification, and (iii) 3D reconstruction techniques to determine time varying positional information of light equipment and contents. Each of these techniques, as used herein, is described in the sections below.
3.3.1 Camera Calibration

Camera calibration involves the estimation of intrinsic (internal) and extrinsic (external) parameters of the cameras. Intrinsic parameters provide the relationship between the image and the camera, and include such things as the focal length $f$, principal point (center of the camera) $(o_x, o_y)$, effective horizontal and vertical pixel size $(s_x, s_y)$, and radial distortion parameters $(k_1, k_2)$. From these, the field-of-view (fov) and coordinates of pixels on an image (e.g. for any point $i$: $x_i, y_i$) can be derived based on geometric relationships (see for example, Trucco and Verri (1998)). Extrinsic parameters provide the relationship between the camera and the real world. These include translational $T$ and rotational $R$ vectors relating camera to real world coordinate systems. Determination of $T$ and $R$ requires knowledge of the camera’s orientation, which may be described by the $upvector$ and $lookat$ vector [Figure 3.1(a)]. $Lookat$ defines a point on the cameras’ viewing axis, while $upvector$ defines the rotation of the camera around the viewing axis. Additional camera calibration techniques, as more generally applicable are described in the Appendix of this report.

![Figure 3.1](image-url)

Figure 3.1: Schematic illustrations of image capture concepts: (a) multiple camera correlation and (b) epipolar constraint.

In use of the light-based system, a static and dynamic calibration approach is used to determine the camera parameters. Static calibration involves placing an L-shaped frame with points (of known distance) within the scene and capturing approximately 50-100 images of this static object. Dynamic calibration involves sweeping a T-rod, with one point on either end of the ”T” (with known distance between the two
points), and tracking these two points in time. Movement of the rod is continued through the capture volume until at least 300-400 frames are collected. The distance between the two tracked points is then calculated, and compared to the exact value, for each frame.

3.3.2 Feature Detection

The presented approach uses highly retro-reflective (passive) markers to identify points of interest within the environment that should be tracked, significantly reducing the amount of information to be collected, decreasing processing time, thus allowing higher capture speeds and resolution. A strobe constructed of a cluster of high-intensity LEDs is used on a per camera basis to illuminate the scene with red light. This strobe can easily illuminate retro-reflective markers at distance between 2-25 meters. The CCD cameras combined with a specially designed hardware filter are then used to measure the light intensity reflected by the passive markers corresponding to the LEDs wavelength.

During the data acquisition stage, only raw data (images showing the marker position) is acquired on a per camera basis. Assuming that other components in the scene have lower retro-reflective properties than the markers, they can easily be filtered from the final image by adjusting LED intensity and camera sensitivity. Each marker is represented by multiple pixels in the final image and before the spatial position can be calculated, the marker "blobs" are approximated (by ellipses or circles) and their respective centroids determined ("blob detection"). Figure 3.2 illustrates both good and poor blob detection applied to individual markers. Poor detection occurs when spacing between neighboring markers is too tight [part (a)] or if the resolution is too low relative to the spherical elements size [part (b)]. The later results in too few scan lines attempting to identify an individual marker. Erroneous detection will also occur if light intensity is too strong or weak, relative to overall scene lighting [part (c)]. Figure 3.2(d) shows a good correlation between the circles and a spherical marker, where the cameras have been placed close enough to the specimen to obtain an optimal number of scan lines per circle (in this case 12.5 mm/8 scan lines = 1.56 mm / scan line).

3.4 Three-Dimensional (3D) Reconstruction

Knowledge of the orientation of multiple cameras focused on a viewing volume allows the determination of three-dimensional positional information of a specific target. The position of corresponding points within individual image pairs can be obtained from multiple cameras and triangulated to obtain their three-dimensional position. Corresponding markers between two images can be found by applying the epipolar constraint, which states that a point in the first image must lie on the epipolar line in the second. This
Figure 2. Examples of good and poor 'blob' detection and correlation: (a) poor spacing selection (too closely spaced), (b) limited resolution, (c) high light intensity and/or poor camera sensitivity, and (d) best correlation.

Figure 3. Spherical marker identification – example of a 3D point cloud in a scene (shake table configuration 1A).

reduces the matching problem from an area to a line segment, as illustrated in Figure 3.1(b). The stereo geometry shown in Figure 3.1(b) is termed epipolar geometry since the lines through the center of the projection, which intersect the image plane, are termed epipoles. Given the projection of a marker into one image plane at p1 and into another image plane at p2, the epipolar constraint is expressed by:

\[ p_2^T F p_1 = 0 \]  \hspace{1cm} (3.1)

where \( F \) is the fundamental matrix (Faugeras, 1993). The fundamental matrix allows the mapping between the essential matrix \( E \) and the intrinsic properties of the camera:

\[ F = M_r^{-T} E M_l^{-1} \]  \hspace{1cm} (3.2)

where \( M_r \) and \( M_l \) are the matrices describing the intrinsic properties \( M_{int} \) of the right and left cameras respectively, for example:

\[
\begin{bmatrix}
-\frac{f}{s_x} & 0 & o_x \\
0 & \frac{-f}{s_y} & o_y \\
0 & 0 & 1
\end{bmatrix}
\]  \hspace{1cm} (3.3)

Note that Equation 3.4 ignores any radial distortion, assumes that \((o_x, o_y)\) are the coordinates in pixel space of the image center, and that \((s_x, s_y)\) are the effective size of the pixel in the horizontal and vertical directions, respectively. The essential matrix \( E \) and fundamental matrix \( F \) are different only in that is defined in terms of pixel coordinates, and is defined in terms of camera coordinates. Given the above intrinsic and extrinsic parameters of the camera system, the three-dimensional position of the markers can be computed from the acquired marker sets. Positional information \((x, y, z)\) on a per marker basis and orientation (yaw, pitch, roll) is available through groups of markers aligned along two different axes. Application of the epipolar constraint allows subsequent computation of the position of arbitrary markers.
3.5 Shake Table Test Program

3.5.1 Experimental Setup

The methodology of light-based motion tracking was applied to monitoring the three-dimensional movement of equipment and contents mounted within a mock-laboratory environment. The mock-laboratory environment, with details representative of typical biological and chemical laboratories in science buildings, was assembled on the 3.1 m x 3.7 m bi-axial seismic simulator facility at the University of California, Irvine.

Two different integral bench-shelf configurations were assembled in the environment, as shown in Figure 3.4(a) and (b). Figure 3.5 shows a photograph of configuration one fully assembled. The two different configurations allow investigation of the seismic response of equipment and contents mounted on bench-shelving systems installed against a wall location. The mock-laboratory was constructed with two full height timber walls at the rear of the configuration, providing support for a Unistrut frame system (which in turn supports the bench-shelving systems) and a rear enclosure for the room. The front of the mock-laboratory provides a clear viewing path for camera capture through the construction of two orthogonal full height steel moment resisting frames. The moment frames were designed with stiffness comparable with the rear timber
walls to ensure minimal torsion under imposed uni-axial earthquake input motion.

Figure 3.4: Schematic layout of the mock-laboratory including bench and shelving details: (a) configuration one and (b) configuration two.

The continuous laboratory bench has two laterally placed rigid shelving systems stacked a single level high, which are anchored onto a horizontal channel system. Horizontal channel elements are in turn supported by continuous vertical uni-strut elements. Laboratory benches are supported vertically and anchored to timber cabinets, which are then connected to a rigid concrete floor. Vertical Unistrut channels are anchored at the floor and ceiling. Layout and anchorage details for the integral laboratory bench and shelving system and supporting Unistrut elements were based on detailed review of common science laboratories (Comerio and Stallmeyer, 2002).

In each configuration, between 80 and 100 passive spherical markers are placed at discrete locations as shown in Figure 3.5, where (a) shows a photograph of the scene not illuminated and (b) shows the same photograph, with the scene illuminated. The approximately mass less spherical markers appear as bright white elements in the scene in Figure 3.5(b). The location of these elements was selected to: (i) monitor the important features of response of elements in the scene, (ii) provide redundancy of individual measurements, and (iii) minimize occlusions from moving objects within the scene or neighboring markers [as shown in Figure 3.3].
3.5. SHAKE TABLE TEST PROGRAM

Figure 3.5: Photographs showing experimental configuration one - mock-laboratory environment: (a) no scene illumination and (b) scene illuminated.

Various types of light, unattached, oddly shaped equipment and contents commonly found in biological and chemical science laboratories are mounted on the ceramic laboratory bench top and on the timber shelving systems. Table 1 summarizes the equipment and contents tested, which can be divided into three general categories: (i) scientific equipment, (ii) computer monitors, and (iii) Silicon Graphics Inc (SGI) computer workstations. Equipment was unanchored, free to move on the laboratory bench top surface. Sufficient edge clearance was provided to minimize the potential of the equipment falling from the bench top. In addition, an array of commonly used chemical glassware was placed on the timber shelving systems as shown in Figure 3.6. The chemical glassware was tested both empty and filled as noted in Table 3.2. Gelatin filling within the glassware was used to maximize liquid containment in the event of spillage. A nominal 25 mm tall Plexiglas lip was provided at the front edge of the shelf to restrain the glassware from falling. In general, all equipment and contents tested were less than 40 kg, fairly short, squat elements, thus dominated in response by their tendency to slide. Additional details of these experiments are provided elsewhere (Hutchinson and Ray Chaudhuri, 2003).

3.5.2 Selection of Earthquake Input Motions

A total of ten earthquake motions were selected as input motions for the seismic testing. Eight of these motions were amplitude-scaled to a range of design level seismic hazards (Sommerville 2002). Ground motions representing a seismic hazard level with a probability of exceedance of 50% in 50 years, 10% in 50 years, and 2% in 50 years were used as input motions. In addition, two of the motions provided by Sommerville (2002) were acceleration-amplitude scaled by one-half to study the low amplitude response of the mock-laboratory and its contents. These motions are denoted GM-1 through GM-10 and listed in Table 3.2. The peak ground acceleration (PGA) of these motions ranges from PGA = 0.13 g to 1.16 g.
Figure 3.6: Photographs showing detail of retro-reflective passive markers used at shelf level to monitor chemical glassware and textbook movements: (a) no scene illumination and (b) scene illuminated (G#1 denotes glassware #1).

![Photographs showing detail of retro-reflective passive markers used at shelf level to monitor chemical glassware and textbook movements: (a) no scene illumination and (b) scene illuminated (G#1 denotes glassware #1).](image)

Table 1: Details of the equipment and contents tested in configurations one and two.

<table>
<thead>
<tr>
<th>Location</th>
<th>Category</th>
<th>Description</th>
<th>$(d\times w \times h)$ (cm)</th>
<th>Mass (kg)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bench Top-Mounted</td>
<td>Scientific Equipment</td>
<td>Small Microscope</td>
<td>$41.9 \times 38.1 \times 20.3$</td>
<td>10.5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Large Microscope</td>
<td>$45.7 \times 55.9 \times 39.4$</td>
<td>21.7</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Techtonic Analyzer</td>
<td>$35.6 \times 48.3 \times 40.5$</td>
<td>17.8</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Eppendorf Centrifuge</td>
<td>$28.6 \times 27.9 \times 21.0$</td>
<td>5.9</td>
</tr>
<tr>
<td></td>
<td>Computer Monitors</td>
<td>38 cm (15&quot; Standard) Diagonal CRT</td>
<td>$38.1 \times 36.8 \times 35.6$</td>
<td>14.1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>43 cm (17&quot; Standard) Diagonal CRT</td>
<td>$41.9 \times 44.5 \times 40.6$</td>
<td>28.3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>48 cm (19&quot; Standard) Diagonal CRT</td>
<td>$44.5 \times 58.4 \times 45.7$</td>
<td>31.1</td>
</tr>
<tr>
<td></td>
<td>Silicon Graphics Inc (SGI)</td>
<td>Indy</td>
<td>$7.6 \times 40.6 \times 34.3$</td>
<td>6.8</td>
</tr>
<tr>
<td></td>
<td>Workstations</td>
<td>Indigo</td>
<td>$47.0 \times 47.0 \times 12.1$</td>
<td>18.2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Octane</td>
<td>$29.8 \times 40.6 \times 27.9$</td>
<td>24.5</td>
</tr>
<tr>
<td>Shelf-Mounted</td>
<td>Chemical Glassware</td>
<td>900 ml Bottle</td>
<td>$\phi = 8.9$, $h = 22.9$</td>
<td>$0.5$ (e) or $1.4$ (f)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>900 ml Bottle</td>
<td>$\phi = 10.2$, $h = 22.9$</td>
<td>$0.7$ (e) or $1.6$ (f)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>500 ml Flask</td>
<td>$\phi = 9.5$, $h = 26.3$</td>
<td>$0.2$ (e) or $0.7$ (f)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>500 ml Flask</td>
<td>$\phi = 9.5$, $h = 29.2$</td>
<td>$0.1$ (e) or $0.6$ (f)</td>
</tr>
</tbody>
</table>

1 (depth x width x height), $\phi = \text{base diameter}$, $h = \text{height}$
2 e = empty and f = full (to capacity)

The selected range envelops the static friction coefficients of the equipment and contents as determined from static bench top testing. Earthquake motions with large displacement magnitudes were considered, up to a peak ground displacement of PGD = 19 cm (GM-6). In addition, motions recorded in the near field, with large velocity pulse characteristics were considered, with the strongest motion having a peak ground velocity of PGV = 64 cm/sec (GM-6).

### 3.5.3 Instrumentation - Conventional and Light-Based

Conventional, wired instruments, linear variable displacement transducers LVDTs and peizometric accelerometers, were used throughout the environment. In each experiment, over 30 conventional transducers were used to monitor the bench and shelf system response as well as the individual equipment response.
### 3.5. SHAKE TABLE TEST PROGRAM

<table>
<thead>
<tr>
<th>Input Motion</th>
<th>Earthquake Name and Location of Recording</th>
<th>Date M/D/YY</th>
<th>Station</th>
<th>PGA (g)</th>
<th>PGV (cm/sec)</th>
<th>PGD (cm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>GM-1</td>
<td>Morgan Hill</td>
<td>4/24/1984</td>
<td>Anderson Dam Down (T)</td>
<td>0.13</td>
<td>7</td>
<td>1.7</td>
</tr>
<tr>
<td>GM-2</td>
<td>Morgan Hill</td>
<td>4/24/1984</td>
<td>Hall valley (T)</td>
<td>0.18</td>
<td>24</td>
<td>4.8</td>
</tr>
<tr>
<td><strong>50% in 50 Year Hazard Level</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GM-3</td>
<td>Morgan Hill</td>
<td>4/24/1984</td>
<td>Anderson Dam Down (T)</td>
<td>0.26</td>
<td>14</td>
<td>3.5</td>
</tr>
<tr>
<td>GM-4</td>
<td>Morgan Hill</td>
<td>4/24/1984</td>
<td>Hall valley (T)</td>
<td>0.36</td>
<td>47</td>
<td>9.3</td>
</tr>
<tr>
<td><strong>10% in 50 Year Hazard Level</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GM-5</td>
<td>Kobe, Japan</td>
<td>1/17/1995</td>
<td>Kobe JMA (L)</td>
<td>0.44</td>
<td>50</td>
<td>11.0</td>
</tr>
<tr>
<td>GM-6</td>
<td>Loma Prieta</td>
<td>10/17/1989</td>
<td>Corralitos (T)</td>
<td>0.53</td>
<td>64</td>
<td>19.0</td>
</tr>
<tr>
<td>GM-7</td>
<td>Loma Prieta</td>
<td>10/17/1989</td>
<td>Gavilan College (T)</td>
<td>0.66</td>
<td>63</td>
<td>13.0</td>
</tr>
<tr>
<td>GM-8</td>
<td>Tottori, Japan</td>
<td>10/6/2000</td>
<td>Kofu (T)</td>
<td>0.69</td>
<td>33</td>
<td>6.0</td>
</tr>
<tr>
<td>GM-9</td>
<td>Loma Prieta</td>
<td>10/17/1989</td>
<td>Lexington Dam (L)</td>
<td>0.84</td>
<td>49</td>
<td>7.3</td>
</tr>
<tr>
<td><strong>2% in 50 Year Hazard Level</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GM-10</td>
<td>Tottori, Japan</td>
<td>10/6/2000</td>
<td>Kofu (T)</td>
<td>1.16</td>
<td>55</td>
<td>10</td>
</tr>
</tbody>
</table>

1 T=Transverse, L=Longitudinal
2 PGA = peak ground acceleration, PGV = peak ground velocity, PGD = peak ground displacement

Table 3.2: Summary of earthquake motions used for base excitation input to the different bench-shelf systems.

Light-based measurements were captured using six high-resolution (1024x1024 pixel) CCD cameras (manufactured by Oxford Metrics, Oxford, England) with maximum capture rates of 250 Hz mounted on tripods and strategically placed facing the mock-laboratory environment. The CCD cameras were outfitted with special filters and 12.5 mm diameter C-mount lenses. The lens selection was designed to capture the full viewing volume based on camera placement. Figure 3.7 shows an isometric view of a geometric model of configuration one, including the placement of the six high-speed, CCD cameras surrounding the environment. Cameras were evenly placed at approximately 1.5-2.0 meters on center, surrounding the specimen. They were aligned such that at least three of the cameras can observe any of the desired points on the specimen at any given time. The closest camera was approximately 4.0 m, and the farthest camera 6.0 m from the center of the specimen. The data-streaming pipeline designed for collecting, processing and storing the light-based data is also shown in Figure 3.7. Individual camera data is collected, streamed in real-time to a unified break-out-box and collectively transferred digitally to a unified data station. Data is then transferred via a 10/100 Mbit Ethernet connection to a dedicated 1.8 GHz processing PC.

The passive markers used in the scene are 12.5 mm diameter Styrofoam spheres wrapped in retro-reflective tape. Compared with the mass of the equipment, contents or the conventional transducers, the mass of the marker elements is negligible. A minimum of three strategically placed passive markers was placed on the bench-mounted equipment and either one or two marker elements were placed on the chemical glassware.
3.6 Experimental Results

3.6.1 Translational Motion

Sample results comparing the input displacement measurements obtained at the base of the shake table using a single LVDT and the average of multiple (between eight and ten) passive markers mounted at the floor level of the mock-laboratory environment are shown in Figure 3.8. This data is shown for the largest displacement amplitude motion used in the study (GM-6). The mean plus and minus one standard deviation is also provided (in the lower figure for clarity). The light-based measurements illustrate good correlation, both in terms of period of motion and amplitude, with the LVDT measurements. The maximum deviation between the LVDT and light-based measurements in the positive and negative direction of motion is (+) 0.11 cm and (-) 0.13 cm, respectively. In addition, the positive and negative standard deviation of multiple passive markers mounted along the rigid base floor and aligned parallel to the direction of motion is very low, as shown in Figure 3.8. The maximum positive and negative standard deviation for these measurements was $\sigma_{max} = \pm 0.15$ cm. Summary statistics for each of the ten earthquake input motions comparing the LVDT
and the light-based measurements are listed in Table 3.3. Error between the two measurement approaches is described in terms of the average maximum positive and negative deviations and maximum positive and negative deviations normalized by PGD, i.e.,

\[
\overline{Dev}^+_{max} = \max^+(\Delta_x^{LVDT}(t) - \Delta_x^{LB}(t))
\]

(3.4)

\[
\overline{Dev}^-_{max} = \max^-(\Delta_x^{LVDT}(t) - \Delta_x^{LB}(t))
\]

(3.5)

\[
\epsilon^+_{max} = \frac{\overline{Dev}^+_{max}}{PGD}
\]

(3.6)

\[
\epsilon^-_{max} = \frac{\overline{Dev}^-_{max}}{PGD}
\]

(3.7)

where \(\Delta_x^{LVDT}(t)\) = x-direction displacement measured by the LVDT transducer and \(\Delta_x^{LB}(t)\) = x-direction displacement obtained by the light-based approach, using multiple marker locations. The range of maximum positive deviations between the two measurement techniques is \(\overline{Dev}^+_{max} = (+) 0.002 - 0.178\) cm, with an average of (+) 0.082 cm. The range of maximum negative deviations between the two measurement techniques is \(\overline{Dev}^-_{max} = (-) 0.009 - 0.182\) cm, with an average of (-) 0.090 cm. With respect to peak ground displacement (PGD) of the input motion, maximum normalized positive deviations range from \(\epsilon^+_{max} = (+) 0.05 - 2.65\)%, averaging (+) 1.14% and maximum normalized negative deviations range from \(\epsilon^-_{max} = (-) 0.05 - 2.79\)%, averaging (-) 1.28%. The standard deviations of the multiple passive marker measurements also indicate efficient measurement ability of the light-based approach. The range of positive standard deviation is \(\sigma^+_{max} = (+) 0.03 - 0.16\) cm, averaging (+) 0.09 cm. The range of negative standard deviation is \(\sigma^-_{max} = (-) 0.02 - 0.15\) cm, averaging (-) 0.06 cm. In addition, it is important to note that multiple runs, considering the described configurations, resulted in consistently low standard deviations, thus the repeatability of the approach is good.

### 3.6.2 Three-Dimensional Motion

The non-uniform mass distribution of many types of light equipment and contents makes it very difficult to monitor their three-dimensional response under imposed uni-axial motions. For this discussion, sample results are provided for GM-6 and GM-10. Time histories, illustrating the acceleration, velocity and displacement characteristics of these motions, are provided in Figure 3.9(a)-(f). Figure 3.10 illustrates an example of the importance of fully defining an equipment elements three-dimensional response. Figure 3.10(a) shows the locations of three discrete marker elements placed on a small bifocal microscope. Figure 3.10(b) shows a photograph of the final position of this microscope upon subjecting the base of the mock-laboratory to
GM-10. Resulting displacement time histories of the bench top, taken as the average of multiple markers mounted at the front of the bench, and that of the microscope, using the three marker elements mounted on it, are shown in Figure 3.10(c). Observation of the movement of these three locations, reveal residual displacements of varying amounts $[\Delta x_{\text{res}} = 21.5, 23.7, \text{ and } 26.5 \text{ cm, for locations two, three and one, respectively}]$, indicating that the element is rotating in the x-y plane. Figure 3.10(d) shows the rotational time history, determined by assuming rigid body rotation of location one and location three, which results in a residual rotation of $\theta_z = 33.6^\circ$. This rotation is confirmed reviewing the photograph shown in Figure 3.10(b). The outlined chalk pattern in Figure 10(b) denotes the equipments original position. It should be noted that small positive and negative reversals of rotation about the z-axis are observed prior to the large translation and rotation (at approximately time $t < 12$ sec) and final resting of the microscope, indicating that the equipment is rotating about its own original axis with minimal translation during the early loading stages.
3.6. EXPERIMENTAL RESULTS

<table>
<thead>
<tr>
<th>Input Motion</th>
<th>$\overline{\Delta v_{\text{max}}}$ (cm)</th>
<th>$\overline{\Delta v_{\text{max}}^{+}}$ (cm)</th>
<th>$\varepsilon_{\text{max}}^{-}$ (%)</th>
<th>$\varepsilon_{\text{max}}^{+}$ (%)</th>
<th>$^{1}\overline{\sigma_{\text{max}}^{-}}$ (cm)</th>
<th>$^{1}\overline{\sigma_{\text{max}}^{+}}$ (cm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>GM-1</td>
<td>-0.009</td>
<td>0.003</td>
<td>-0.49</td>
<td>0.16</td>
<td>0.045</td>
<td>0.041</td>
</tr>
<tr>
<td>GM-2</td>
<td>-0.016</td>
<td>0.002</td>
<td>-0.05</td>
<td>0.05</td>
<td>0.038</td>
<td>0.071</td>
</tr>
<tr>
<td>GM-3</td>
<td>-0.050</td>
<td>0.037</td>
<td>1.52</td>
<td>0.92</td>
<td>0.021</td>
<td>0.063</td>
</tr>
<tr>
<td>GM-4</td>
<td>-0.089</td>
<td>0.081</td>
<td>-1.01</td>
<td>1.11</td>
<td>0.017</td>
<td>0.125</td>
</tr>
<tr>
<td>GM-5</td>
<td>-0.140</td>
<td>0.138</td>
<td>-2.05</td>
<td>2.01</td>
<td>0.086</td>
<td>0.089</td>
</tr>
<tr>
<td>GM-6</td>
<td>-0.125</td>
<td>0.114</td>
<td>-0.95</td>
<td>0.87</td>
<td>0.151</td>
<td>0.147</td>
</tr>
<tr>
<td>GM-7</td>
<td>-0.151</td>
<td>0.148</td>
<td>-1.44</td>
<td>1.41</td>
<td>0.088</td>
<td>0.155</td>
</tr>
<tr>
<td>GM-8</td>
<td>-0.037</td>
<td>0.022</td>
<td>-0.62</td>
<td>0.37</td>
<td>0.030</td>
<td>0.106</td>
</tr>
<tr>
<td>GM-9</td>
<td>-0.099</td>
<td>0.094</td>
<td>-2.79</td>
<td>2.65</td>
<td>0.074</td>
<td>0.028</td>
</tr>
<tr>
<td>GM-10</td>
<td>-0.182</td>
<td>0.178</td>
<td>-1.85</td>
<td>1.81</td>
<td>0.060</td>
<td>0.094</td>
</tr>
</tbody>
</table>

$^{1}\sigma_{\text{max}}^{-}$ = maximum negative standard deviation of multiple passive markers (light-based measurements).

$\sigma_{\text{max}}^{+}$ = maximum positive standard deviation of multiple passive markers (light-based measurements).

Table 3.3: Summary statistics comparing LVDT and light-based measurements - earthquake input motions at the base of the shake table.

Figure 3.11 shows an example of the response of a small monitor [38 cm (15” standard) Diagonal CRT] with a fairly flexible swivel base support. Translational response of this monitor was observed to be fairly small, however large amounts of rotation about the swivel base of the monitor were observed. Multiple identical earthquake input motions were imposed upon this equipment, with resulting response substantially different. In one case (denoted trial #2), overturning and subsequent toppling of the monitor was observed [Figure 3.11(a)], while the identical motion (denoted trial #1) resulted in a large rotation about the swivel base with no final toppling. This anomaly is likely due to the original, manually placed position and orientation of the monitor at the onset of testing. Due to the non-uniform mass distribution, small orientation skews can cause differences in subsequent response. Light-based measurements at four discrete locations on the face of the monitor are used to determine the rotation about the y and z-axis and these rotational time histories are shown in Figure 3.11(b) and (c). These time histories indicate that the combined rotation about multiple axes can be significant. Upon toppling, the trial #2 rotation time history confirms the 90-degree residual rotation, as the monitor is coincident with the bench top. The monitor that remained stable (trial #1), however, was observed to undergo multiple repeated rotational oscillations and come to rest at residual rotations of $\theta_{y(\text{res})} = -2.3^\circ$ and $\theta_{z(\text{res})} = 7.5^\circ$. In this case, the randomness and multiple axis of response were captured fairly well by the light-based measurement technique.
Figure 3.9: Characteristics of the input time histories GM-6 (1989 Loma Prieta, Corralitos) and GM-10 (2000 Tottori, Japan, Kofu): (a,d) acceleration, (b,e) velocity and (c,f) displacement.

### 3.6.3 Glassware Motion

For light chemical glassware, the inclusion of external, wired conventional transducers is fairly cumbersome. Moreover, these transducers will easily change the behavior of the glassware due to their large mass, in comparison with the specimen itself. With the absence of a discrete measurement approach, only qualitative information may be obtained to characterize the dynamic behavior of light glassware. In this study, the light-based approach was applied to allow for monitoring the motions of the different sizes and configurations of glassware considered. Figure 12 shows a sample of the results obtained for the large glassware (flasks and
3.6. EXPERIMENTAL RESULTS

Figure 3.10: Three-dimensional response of a small bifocal microscope - subjected to the 2000 Tottori, Japan, Kofu input motion (GM-10) in configuration one: (a) location of passive markers, (b) photograph of final position at end of motion, (c) displacement of bench and relative x-direction displacement time history of the three passive markers placed on the microscope, and (d) resulting rotational time history about the z-axis.

The bottles) listed in Table 3.2 subjected to GM-10. Figure 3.12(a) shows a photograph of the final position of glassware #2 and #3, where glassware #3 was observed to topple during the simulation. Figure 3.12(b) shows the measured displacement time history of the four glassware elements. Displacement measurements shown in Figure 3.12(b) are relative to the shelf motions, obtained by averaging multiple passive markers aligned along the base of the shelf. Measured displacement results indicate the coincident instigation of sliding of the four glassware elements is clearly captured (at approximately $t = 12$ seconds). This is anticipated since the base characteristics of the different glassware are approximately similar. Residual displacement measurements obtained at the marker locations corresponded with those measured manually post-simulation.
CHAPTER 3. LIGHT-BASED VISION SYSTEMS: OVERVIEW AND VALIDATION

Figure 3.11: Response of a small 38 cm (15” Standard) diagonal CRT computer monitor – subjected to two repeated trials of the 1989 Loma Prieta Corralitos input motion (GM-6) in configuration one: (a) photograph of toppled monitor (Trial #2) at end of motion, (b) rotational time history about the y-axis and (c) rotational time history about the z-axis for Trial #1 and #2.
3.7. Summary Remarks

Traditional sensors, such as accelerometers and displacement transducers are widely used in laboratory and field experiments in earthquake engineering to measure the motions of both structural and nonstructural components. Such sensors, however, must be physically attached to the structure and require cumbersome cabling, configurations and substantial time for set-up. For reduced scale experiments, conventional sensors may substantially alter the dynamic properties of the system by changing the mass, stiffness and damping properties of the specimen. Moreover, it is very difficult with traditional sensors to capture the three-dimensional motions of light, oddly shaped components such as microscopes, computers or other building.
In this chapter, the methodology of light-based motion tracking is applied to the measurement of the three-dimensional motions of various types of equipment and building contents commonly found in biological and chemical science laboratories. Positional time histories are compared with measurements obtained using conventional sensors. Maximum deviations between the two measurement techniques are less than 2.0 mm in all cases and on average less than 1.0 mm. The standard deviations of multiple passive marker measurements also indicate efficient measurement ability of the light-based approach. Results show that the non-intrusive, light-based approach is extremely promising in terms of its ability to capture relative displacements in three orthogonal directions and complementary rotations.
Chapter 4

Image-Based System: Hardware and Software Design

In contrast to the light-based system described in Chapter 3, which uses only specific color channels to track features within a scene, the more general condition, of using all information captured by the sensor is much more challenging and requires rigorous consideration of the hardware and software pipeline to obtain accurate tracking results. In this chapter, we describe the hardware and software design implemented for the development of a fully image-based capture system, which is deemed suitable for a general building interior.

4.1 Image-based System: Hardware Design

The hardware for the image-based capture system consists of an array of four high-speed charged-couple-device (CCD) cameras, a suitable computational platform, data storage, and the necessary cabling and power supplies. The following sections discuss the selection of the camera array and the design of the computational platform for this system.

4.1.1 Camera Array Design

A four camera array was selected to provide redundancy in image collection and to represent the idea of placing image-sensors throughout a room for observation (such as in each of the four corners of a typical building room). This camera array will allow us to determine motions in all three dimensions (translations and rotations) and overlapping fields-of-view (FOVs) will provide redundancy, thereby increasing resolution.

For our work, CCD camera’s were selected due to their high linearity, solid-state nature and potential for high resolution and high speed. The terminology 'CCD camera’ comes from the premise that images are collected on a CCD chip, in charge packets. Light hitting the chip substrate (on individual photodetector sites located on the CCD chip) is moved in individual charged packets off of the chip. The CCD itself
is manufactured on a light-sensitive crystalline silicon chip, with photodetector sites built onto the chip. Currently, there are three basic types of CCD architectures available: (i) full frame CCD, (ii) interline-transfer CCD, and (iii) frame-transfer CCD. The variations in architecture define the mode of transfer of accumulated charge off of the CCD image-sensing device. Perhaps the most popular, which we have selected, is the interline-transfer CCD. In the interline-transfer CCD, every other column of the sensor is covered by an opaque mask, and information is moved laterally onto these masks then shifted (down the chip) to a serial register, as illustrated in Figure 4.1. This mode of transfer is important since it means that no more than 50% of the chip area is light sensitive. It also means a tight spacing of light sensitive areas is provided across the entire chip, helping maximize the chips ability to retain the input light.

Figure 4.1: Interline transfer of charge packets of photodiodes laterally and then vertically (into register) on the CCD chip (schematic courtesy of the Eastman Kodak Company (2003)).

4.1.1.1 Survey of Available Camera Technologies

Based on the dominant range of frequencies of movement generated during earthquakes and satisfying optimal sampling rates (i.e. to ensure aliasing of images does not occur), data acquisition rates of at least 30 frames per second are necessary. Ideally, higher frequencies (i.e. greater than 60 frames per second) are desirable to allow for differentiation of acquired movement data to obtain velocities and accelerations. In addition, the vision sensors and associated processing system should be capable of capturing image data at high resolutions. Given only these specifications, there are hundreds of vision sensors available. However, given other desirable attributes, such as low cost, small size, high shock rating, readily available and compatible interfaces (such as IEEE firewire or USB-2 interfaces), the selection greatly reduces. Table 4.1 lists three potential models that may be considered for the application area of earthquake monitoring.
Although these models are rather expensive compared to regular off-the-shelf web cameras, the cost compared to the overall cost of a building is rather small, which makes the investment feasible. Other limiting factors for the overall system include interface transfer hardware (e.g. PC-specific hardware such as the PCI bus, hard drives, interface cards, etc.). Any of these elements may have the potential for limiting the system bandwidth and subsequently the capture rate. Running full resolution at the highest speed may challenge PCI bus read and write speeds, for example. Upon capturing these large volumes of data, compression and storage techniques will also need to be carefully considered.

Table 4.1: Comparison of specifications and cost of different available camera models.

<table>
<thead>
<tr>
<th>Model</th>
<th>Basler</th>
<th>Pulnix</th>
<th>Cooke</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resolution</td>
<td>658x494</td>
<td>640x494</td>
<td>640x480</td>
</tr>
<tr>
<td>Frame rate</td>
<td>80 fps</td>
<td>60 fps</td>
<td>40 fps</td>
</tr>
<tr>
<td>Color depth</td>
<td>8 bit</td>
<td>24 bit</td>
<td>24/36 bit</td>
</tr>
<tr>
<td>Price (U.S.)</td>
<td>$1,895</td>
<td>$1,795</td>
<td>$6,090</td>
</tr>
</tbody>
</table>

Based on the survey of available CCD cameras, we selected a suitable high-speed, reasonable resolution camera for use in the four-camera array. Camera selection was also based on robustness of driver libraries available by the various manufacturers. The selected cameras are manufactured by Basler (Basler Vision Technologies, 2003) and distributed locally by Aegis Electronics (Aegis Electronics Group Inc., 2003). The specific cameras selected were the Basler A301fc (color) model, with a rated 658x494 (8bit per pixel) resolution, capable of 80 frames per second (fps), and interfaced via IEEE 1394 (FireWire) technology. The gain, brightness, and exposure time of these cameras are programmable through the IEEE interface, adding flexibility and control within the software. The use of the IEEE interface also allows us to run relatively long cable lengths from the cameras (72m with repeaters or 40 km with an optic link), while still providing flexibility to connect to laptop and desktop systems without the need for extra hardware (such as frame grabber cards). The FireWire standard also allows devices to be powered through the data cable, further reducing the amount of required wiring. The cells within the CCD chip on these cameras contain square 9μm x 9μm pixels (12.7mm square CCD sensor), which are progressively scanned at once. The progressive area scan nature of these cameras allows us to obtain the higher frame rate.

The camera itself, as shown in Figure 4.2(a), is small (roughly 62mm square body with variable length, associated with the extension of the c-mount lens) and lightweight (310 g without a lens). The cameras are also rated for 10g in vibration, well above anticipated demands during seismic motion. Four steel mounting 'wings' [Figure 4.2(b)] were built for each of the cameras to allow ease of attachment (e.g. using wood or
metal screws).

![Image of Basler A301fc camera with pen and mounting wings]

Figure 4.2: Photographs of the Basler A301fc camera: (a) with pen for scale and (b) with mounting wings ready for assembly.

The silicon chip that the CCD is mounted on is unable to distinguish colors, therefore either color separation optics (e.g. three CCD chips) or color separation filters must be used to obtain coloring in the final image. Color filters must be placed in front of the chip and after capturing an image one can interpolate to obtain the desired coloring. The A301fc camera’s have a color separation filter (known as a Bayer, or mosaic-style filter) in front of the CCD chip, allowing only one color to pass on any pixel (either red ($\lambda \geq 600\text{nm}$), green ($\lambda = 500-600\text{nm}$), or blue ($\lambda = 400-500\text{nm}$)). Across any 2x2 pixel array, two diagonally placed green sensitive, one red sensitive, and one blue sensitive pixels may be found, as illustrated in Figure 4.3. Using bilinear interpolation the full color information for all pixels can be computed, resulting in a colored image. These cameras cover a spectral range from $\lambda = 400 - 1000\text{ nm}$, well outside of the human vision system ($\lambda = 400 - 800\text{nm}$).

Two different sets of Cannon lenses were purchased for these cameras with fixed focal lengths of 4.8mm and 12.5mm. With variable magnifications of $\beta = 1:1.8$ to 1:16, these lenses allow millimeter resolution at distances of between 1.5-6.0 meters. The lenses were equipped with locking screws to assure they would remain fixed at their final settings during shaking.

### 4.1.2 Computational Platform and Data Flow

The full resolution and frame rate of these cameras results in acquisition of approximately 24.8 MB/s of data per camera, which must then be transferred across the IEEE 1394 (FireWire) bus through a computational
platform and onto a storage media or device. Four cameras acquiring data at full resolution and acquisition rates at the same time result in a total data rate of approximately 99.2 MB/s. The Basler cameras use the FireWire 400 standard, which is limited to 400 Mbit/s or 50 MB/s. Therefore, one FireWire PCI card per camera is used to guarantee the necessary bandwidth. Although FireWire 800 cards and cables (with 800 Mbit/s) have recently become available, at the present time, camera’s are not available to interface with this standard. Once available, multiple camera’s could be interfaced with a single FireWire (800 standard) PCI card.

After data is acquired and passed to the FireWire PCI card, it must then be transferred through the systems PCI bus. Table 4.2 lists the common PCI buses and their speeds in the current market. Of these, the 32-two bit PCI bus running at 33MHz is the most commonly found in current desktop computers. These buses have a theoretical peak transfer rate of 132 MB/s. However, using four cameras, simultaneous read and write operations across the PCI bus would require twice the full output from all four cameras or 198.4 MB/s, well in excess of theoretical rates for typical desktop computers. A more optimal solution is to allow read operations to occur across one bus and write operations to occur across another bus, suggesting the use of a motherboard with multiple PCI buses. Server-style PC systems in this case provide such a configuration, where at least two independent PCI buses are typically arranged on the primary motherboard. In this case, we have selected Intel’s server mainboard SE7501BR2, which features three independent PCI buses, two of which are 64bit wide PCI-X buses running at 100 MHz. The wider bus architecture and faster clock frequency allows data transfer rates up to six times greater than the conventional 32bit PCI standard. A
CHAPTER 4. IMAGE-BASED SYSTEM: HARDWARE AND SOFTWARE DESIGN

<table>
<thead>
<tr>
<th>Bus Type</th>
<th>Bus Width</th>
<th>Bus Speed</th>
<th>Bandwidth</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCI</td>
<td>32 bits</td>
<td>33 MHz</td>
<td>132 MB/sec</td>
</tr>
<tr>
<td>PCI</td>
<td>64 bits</td>
<td>33 MHz</td>
<td>264 MB/sec</td>
</tr>
<tr>
<td>PCI</td>
<td>64 bits</td>
<td>66 MHz</td>
<td>512 MB/sec</td>
</tr>
<tr>
<td>PCI</td>
<td>64 bits</td>
<td>133 MHz</td>
<td>1 GB/sec</td>
</tr>
</tbody>
</table>

Table 4.2: Common PCI buses and their associated speed and bandwidth.

A single Intel 2.8GHz XEON CPU is mounted on the board.

Figure 4.4 shows an overview of the dataflow inside the computational platform designed for capturing data from our four camera array. The left photographs also show the layout of the camera’s on the shake table. Data rates produced by the capturing process are provided next to each bus in the computational platform. Below that, the maximum theoretical capacity of the bus (at that location in the transfer) is listed in parenthesis.

![Figure 4.4: Capturing and storing data using the image-based system: (left) photographs of cameras located on the shake table and (right) data flow inside the computer system after capturing.](image)

As illustrated in Figure 4.4, image data must first be moved from the camera via the IEEE interface to the PCI and PCI-X buses. Since the amount of data produced by the cameras already exceeds the bandwidth of...
a single 32bit PCI bus, two of the four FireWire cards are attached to a 64bit PCI-X bus while the other two remain on the 32bit PCI bus. Data must then transfer to the CPU, while moving across the main memory modules. This path involves data transmitted through the 533 MHz Front Side Bus and the 266 MHz DDR Ram memory bus. The bandwidth limitations of both bus systems are much higher than the demands of this application, therefore, these buses do not present any limitations even though the images have to cross both buses twice. After moving across the processor and RAM modules, the data must be stored for future analysis. For this purpose, the third PCI-X bus is linked with a SCSI bus and interfaced with fast SCSI hard drives. This third PCI bus is dedicated for data transfer to the hard drive, therefore its full bandwidth (480 MB/s) may be utilized.

Hard drives that are currently available typically have an advertised transfer rate of 60 MB/s or greater. Unfortunately, these speeds can only be attained while reading information. Observing the high bandwidth requirements of our application, we equipped our system with fast Seagate ST336752LW SCSI hard drives. Benchmark tests showed that the transfer rate while writing an AVI stream to these devices is about 27 MB/s. Consequently, assigning a single hard drive per camera was possible, leaving a small margin of 2.2MB/s between the write throughput of 27MB/s and the demand (data rate per camera) of 24.8MB/s. Finally, the architecture of the computational system designed and shown in Figure 4.4 allows us to capture data at the full frame rate and resolution produced by the four camera array attached to the system.

4.2 Image-Based System: Software Design

A primary component of our development work is centered around SceneIdentifier, our software framework for (i) image capture and acquisition, (ii) image processing and (iii) data analysis and results preview. The system was implemented in C++ and uses a development platform known as QT\textsuperscript{1} for its graphical user interface (GUI) development. QT provides the thread management and event handling features needed, and is platform independent. The source code is developed using internal coding standards\textsuperscript{2} and an auto-generation documentation system called Doxygen\textsuperscript{3}. A revision control system is used to adhere to the latest software developments\textsuperscript{4}. The following sections present the image capture and acquisition process and the performance testing of this process.

\textsuperscript{1}QT for QUI Development (http://www.trolltech.com)
\textsuperscript{2}VIS coding standards (http://vis.eng.uci.edu/standards)
\textsuperscript{3}Doxygen documentation system (http://sourceforge.net/projects/doxygen)
\textsuperscript{4}Concurrent Version System (CVS) revision control (http://www.cvshome.org/docs/manual)
CHAPTER 4. IMAGE-BASED SYSTEM: HARDWARE AND SOFTWARE DESIGN

4.2.1 Image Capture and Acquisition

Digital image/video data can be directly acquired from the Basler 301fc camera through its IEEE 1394 (FireWire) interface. One advantage of this configuration is that no additional framegrabbers are required. Framegrabbers are generally needed for conventional video cameras adding cost to the overall system configuration. However, as will be described, we designed a special software pipeline to bypass the use of hardware framegrabbers altogether. Prior to describing this, we highlight the capture process within the software.

4.2.1.1 Capture Process

While the cameras can acquire images at speeds of up to 80 frames per second, they do not have a notion of video streams. In order to obtain videos, individual images have to be acquired and composited into a video. Microsoft’s AVI (Audio Video Interleaved) format was selected as the preferred data format to manage these streams. AVI uses the resource interchange file format (RIFF) to efficiently encode multimedia audio and video information. Our framework supports the sequential acquisition of still images from the Basler cameras at up to 80Hz and the real-time generation of the corresponding AVI file. The AVI encoding and decoding is included in our SiAviInput and SiAviOutput classes.

As described in Section 4.1.2, it was desirable to control the entire sensor array (consisting of the four high-speed cameras), from a single host computer. To avoid unnecessary wait states and allow proper control of timing and sequencing of the image capture, we have developed a multi-threaded application to handle data acquisition. Figure 4.5 illustrates the parallel threads running in our application. One thread is responsible for handling requests for images that are dispatched to the camera array and the other four threads (one for each camera) are responsible for transferring images from a shared buffer to the AVI stream that is stored on a camera specific system disk. The storing threads runs with real time priority to ensure that all images are merged into the respective AVI stream as soon as they become available. If a thread observes delays in data acquisition it may also temporarily release computational resources to the other system threads to increase overall system performance, leaving enough computational cycles for all of the required operations.

The image capture is synchronized with the image transfer threads through multiple image buffers. One of these image buffer threads is available for each camera and accessed by the corresponding transfer thread, allowing temporary storage of up to 256 acquired images, in the event that system resources or bandwidth become limited for a short period of time. System performance tests, as will be presented in Section 4.3, show that our software design successfully allows the capture of video data at full resolution and frame rate
for the four camera array, using the computational system described in Section 4.1.2.

4.2.1.2 SceneIdentifier Graphical User Interface (SIGUI)

A user friendly GUI was developed to provide complete control over all camera parameters and options such as the shutter speed, gain, acquisition rates, and white balance. Figure 4.6(a) shows a snapshot of the SceneIdentifier with the capture dialog opened. The camera controls dialog allows the user to control specific camera parameters in the main window, as shown in Figure 4.6(b). The main window shown in Figure 4.6(a) includes tabs for image analysis such as a playback of the original video sequence (after it has been interpolated to present color values), edge detection and visualization, reconstruction of objects, and the analysis of time-varying waveforms of selected items. Some of these features can be observed in real-time, while others, such as object reconstruction, must be replayed from a previously acquired record.

The user can also specify the desired location and anticipated size of the video file that will be generated by each camera, through SIGUI's file management system. In addition, the user interface of SceneIdentifier supports the following capabilities:

- An arbitrary number of cameras (no hard-coding)
• Auto-detection of newly installed cameras

• Camera listing for the user, including options to enable/disable individual cameras

• Grouping of individual cameras observing specific parameters (frames per second, etc.)

• Operation as a client for remote control

Figure 4.6: Snapshot of the SceneIdentifier graphical user interface (SIGUI): (a) overall view and (b) camera controls.
4.2.1.3 Camera View and File Management in SIGUI

SIGUI has a unique camera view feature, which allows control over all of SceneIdentifier’s data acquisition capabilities. At startup time, the application performs a hardware query to determine the type and number of firewire cameras connected to the data acquisition (DAQ) node. Available cameras are subsequently displayed in enumerated form in the camera tree view, as shown below.

```
+Cameras
  +-- Camera 1
  +-- Camera 2
  +-- Camera 3
  +-- Camera 4
```

The database view provides the user with access to all data that was previously acquired, including image streams, and experiment logs. It provides the basis for users to manage files in logical sequence with experiments. Figure 4.7 highlights the basic features of the GUI. The tree view presented can provide the following information:

- Experiment Classification
  -- Experiment
    --- Experiment Configuration
    ---- Experiment Parameters
    ----- Camera Data

For a typical experiment this may look as follows:

- "Field Test" (name of the data base entry)
  -- "02072004" (name of the specific test sequence, e.g. date of experiment)
    --- "Configuration 1" (name of the experimental setup)
      ---- "GM 1" (name of the ground motion)
        ----- "Camera xx" (the camera capturing the data for the above series)

4.3 Acquisition System Performance Evaluation

The specialty design of the camera hardware and software acquisition system as described above required testing to assure video streams could be properly collected. Of particular interest is the evaluation of the
parallel thread architectures’ capability to maintain proper synchronization and acquisition speeds during long capture runs. A number of experiments were performed to test these capabilities, and results of these are presented in this section.

Two variables in the hardware and software will be sensitive to the final acquisition speed. First, we recognize that the camera’s internal ability to maintain acquisition speed will be sensitive to the shutter speed settings. Therefore, different shutter speeds are selected for the test runs. To determine the selections, we seek the theoretical shutter speeds associated with the cameras optimal capture limit.

There are two main hardware limitations in the capture process, the first is the exposure time, the second is the time to transfer data from the sensor. The theoretical shutter speed \( v_s \) is simply the inverse of the exposure time \( t_e \). For the camera sensors used in this work, \( t_e \) is defined as \((\text{shutterValue} + 1) \times 20\mu\text{sec}\), where the \text{shutterValue} varies from 0 \(\rightarrow\) 4095 (Basler Vision Technologies, 2003). This means that an optimal shutter value of 624 would result in the theoretical limit of the camera, i.e. 80 Hz. However, these
cameras require 11.7 milliseconds to transfer data from the sensor. While a frame is being created, image data can be transferred in parallel, however, a second image cannot be processed. Therefore, to obtain the optimal 80Hz capture rate, a maximum shutterValue of 584 can be set. Based on this argument, we select shutter values just near maximum (500) and approximately half of maximum (250) for these experiments.

Second, given the hardware pipeline described above, a likely bottleneck for acquisition speed is the number of cameras, as the computer must process the incoming data. We therefore consider configurations varying from one through four cameras. Thus, the final acquisition experiment matrix includes shutterValues = 250, 500; configurations consisting of 1, 2, 3, and 4 cameras; five trial runs each; and a 60 second capture duration.

### 4.3.1 Results and Discussion

Sample results for a single camera, with the shutterValue setting at 250 are shown in Figure 4.8. This figure shows all five trial runs, where a target setting of 80 fps was set in the software. Although in these performance runs, capture times were set to 60 seconds, for this plot we truncate the x-axis to 20 seconds for clarity. Periodic single point 'spikes' (drops to 40 fps) are evident, where we force the frame rate to split, indicating that we may have a data overload. These spikes are only instantaneous and very minimal, occurring only once per approximately every 4-6 seconds. As such, they may therefore be easily removed. After post-processing the data, the final results for the different configurations are shown in Figures 4.9 and 4.10. These figures illustrate that as the number of cameras is increased, a nominal increase in the noise is observed (observing parts a-d). However, summary statistics provided in Table 4.3 illustrate excellent performance of the system, with mean frame rates of above 79.99 attainable in each trial case. Moreover, the highest standard deviation $\sigma$ of these runs was 0.246 fps, approximately 0.3% of the target 80 fps. Results indicate the lowest mean frame rate of these experiments as 73.19 fps. These results substantiate the stable and superior performance of the camera hardware and software pipeline developed.

### 4.4 Summary Remarks

In this chapter, we describe a specialty hardware and software design for an optimally low cost image acquisition system. The system is designed to be used within a real building environment for object tracking, and as such performance goals, namely high resolution and speed are desirable. A unique multi-threaded software application was developed to avoid busy waits and optimize the computational power of a standard server style computer. Performance experiments considering different shutter speeds and camera configurations il-
Figure 4.8: Sample performance data for one camera with a *shutter value* = 250.

<table>
<thead>
<tr>
<th>Performance Run Series</th>
<th>Details</th>
<th>Number of Cameras</th>
<th>Ave Mean (m) (fps)</th>
<th>Ave Minimum (fps)</th>
<th>Standard Dev (σ) (fps)</th>
<th>m + σ (fps)</th>
<th>m - σ (fps)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>5 runs at 250 shutter speed</td>
<td>1</td>
<td>79.998</td>
<td>76.428</td>
<td>0.099</td>
<td>80.097</td>
<td>79.899</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2</td>
<td>79.997</td>
<td>73.194</td>
<td>0.169</td>
<td>80.166</td>
<td>79.828</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3</td>
<td>79.998</td>
<td>77.012</td>
<td>0.181</td>
<td>80.179</td>
<td>79.817</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4</td>
<td>79.999</td>
<td>76.334</td>
<td>0.190</td>
<td>80.189</td>
<td>79.809</td>
</tr>
<tr>
<td>2</td>
<td>5 runs at 500 shutter speed</td>
<td>1</td>
<td>79.998</td>
<td>77.065</td>
<td>0.087</td>
<td>80.085</td>
<td>79.911</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2</td>
<td>80.000</td>
<td>76.025</td>
<td>0.191</td>
<td>80.191</td>
<td>79.809</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3</td>
<td>79.997</td>
<td>76.511</td>
<td>0.180</td>
<td>80.177</td>
<td>79.817</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4</td>
<td>79.995</td>
<td>76.536</td>
<td>0.246</td>
<td>80.241</td>
<td>79.749</td>
</tr>
</tbody>
</table>

Table 4.3: Summary of capture performance experiments considering multiple camera configurations and different shutter exposure times.

I llustrate the superior performance of the software and hardware system developed. In the following chapter, the key image processing and data analysis algorithms implemented for feature tracking are presented, using the above hardware and software system.
Figure 4.9: Performance data summary plots for \textit{shutterValue} = 250 for: (a) one camera, (b) two cameras, (c) three cameras, and (d) four cameras.
Figure 4.10: Performance data summary plots for \textit{shutterValue} = 500 for: (a) one camera, (b) two cameras, (c) three cameras, and (d) four cameras.
Chapter 5

Image-Based System: Processing and Data Analysis

Using the image-based hardware and software acquisition system described in Chapter 4, image processing and data analysis tools are implemented to support tracking of dynamic movements in the scene. Select image processing and data analysis capabilities are supported for the image streams as they are being acquired (in real-time), while other computationally expensive tasks, are performed in a post processing step if extensive access to the video streams is required. In this chapter, we describe the actual processing applied to the images and the data analysis steps undertaken. It is important to note that all data is initially stored in Bayer filtered (raw monochrome) format to reduce storage size. This means that the recorded AVI streams have to be converted to their color equivalent before they can be processed with some of the image processing algorithms used (Figure 4.3). Once colors are computed (as described in Section 4.1.1), features may then be detected and tracked between consecutive images based on the presented image processing pipeline.

5.1 Pixel-Based Image Processing Chain (PIPC)

Image processing and data analysis is realized using a pixel-based motion detection approach, whereby pixels identify objects and features. For both motion detection and object recognition tasks, we use and extend the Open Source Computer Vision Library (OpenCV) (Intel Research Group, 2004). This library is a collection of computer vision algorithms and sample code for various image processing tasks that can be applied to a special pixel based image format (IplImage). In addition, the library also includes basic functions for motion detection. The general outline of the pixel-based image processing chain (PIPC) implemented in this work is illustrated in Figure 5.1. This analysis is applied sequentially for all images in a given video stream.

The process begins with camera calibration and correction of the image, particularly for radial and
CHAPTER 5. IMAGE-BASED SYSTEM: PROCESSING AND DATA ANALYSIS

Figure 5.1: Pixel-based image processing chain (PIPC) flowchart.

tangential distortions introduced by the camera lens. Subsequently, to identify trackable objects in the scene and to reduce computation time, we use a mask-based object identification approach. The object mask can be determined either by user-defined regions in the acquired image, or by referencing an image to a precalculated model of the clean background. In support of the later approach, several background detection methods were implemented and evaluated. Features of interest are then identified within the object mask in each image. Feature data sets are then stored and passed to a 2-dimensional (2D) feature tracking algorithm. The implemented feature tracking algorithm is based on the calculation of the optical flow of pixel values in an image sequence. The following sections describe each of these steps of the PIPC in detail and provide examples from real scenes.

5.1.1 Camera Calibration and Image Correction

Camera calibration and image correction are fundamental steps that must be applied before reliable image processing and analysis can occur. A complete review of camera calibration fundamentals and techniques applied is provided in the Appendix (Villa-Uriol et al., 2004) and herein we focus only on the description and identification of the intrinsic parameters for each camera. Of particular interest is the image correction for radial and tangential distortions as it is commonly assumed that all camera lenses introduce such distortions.

The intrinsic camera parameters establish a relationship between the image and the camera (Figure 5.2) and include the coordinates of the camera’s principal point \((c_x, c_y)\), the focal length in the \(x\) and \(y\) directions \((f_x, f_y)\), the radial distortion coefficients \((k_1, k_2)\) and the tangential distortion coefficients \((p_1, p_2)\).
5.1. PIXEL-BASED IMAGE PROCESSING CHAIN (PIPC)

These parameters can be computed from sets of image pairs if a set of reference points are known. In this work, we use an implementation in MatLab (The MathWorks™, 2003), which operates directly on acquired video streams and returns a set of transformation matrices containing both the intrinsic as well as extrinsic camera parameters. We then pass these parameters to the OpenCV image correction function and apply them to each frame of the video sequence. Figure 5.3 illustrates the effect of the correction function when applied to a reference image. Small distortions can be seen by comparing the subtle differences in the gap between the two shelves or the perspective distortion of the lower edge of the two shelves seen in each image shown in parts (a) and (b).

5.1.2 Defining Plausible Features for Tracking using an Object Mask Method

To identify trackable objects in the scene and to reduce computation time, we use a mask-based object identification approach. The object mask can be defined manually for the acquired images, or automatically computed by comparing the current image in a sequence to a calculated background image. To reliably use the later automatic method, clearly a very clean background image is required. A clean background image is one that captures all of the static information in the scene, i.e. before any movement occurs. Therefore, in the following sections, we first define plausible background detection and extraction approaches and then provide illustrative examples of each. Specifics of the object mask calculation are then provided using the automated approach.
5.1.2.1 Background Detection and Extraction

Three different methods are evaluated to detect and extract the background image. These methods include (1) a static background image method, (2) a running average method and (3) a mean-standard deviation method. All of these methods perform best if a clean image of the static environment can be created before the scene is populated with objects or reference patterns that will subsequently be tracked.

The evaluated background detection methods and their results under different environmental conditions are presented. Results are demonstrated for the test environment and settings depicted in Figure 5.4. The environmental settings were selected based on synthetic as well as realistic field conditions to allow for an
evaluation of isolated effects (real scenes tend to have many environmental contributions). Three test cases are selected as follows:

- **Case I**: static light and background conditions
- **Case II**: slight movements of the background
- **Case III**: varying lighting conditions that introduce shadows

![Figure 5.4: Baseline image sequence for evaluating background detection methods (picture on a wooden desk, white background with holes): (a) Case I: static test environment, (b) Case II: moving background (note directional vectors at holes behind object), and (c) Case III: lighting/shadow effects (note shadow at right upper corner).](image)

### 5.1.2.2 Background Detection: Threshold Definition

All of the described methods use a threshold value to define which pixel belongs to the background and which belongs to the objects in the scene. This threshold value is used during pixel by pixel comparison between the current processed image and the background model. The range of possible pixel values is between 0 and 255 for each color channel. At time $t$, the absolute difference between the new image $p_t$ containing the object and the background image $p^{bckg}$ is then calculated for each pixel $(x, y)$ in the image:

$$p_t^{\text{diff}} = \left| p_t - p^{bckg} \right|$$  \hspace{1cm} (5.1)

If a pixel color value has changed more than the defined threshold value, the pixel will be assigned to the object mask, otherwise it is assigned to the background. Therefore, the definition of an appropriate value for the threshold is an important consideration. If the threshold value is set too high, select objects may not be recognized. Figure 5.5 illustrates this problem for threshold values of $thr = 60$ and 30. In this example, the tracking frame (part a) shows two objects in the scene. The original (background image) is shown in Figure 5.4(a), therefore it is clear that the desirable feature to track is the second object to the right. By
selecting a threshold value of 60, the change detection does not identify the second object as belonging to the object mask (part b). If the threshold is set to 30, the object can be recognized and additional features from the first object can be identified as well.

![Figure 5.5: Example of sensitivity to threshold settings: (a) image of interest (tracking frame with two objects), (b) identified object mask \( \text{thr}=60 \), and (c) identified object mask \( \text{thr}=30 \).](image)

5.1.2.3 Background Detection: Snap Shot Method

The snapshot method uses just one image frame to identify the overall background. Subsequently, the object mask is generated by using the absolute difference between the background image and a processed frame with the object (picture) in it, i.e. Equation 5.1. In the following examples, the threshold parameters of \( \text{thr} = 60 \) and 30 are used to illustrate this method. The baseline static background image includes (i) no changes in lighting and (ii) no camera or background movement (a static condition), i.e. Case I. As depicted in Figure 5.6, the object detection works well under static laboratory conditions and the object can be clearly identified. Note the cases shown in the following sections use the Canny edge detector and Contours algorithm as a method to identify the new object introduced in the background. Each of these are described later in Sections 5.1.4.1 and 5.1.3.

**Issues with the Snap Shot Method** – This method has difficulties reconciling the background when either the camera or the background (or objects in the background) is slightly different during acquisition (Case II). This effect is illustrated in Figure 5.7, where additional objects are detected when the background changes (holes move into the background) after the snapshot is taken. This change can be compensated by adjusting the threshold value if the contrast of the noise is not too significant. However, in most cases moving background objects will always be detected as noisy elements. The same results will be present when the camera position changes during tracking.
Figure 5.6: Background detection using the **snapshot method**, **Case I**, images with $thr=60$: (a) mask, (b) edges defined by Canny, and (c) contours.

As depicted in Figure 5.8, when lighting conditions change (**Case III**), this method also has difficulties, as the mask will contain noisy objects and additional contours will be detected. To overcome this problem with slightly changing light conditions, again larger threshold values can be used. However, excessively large threshold values will cause pixels in subsequently tracked objects to disappear in the mask. It may be challenging to identify a suitable value of $thr$, as it will change from one test sequence to another disallowing real consideration of changing light conditions over time.

Given the above limitations, this method will introduce problems when choosing suitable threshold values to minimize the effects of slightly varying lighting conditions. This method is therefore classified as a low quality method, and should only be used for short duration data acquisition.
CHAPTER 5. IMAGE-BASED SYSTEM: PROCESSING AND DATA ANALYSIS

5.1.2.4 Background Detection: Running Average Method

The running average method uses a sequence of image frames to calculate the background model. In general the background model is calculated by weighting and accumulating the image pixel values over a number of frames. The updated background model \( \mu_t \) at time \( t \), using the running average method, is expressed as:

\[
\mu_t = \alpha y_t + (1 - \alpha) \mu_{(t-1)} \quad 0 \leq \alpha \leq 1
\]

where \( \alpha \) is an averaging constant, \( y_t \) is a new image observed at time \( t \), and \( \mu_{(t-1)} \) is the \( t - 1 \) time step calculated background model. This weighting approach generates a background model from the mean pixel values over an applied number of frames. The value \( \alpha \) is thereby used to define how much the last captured background frame affects the background model. The mask is again generated by using the absolute difference between the background model and a tracking frame image with the object in it (Equation 5.1).

A number of examples of the running average method are presented herein, with the following settings used: (i) threshold parameters of \( thr = 60 \) or \( 30 \), (ii) number of frames = 10 or 30, and (iii) \( \alpha = 0.05 \) or 0.5. The baseline static background image is again defined by no changes in lighting and no camera or background movement (Case I, a static condition). If the background is static, the same (positive detection) results as observed with the snap shot method can be achieved.

Issues with the Running Average Method – Using the running average method, if the background is changing during acquisition, the frame weight function will "smoothen" the background model (Figure 5.9). This results in less noise in comparison with the snap shot method (Figure 5.7). If additional frames are used for the background detection, the noise can be eliminated without changing the threshold value (Figure 5.10).

The selection of a suitable weighting per frame may require special consideration, as shown in Figure 5.11,
if subsequent images have significant pixel value changes, where a *spike* in the normal background distribution results. This can occur, for example, if an artificial light source such as a neon light illuminates the scene. Neon lights introduce a high frequency change in lighting conditions that heavily bias the calculation of the background image if based on only a few frames. Clearly, the number of frames used for the background acquisition depends on the capture frequency, the frequency of the neon light (normally 60 Hz in the U.S.), the overall ambient light conditions, and the exposure time of the camera sensor as shown in Figure 5.12.

Longer exposure times will lead in general to higher pixel values but will also decrease the achievable frame rate, as described in Section 4.3. The selection of a sufficiently long image sequence in combination with a small averaging constant $\alpha$ (Equation 5.2) can counteract the effects of neon lighting such that a background model with a mean lighting can be achieved. However, the optimal parameter combination must be set individually for a particular environment and cannot be defined in a general sense.

![Figure 5.9: Background detection using the running average method, Case II, images with $thr=60$, frames = 10, $\alpha = 0.05$: (a) mask, (b) edges defined by Canny, and (c) contours.](image)

![Figure 5.10: Background detection using the running average method, Case II, images with $thr=60$, frames = 30, $\alpha = 0.05$: (a) mask, (b) edges defined by Canny, and (c) contours.](image)
Figure 5.11: Background detection using the **running average method, Case II**, images with $thr=60$, frames = 10, $\alpha = 0.50$: (a) mask, (b) edges defined by Canny, and (c) contours.

If the background light conditions change slightly, the running average method can filter these out to a certain degree. In comparison with the snap shot method (Figure 5.8), less background noise is detected in Figure 5.13. This method provides a simple consideration of changing light conditions in the background, by including several images in the calculation. Given the above observations, this method is classified as a medium quality method, and can be used, again depending upon the anticipated changing background conditions during acquisition.

### 5.1.2.5 Background Detection: Mean-Standard Deviation Method

The mean-standard deviation method also uses a sequence of images to calculate the background model. In this method we calculate the mean of the intensity of all pixels and the standard deviation of the pixel values during the background detection sequence. The assumption here is that the brightness of every background pixel varies independently. For every pixel location the sum $S(x, y)$ has to be determined, and for the total number of frames $N$, the mean $m$ can then be determined as $m(x, y) = \frac{S(x, y)}{N}$ and the standard deviation is calculated as:

$$\sigma(x, y) = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (S(x, y)_i - m(x, y))^2}$$

During the object mask detection sequence in the process chain, all pixels in a frame are compared to the calculated standard deviation. To determine if a pixel $p(x, y)$ of a given image frame is within the object mask or within the background, the following is evaluated:

$$|(m(x, y) - p(x, y)| > C\sigma(x, y))$$

where $C$ is a user specifiable constant. Each pixel in a new image that fulfills the above condition will be assigned to the object mask. The most appropriate $C$ value has to be determined during the initialization.
Figure 5.12: Image capturing under neon light conditions (60 Hz lighting): (a) image acquisition at 40 Hz (shutter at 500) and (b) image acquisition at 80 Hz (shutter continuous at 584).

phase in the real environment, however, if $C$ is set to 3, the three-sigma rule is satisfied. Assuming a normal distribution, $3\sigma$ results in less than 1% of the pixels detected as part of the object mask.

If the calculated standard deviation for a pixel is 0, however, the tuning of this method with the sigma parameter is impossible, as shown in Figure 5.14. To avoid this, an additional threshold was implemented,
CHAPTER 5. IMAGE-BASED SYSTEM: PROCESSING AND DATA ANALYSIS

Figure 5.13: Background detection using the **running average method**, **Case III**, images with $thr=30$, frames = 30, $\alpha = 0.05$: (a) mask, (b) edges defined by Canny, and (c) contours.

i.e. the cut-off threshold $co$. All pixels with a value below the value of $co$ will be ignored during the mask calculation. The following settings are used in subsequent examples: (i) threshold parameters of $thr = 60$ and 30, (ii) number of frames = 10 and 30, (iii) cut-off filter $co = 15$, and (iv) $C = 3.0$ (i.e. $3\sigma$). If the method is used as described above, all changes in the background pixel values will be detected. If the background is static the same (positive detection) results as observed with the snap shot method can be achieved, as shown in Figure 5.15. These images illustrate that once this threshold is adjusted properly, only the shadow of the object itself will be detected as an additional artifact. These shadows can subsequently be removed by further increasing the threshold value. In order to maintain important features of the target object, this value should be set as low as possible. For subsequent examples, a value of $co = 15$ was selected.

Figure 5.14: Background detection using the **mean-standard deviation method**, **Case I**, images with $thr=60$, frames = 10, $3\sigma$: (a) mask, (b) edges defined by Canny, and (c) contours.

**Issues with the Mean-Standard Deviation Method** – Even with a background detection sequence of 10 frames, the mean-standard deviation method can eliminate all object movements in the background
model to a reasonable extent. For background objects with higher contrast with the surrounding and much more significant movements, additional information must be provided, e.g., additional frames are needed to calculate the background. Figures 5.16 and 5.17 show that the method performs well even under significant movement, using either $3\sigma$ or $6\sigma$ gave good results. The later figure was also displaced in the vertical direction (note 30 frames were also used in the calculation of Figure 5.17).

Similar to the experiments applied to the previous two methods, a shadow was applied during the background acquisition sequence [Figure 5.4(c)]. Using this method, it was clearly observed that the applied shadow could be eliminated, as shown in Figure 5.18. In comparison with the snap shot and running average methods, using the mean-standard deviation method will provide the best approach when dealing with changing lightning conditions. More importantly, object movement in the background can be eliminated if the movements occur in nearly the same region of the background. This method is therefore deemed of medium-high quality and is the best in comparison with the snapshot and running average methods.

### 5.1.3 Object Mask Calculation

In our processing pipeline, (Figure 5.1), either predefined regions (manual) or the difference between the background model and new images (semi-automatic) are used to define an object mask. Within this object mask, features are subsequently identified and tracked. Using a predefined region approach, the user can constrain the complexity of the image processing step by manually identifying objects or regions-of-interest (ROI). If the manual mode is selected, the masks will be specified by user definable rectangular regions. In the case of manual feature selection, all of the features in the reference images have to be marked manually for each image of the video stream. Especially for larger video sequences containing many desirable features,
Figure 5.16: Background detection using the **mean-standard deviation method**, Case II, images with \( thr=60, \) frames = 10, \( 3\sigma, co = 15 \): (a) mask, (b) edges defined by Canny, and (c) contours.

Figure 5.17: Background detection using the **mean-standard deviation method**, Case II (vertical movement), images with \( thr=60, \) frames = 30, \( 6\sigma, co = 15 \): (a) mask, (b) edges defined by Canny, and (c) contours.

Figure 5.18: Background detection using the **mean-standard deviation method**, Case III), images with \( thr=30, \) frames = 30, \( 3\sigma, co = 15 \): (a) mask, (b) edges defined by Canny, and (c) contours.

this process becomes time consuming and unreliable and alternative approaches were desirable. For this reason the semi-automatic feature detection method was implemented that automatically identifies *good*
features to track within a user definable ROI. Before object tracking functions can be applied to object features they first have to be clearly identified in the target image. The set of all rectangular ROIs specified then collectively defined the object mask. As shown in Figure 5.19 the functions applied during the mask calculation depending on the precalculated background model.

![Figure 5.19: Mask calculation flowchart.](image)

The filters for the automatic mask calculation mode were developed with particular focus on removing background noise caused by changing lighting conditions. While the filters reduce or eliminate noise due to small object movements in the background, the filters will fail when a background pixel value exceeds the defined threshold. The likelihood of this problem occurring increases when the pixel values (object color or texture) of the to be tracked object are nearly equal to the background model. To address this, the threshold value should be selected as small as possible resulting in additional noise mask pixels being generated.

To overcome this problem an additional filter based on the OpenCV function `cvFindContours()` was implemented. This **contour filter** stores all detected contours of a provided image region in a hierarchical tree data structure. The highest level of this tree structure includes all outer contours. Contours that are included and enclosed by another contour will be stored as a child below this parent contour. After generating the tree structure, a contour of a certain dimension in the highest level will be identified as an object that has to be observed. The filter then deletes all contours from the calculated object mask (image region) with a dimension smaller than the predefined $minX_{thr}$ and $minY_{thr}$ cut off dimension.

Figure 5.20(b) shows the output of the algorithm (applied to the image in part a) with the contour of the detected object highlighted in red. All other contours at this level are marked green and can be excluded from further processing. The contour filter method was implemented separately to provide additional object
identification clues. This type of filter could further be enhanced by considering the relation between the rectangular region around the processed contour and the number and position of mask pixels inside the contour region.

5.1.4 Feature Detection

The general structure of our feature detection method is shown in the process flowchart provided in Figure 5.21. The implemented method is based on OpenCV’s `cvGoodFeaturesToTrack()` function, based on the work by Shi and Tomasi (1994). This method searches for corners with large eigenvalues in the image and returns a number of features based on two user definable parameters: (i) the desired quality of a detected feature and (ii) the minimum distance between two nearby features.

![Feature Detection Flowchart](image)

**Figure 5.21: Feature detection flowchart.**

As depicted in Figure 5.22(a) many features would normally be detected that are not part of the moving
object. To reduce the overall feature set, and hence the computation time, the background mask in Figure 5.22(b) is used to focus on features located specifically on the moving object. The result is that elements not located within the object mask are not identified as features to track, as shown in Figure 5.22(c).

A second example shown in Figure 5.23(a)–(d) illustrates the use of user-specified ROIs for defining the best features to track. Part (b) shows the initial features identified across the entire image. However, given the interest in only tracking the glassware, the computer, and the reference pattern, the user might select regions such as those shown in Part (c). Finally, good features are only analyzed within these ROIs (forming the mask), and the resulting, refined good features are shown in part (d) of Figure 5.23.

After thorough evaluation during tracking trials, feature sets minimized to be within object masks may not have stable enough object characteristics to maintain robust tracking over time. Some detected features (characteristics → eigenvalues) significantly change over time or simply disappear resulting in features being lost. To establish more stable characteristics, the feature detection method can be further improved by including the Canny (1986) edge detection algorithm in the process. Canny in this case is used to extract desirable object characteristics that improve the selection of the Shi and Tomasi (1994) good features algorithm to allow us to track seismic movements.

![Figure 5.22: Example of the good feature algorithm by Shi and Tomasi (1994): (a) all features (identified with crosses), (b) computed object mask, and (c) selected features (based on object mask).](image)

**5.1.4.1 Edge Detection Enhancement**

Pixels on the boundary of an object will naturally fall within a zone of gray-level transition. To identify this transition, the slope and direction (magnitude and orientation of the gradient vector) at the edge are of
particular interest. This identification process in image processing is termed *edge detection*. Edge detection is one of the most fundamental concepts for feature identification in image processing. Therefore, it is natural to use a well established edge detection algorithm to enhance the object mask, and hence good features identified for tracking. There are numerous edge detection algorithms available in image processing libraries and packages, most of which are based on the concept of *convolution* with a set of directional derivative masks. One can envision the process, by considering a smooth transition function, for example, a simple S-shaped function. The transition can be definitively located by either the maximum of the first derivative or the zero-crossing of the second derivative of the function.
In our processing pipeline, we use the Canny (1986) algorithm to strengthen the good features selected for tracking. The Canny algorithm uses a Gaussian mask to smooth the processed image and to remove noise during edge detection. The larger the dimension of this Gaussian mask, the lower the detector’s sensitivity to noise. After smoothing the image and eliminating the noise, the edges are identified using the gradient information in the image. This is done with two Sobel (Davis, 1975) operator masks \( G_x \) and \( G_y \) with a size of (3x3) each in the \( X \) and \( Y \) directions. The magnitude or edge strength of the gradient is then approximated as:

\[
|G| = |G_x| + |G_y|
\] (5.5)

Once the gradients in \( x \) and \( y \) directions are known, the edge direction can be calculated as:

\[
\theta = \tan^{-1} \left( \frac{G_x}{G_y} \right)
\] (5.6)

The identified edge direction has to be related to an edge that can be tracked in a pixel-based image. The algorithm checks to see which sector of the image the calculated edge direction is, before the actual pixel will be attached to an edge line.

Hysteresis thresholding is applied with a high and low threshold as a means to eliminate edge separation. Edge separation results in discontinuous edges and appears as 'holes' in the identified edge image. Applying a low and high threshold \( T_{low} \) and \( T_{high} \) can stabilize the output results and repair disconnected edges. If a single threshold \( T_{low} \) is applied to an image and an edge has an average strength equal to \( T_{low} \), then, due to noise, there will be instances where the edge dips below the threshold. Equally, it will also extend above the threshold making an edge appear as a dashed line. To compensate for this, a second threshold is used and all detected edge pixels with a pixel value above the high threshold \( T_{high} \) are clearly identified as edges. Edge values between the thresholds \( T_{low} \) and \( T_{high} \) will be defined as edges if a direct connection to a clearly defined edge pixel can be found. In this case, the quality of the edge detection strongly depends on the amount of noise in the input image, thus, all edges below the low threshold \( T_{low} \) will be ignored.

Figure 5.24 show results obtained with and without the background subtraction approach and Figure 5.25 shows results with different background subtraction thresholds. Comparing first Figure 5.24(a) and(b), it is clear that without a clean background, many background features are detected and it becomes difficult to establish a clear contour of the object that subsequently will be tracked. Figure 5.25 illustrates one problem scenario, where significant noise is introduced within the detected objects mask. These results demonstrate that lower values greatly reduced the tracking accuracy since the increased number of detected edges (also within the mask) impacts the quality of the detectable object contours for moving objects. These figures
also illustrate that if the background filter threshold is set too high, some features of the object will not be detected, where Figure 5.25(b) shows the absence of the support brackets at the base of the computer. These examples highlight the importance of properly configuring the edge detection function for each test environment.

Figure 5.24: Canny algorithm: (a) without and (b) with background subtraction ($T_{low}=30$ and $T_{high}=100$ in both cases).

Figure 5.25: Canny algorithm used with background subtraction thresholds set at: (a) $thr=60$ and (b) $thr=200$ ($T_{low}=10$ and $T_{high}=30$ in both cases).
5.1.5 2D Feature Tracking

Once object features are detected, they are tracked with a set of optical flow algorithms. The optical flow is defined as an apparent motion of image brightness over time (image frames). Two main assumptions were made to establish a set of boundary conditions: (i) a feature pixel \( p(x, y) \) in an image \( I_t \) at time \( t \) has minimal movement between two consecutive images, allowing feature pixel \( p(x, y) \) to be identified within the same region of image \( I_{t+1} \) and (ii) the brightness of each point of a moving or a static object does not change over time. The optical flow constraint equation establishes the basic formulation for all optical flow functions:

\[
\frac{\partial I}{\partial t} = \frac{\partial I}{\partial x} u + \frac{\partial I}{\partial y} v \quad (5.7)
\]

where the variables \( u \) and \( v \) are the components of the optical field in the \( x \) and \( y \) directions, respectively.

For our process chain, the OpenCV function `cvCalcOpticalFlowPyrLk()` with the technique of Lucas and Kanade (1981) was used.

The basis for the Lucas and Kanade implementation is the minimization of a residual function, based on the image velocity \( d \). If an image point is defined by \( u = [u_x, u_y]^T \) on the initial image \( I \), the feature tracking goal is to find the location \( v = u + d = [u_x + d_x, u_y + d_y]^T \) on the subsequent image \( J \) such that \( I(u) \) and \( J(v) \) are similar. The vector \( d = [d_x, d_y]^T \) is the image velocity at a pixel point \( p(x, y) \), also known as the optical flow. Due to the aperture problem, it is essential to define the notion of similarity in the context of a 2D neighborhood. In this context, the image velocity \( d \) can be defined as a vector that minimizes the residual function \( \epsilon \):

\[
\epsilon(d) = \epsilon(d_x, d_y) = \sum_{x=u_x-w_x}^{u_x+w_x} \sum_{y=u_y-w_y}^{u_y+w_y} (I(x, y) - J(x + d_x, y + d_y))^2 \quad (5.8)
\]

where \( w_x \) and \( w_y \) are integer values that define the window size for the neighborhood search.

The algorithm first searches in a given window area around the actually processed pixel and tests for similarity. If the point falls outside the image or if the image patch surrounding the tracked point varies too much between image \( I \) and image \( J \), the result will be features that are lost. If the internal threshold value for \( \epsilon \) cannot be reached during the optical flow calculation, a feature will also be marked as lost.

Sub-pixel accuracy can be achieved using the optical flow method in combination with the OpenCV `cvFindCornerSubPix()` function. The basis for this algorithm can be described by observing Figure 5.26. This method is based on the observation that every vector from the center \( q \) to a point \( p \) located within a neighborhood of \( q \) is orthogonal to the image gradient at \( p \) subject to image and measurement noise.
Therefore:
\[
\varepsilon_i = \nabla I_{p_i}^T \cdot (q - p_i)
\] (5.9)

where \( \nabla I_{p_i} \) is the image gradient at one of the points \( p \) in a neighborhood of \( q \). The value of \( q \) is found by minimizing \( \varepsilon_i \). Consequently, a system of equations can be found by setting each \( \varepsilon_i \) equal to zero:
\[
\left( \sum_i \nabla I_{p_i} \cdot \nabla I_{p_i}^T \right) \cdot q - \left( \sum_i \nabla I_{p_i} \cdot \nabla I_{p_i}^T \cdot p_i \right) = 0
\] (5.10)

where the gradients are summed with a neighborhood ("search window") of \( q \). By defining the first gradient as \( G \) and the second gradient as \( b \), \( q \) can be expressed as \( q = G^{-1} \cdot b \). The algorithm sets the center of the neighborhood window at this new \( q \) and then iterates until the center stays within a set threshold (Intel Research Group, 2004).

Figure 5.26: Geometric basis for sub-pixel accuracy calculation implemented.

5.1.5.1 2D Tracking Sample Results

Figure 5.27 presents sample results of the implemented optical flow method, at frame 2 (a) and at frame 50 (b). Results are shown for the detected features presented in Figure 5.22(a). Note the actual processed pixels shown as yellow squares. Time histories of the objects response for all tracked features are extracted and presented in Figure 5.28. The linear path maintained by the object subjected to simple sliding can be seen in Figure 5.29. The algorithm exhibits stable linear displacement time history response and thus constant velocity, as observed during tests.

5.2 Resolution of the Processed Data

Two issues must be addressed to evaluate the static resolution obtainable from individual images, and hence link this resolution to the processed tracking results. These can be classified into either hardware or software
5.2. RESOLUTION OF THE PROCESSED DATA

Figure 5.27: Optical flow calculation, results for: (a) frame 2 and (b) frame 50.

Figure 5.28: Displacement time history of identified features: (a) x-direction and (b) y-direction.

limited issues. Each are discussed in the following sections.
5.2.1 Hardware-Limited Resolution

The camera hardware will limit the image resolution naturally due to the sensor size and the lens focal length. This resolution is best defined as a function of the perpendicular distance from the object to the camera. For a general combination of camera-lens pairs, to determine the achievable image resolution, the parameters listed in Table 5.1 are needed.

<table>
<thead>
<tr>
<th>Description</th>
<th>Parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>perpendicular distance to feature</td>
<td>$d$</td>
</tr>
<tr>
<td>horizontal viewing angle</td>
<td>$\alpha_x$</td>
</tr>
<tr>
<td>vertical viewing angle</td>
<td>$\alpha_y$</td>
</tr>
<tr>
<td>camera horizontal resolution (in length/pixel)</td>
<td>$res_x$</td>
</tr>
<tr>
<td>camera vertical resolution (in length/pixel)</td>
<td>$res_y$</td>
</tr>
</tbody>
</table>

Table 5.1: Parameters to calculate theoretical resolution $res_{theor}$.

The $res_x$ and $res_y$ of the camera and the camera viewing angles ($\alpha_x$ and $\alpha_y$) are generally provided by the camera manufacturer. Recall that the Basler A301fc cameras have a rated resolution of 658x494 pixels. If the viewing angles are not provided, an approximation can be obtained experimentally as shown in Figure 5.30, using the following simple steps:

1. Place a reference object in the scene with a length $l$ that can be measured, or simply mark a chosen distance on a wall orthogonal to the camera lens axis.
2. Look through the camera and perpendicular to the object or wall, while moving the camera backwards or forwards until the object or wall markings fit exactly within the viewable area of the camera.

3. Measure the perpendicular distance $d$ between the camera and the object. Calculate the viewing angle and the theoretical resolutions with the simple geometric relations noted in Equations 5.11 and 5.12.

$$\alpha = 2 \arctan \left( \frac{l}{2d} \right) \quad (5.11)$$

$$res_{theor} = \frac{2d \tan \left( \frac{\alpha}{2} \right)}{res_{x/y}} \quad (5.12)$$

Figure 5.30: Simple camera-object setup to determine viewing angle.

This procedure can be used to estimate both horizontal and vertical viewing angles. For the camera lens combination used in this work, viewing angles of $\alpha_x = 71.44^\circ$ and $\alpha_y = 54.58^\circ$ were determined using the above procedure. Table 5.2 summarizes the input values used and Figure 5.31 summarizes the theoretical resolutions calculated, as a function of distance $d$ of the camera to the object.

<table>
<thead>
<tr>
<th>$d$ (mm)</th>
<th>$res_x$ (mm/pixel)</th>
<th>$res_y$ (mm/pixel)</th>
<th>$p^1$ (mm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>500</td>
<td>1.093</td>
<td>1.044</td>
<td>44</td>
</tr>
<tr>
<td>1000</td>
<td>2.186</td>
<td>2.089</td>
<td>88</td>
</tr>
<tr>
<td>1500</td>
<td>3.279</td>
<td>3.133</td>
<td>132</td>
</tr>
<tr>
<td>2000</td>
<td>4.372</td>
<td>4.178</td>
<td>176</td>
</tr>
<tr>
<td>2500</td>
<td>5.465</td>
<td>5.222</td>
<td>220</td>
</tr>
</tbody>
</table>

Table 5.2: Resolution for camera lens combination used in this work.

5.2.2 Software (Algorithm)-Limited Resolution

Under static conditions, the optical flow function is the primary limiting aspect which could reduce the resolution of the capture. To evaluate the accuracy of the selected optical flow function, the pixel based tracking algorithm was first tested in a static setting. The test case, as shown in Figure 5.32, consisted of a checkerboard reference pattern that was installed in the scene and used to establish a trackable feature (edge). The trackable feature of interest is outlined by a yellow box (upper left corner of the reference pattern). Subsequently a video stream was acquired at 80fps over a 20 second time period, resulting in 1600 frames. The scene of interest selected in this example is schematically shown in plan and elevation in Figure 5.33.

$^1 p$ here refers to the size of recommended reference pattern when concerned with the shaking camera (refer to Figure 5.36)
CHAPTER 5. IMAGE-BASED SYSTEM: PROCESSING AND DATA ANALYSIS

Figure 5.31: Theoretical resolution as a function of object distance.

Figure 5.32: Photograph of selected pattern feature for evaluating tracking accuracy of the optical flow algorithm.
5.3. SHAKING CAMERA ISSUE AND CORRECTION PATTERNS

The accuracy in the X and Y directions were calculated based on the perpendicular distance between the camera and an object in an orthogonal plane to the corresponding camera. Parameters listed in Table 5.3 were used for this calculation. Once a mapping between pixel and physical dimension is established for the reference plane, the displacement of the tracked pattern-based feature is calculated, as shown in Figure 5.34. As listed in Table 5.4, the per pixel resolution is less than 3 mm/pixel for this example, whereas using the sub-pixel (corner algorithm described in Section 5.1.5), increases the resolution by a factor of approximately three, to within 1.0 mm in each of the x- and y-directions. The stability of the software algorithm in this case assures the resolution of the image-based system is hardware limited, i.e. as limited by the camera and lens combination selected.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>perpendicular distance d</td>
<td>136.73 cm</td>
</tr>
<tr>
<td>horizontal angle of view (camera) α_x</td>
<td>71.44°</td>
</tr>
<tr>
<td>vertical angle of view (camera) α_y</td>
<td>54.58°</td>
</tr>
</tbody>
</table>

Table 5.3: Software-limited resolution sensitivity study: parameters used for optical flow accuracy calculation.

<table>
<thead>
<tr>
<th>Per pixel resolution in X</th>
<th>~ 2.989 mm/pix</th>
<th>Sub-pixel resolution in X</th>
<th>0.897 mm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Per pixel resolution in Y</td>
<td>~ 2.856 mm/pix</td>
<td>Sub-pixel resolution in Y</td>
<td>0.857 mm</td>
</tr>
</tbody>
</table>

Table 5.4: Software-limited resolution sensitivity study: calculated resolution in the orthogonal plane.

5.3 Shaking Camera Issue and Correction Patterns

Tracking features and objects due to seismic induced movements adds additional challenges since a broad range of objects will move, surfaces assumed rigid in the context of the background detection algorithms shake, twist or warp while the visual sensor array itself may get displaced. This is indeed a very practical and fundamental issue that needs to be considered if the concept of vision-based sensors is to be adopted for seismic risk assessment. Herein, we termed the movement of the camera array itself, the camera shaking problem. An implemented algorithm is discussed that allows for compensation of the error introduced by the movement of the sensor itself.

Our approach is to first detect camera specific movements based on clearly identifiable reference patterns that are inserted into the environment. Subsequently, we correct for the position of the tracked object features based on the reference pattern movement. Clearly, this approach is based on the assumption that a rigid object can be identified within the tracked environment and instrumented with the reference pattern before image data is acquired. In the presented examples, we have placed the pattern on a tabletop surface.
and calculated the relative movement between the camera frame and the tabletop (e.g. refer to Figure 5.32 or 5.33). Subsequently, the movement of an object located on the tabletop can be corrected. This, of course, assumes that the location of the reference pattern results in movements of comparable frequency and amplitude to that of the shaking camera (i.e. the frequencies of the mounting point of the camera is close to the frequency of the reference pattern location). This is reasonable if sufficiently stiff mounting brackets are used for the cameras, and a stiff structure within the scene is selected for the reference pattern location. Given the flexible nature of most buildings (relative to interior support elements), such placement will also help promote filtering of the seismic motions, minimizing the input seismic energy into the camera and reference pattern support structure.

The selection of an appropriate pattern is very important, and therefore, different configurations were evaluated to determine the quality of the correction results in the pixel based pattern detection pipeline. Those patterns evaluated are shown in Figure 5.35. A simple yet versatile pattern layout that worked well under most conditions is show in Figure 5.36. Design considerations when selecting this (or an alternative) pattern include the following:

- High contrast colors (black and white) should be used.

- Clear edges should be drawn, such that easily detectable features are present on the pattern (90° is optimal).

- The number of edges should be greater than 10 (the more the better).

- The distance between the defined edges should be large enough to identify at least three features in one direction of the pattern.

- At least two separate regions around the pattern should be defined to provide a clear pattern area detection and to avoid interferences with background pixels.

The recommended checkerboard pattern within two enclosed square regions aids with the identification of the reference pattern itself even within noisy environments (the final point in the above list of design considerations). To maintain the minimal hardware-limited resolutions presented in Section 5.2.1, square dimensions of $p \times p$ were calculated for a range of camera-to-object distances and are presented in Table 5.2. The pattern is defined such that each square within the pattern covers at least 40x40 pixels in an orthogonal projection plane. The pattern length $p$ is rounded up to the next even number.
5.4 Poor Features to Track

There are two pervasive examples of features within a scene that result in poor tracking results: (i) oscillating lighting conditions and (ii) object occlusion. These were observed in a number of our experiments and are worth enumerating in select examples.

5.4.1 Oscillating Lighting on a Feature

As described earlier, the optical flow tracking method is based on clearly identifiable features with a strong eigenvalue. Objects of low eigenvalues are difficult to track, and this difficulty is increased when their strength fluctuates during an image sequence. A prime candidate for such a phenomena is the occurrence of oscillating light intensity on a feature. One example consistently observed was that of glassware being tracked under a neon light (Figure 5.37). The object of interest is shown in the upper left shelf of this image, where part (a) and (b), obtained at different times, disagree on the four features identified as good features to track.

A 10 second time history of the resulting x- and y-displacements during dynamic input is shown in Figure 5.38. In the x-direction, one of the four points shows inconsistent identification during the time history (as observed by the sharp vertical offset in one trace), while two of the four y-direction traces show inconsistencies. This phenomena is caused by the oscillation of the reflections of light on the glassware, as they are frequently detected as good features, regardless of their random behavior. Additional tests further confirmed that weakly defined features may move across the entire scene. Under these conditions, the detected feature as well as the surrounding pixels in the defined search window will change their appearance "randomly," preventing reliable tracking of the preselected features.

Clearly it is desirable to avoid these errors by using a higher quality tracking parameter during the configuration phase of the feature detection step. If objects with weak features must be tracked, we recommend features are artificially enhanced through the addition of markers (such as the spherical markers used in the work presented in Chapter 3). For 2D tracking, a simple round target as shown in Figure 5.35 is reasonable.

5.4.2 Occlusion of a Feature

Occlusion is a practical phenomena that can only be minimally avoided by modifying the location and/or number of cameras viewing a scene. However, during shaking occlusion is difficult to resolve, and can only be nominally considered as one lays out the camera array to define the field-of-view of interest.

A sample of the occlusion problem and its potential resulting error in tracking is illustrated in Figure 5.39.
This figure illustrates a series of images of a corresponding feature tracked by four cameras simultaneously. Cameras 0 and 3 detect a feature in image 0 between 3 edges of the object. As long as all three edges are visible to the respective camera the feature can be detected and tracked very well. However, the tracking algorithm is affected by occlusion of the object since feature characteristic change (see frame 1200 cam0 and cam3). In the presented test sequence, occlusion of the object due to its own movement begins at approximately frame 400, resulting in the loss of important features for camera 1 and 2. This is partially due to the lower quality level of the detected features (see feature comparison between cameras for frame 0), which becomes exasperated as the object rotates and occludes the cameras view. Figure 5.40 highlight these effects in the x- and y-direction time histories, where the change in slope and divergence of the measured displacements of camera 1 and 2 are seen beyond frame 400.

5.5 Summary Remarks

In this chapter, we describe a fully image-based processing chain for tracking dynamic movements in a scene. The pixel-based image processing chain (PIPC) is realized by using pixels to identify objects and features. The basis for the implementation relies upon the following steps: (i) camera calibration and correction of the image, (ii) robust background detection, (iii) mask-based object identification, (iv) feature detection within the object mask, and (v) 2D feature tracking using an optical flow methodology. A corner algorithm was implemented to increase the resolution to sub-pixel level. Using the camera-lens configurations described in Chapter 4, the resulting implementation was able to achieve better than 1.0mm resolution, within anticipated distances of capture \(d \approx 130\) cm. Given the potential for shaking of the camera sensor during harsh seismic movements, we also introduce a reference pattern correction step and a methodology to select and apply it in the field.
Figure 5.33: Plan and elevation schematic of scene used to evaluate the tracking accuracy of the optical flow algorithm.
Figure 5.34: Software-limited resolution sensitivity study: time history of (a) x-direction and (b) y-direction pixel data, during static collection.

Figure 5.35: Evaluated test patterns.
Figure 5.36: Pattern layout recommended for *shaking camera* correction.
Figure 5.37: Example of poor features to track (glassware in changing lighting conditions): (a) at $t=0$ seconds and (b) $t=10$ seconds.
Figure 5.38: Example of poor features to track (glassware in changing lighting conditions): (a) x-direction and (b) y-direction time histories.
Figure 5.39: Feature tracking result using four cameras at: (row 1) frame 0, (row 2) frame 400, (row 3) frame 800 and (row 4) frame 1200.
Figure 5.40: Four camera comparison over 1200 frames: (a) x-displacement and (b) y-displacement.
Chapter 6

Conclusions and Recommendations for Future Work

6.1 Conclusive Findings

Advances in technology combined with increased public concern over life and economic loss due to earthquake events have motivated us to consider vision-based (camera) sensors for monitoring seismically induced movements. In this report, we evaluate a vision-based system based on reduced light capture, and design and evaluate a system based on a full spectrum of light capture. We term the former a light-based system and the later an image-based system.

The light-based system is found to be very suitable for laboratory experimental use, where it is relatively easy to control ambient lighting and assure clean environmental (background) conditions. Shake table experiments are conducted and the high resolution four camera light-based system is used to track the three-dimensional motions of various types of equipment and building contents commonly found in biological and chemical science laboratories. Positional time histories are compared with measurements obtained using conventional analog sensors. Maximum deviations between the two measurement techniques were calculated as less than 2 mm in all cases and on average less than 1 mm.

The image-based system is designed to address the noisy conditions frequently encountered in general building interiors. The specialty hardware and software design results in an optimally low cost image acquisition system. Performance tests of the system assure that the required resolution and speed are attainable using our unique multi-threaded software platform, which avoids busy waits and optimizes the computational power of a standard server style computer. The image processing framework developed for tracking dynamic movements is realized using a pixel-based image processing chain (PIPC), where pixels are used to identify objects and features. The basis for the implementation relies upon the following steps:
(i) camera calibration and correction of the image, (ii) robust background detection, (iii) mask-based object identification, (iv) feature detection within the object mask, and (v) 2D feature tracking using an optical flow methodology. A corner algorithm was implemented to increase the resolution to sub-pixel level. Using the hardware configuration selected, the resulting implementation was able to achieve less than 5.0mm resolution, for long capture distances (approximately 2.5 m) and less than 1.0mm resolution, for near capture distances (approximately 1 m). More importantly, the hardware limited the resolution, rather than the software (image-processing) developed.

6.2 Recommendations for Future Work

This study has shown that with the proper hardware selection and computer vision algorithms, the tracking of objects due to seismic movements can be realized. Once object types are determined and their tracking is possible, the detection of hazardous situations and secondary risks becomes viable.

For large-scale deployment of such a system a number of practical challenges must be addressed. Of foremost importance is reduction in size and complexity of the acquired data. A first step would be the implementation of carefully calibrated event-based trigger functions that independently initiate and terminate data acquisition. At the same time new compression schemes enabling lossless compression of the acquired image or video data are desirable if the application type requires the archival of all data to facilitate subsequent forensic analysis. If archival data is not required, real-time damage and tracking classification schemes are desirable that automatically evaluate, identify and categorize damage and situational types. These have to be selected with the objects of interest and their associated movement thresholds in mind. Some consideration must also be given to whether or not this is an unsupervised or supervised classification approach. The value of this information will greatly depend on the availability of proper methods for uncertainty analysis that will contribute towards establishing statistical confidence in tracking estimates. The derived information may subsequently be used to create a link to decision support and early warning systems.

Depending on the purpose of the system and the type of data records to be archived, two models coarsely characterized as "store-and-forward" and "process and dispatch" may be considered. While the former is based on acquiring and storing data for subsequent processing, the later would use real-time processing strategies to extract crucial information on the fly, and dispatch the results to a decision support system.


Appendix A

Image-Based System: Calibration

The recovery of three-dimensional information from a set of two-dimensional images requires the use of an accurate, and fast calibration technique. Although diverse techniques have been developed for single and multiple camera configurations, selection of the most appropriate task-specific technique is a non-trivial exercise. In this chapter, we review calibration techniques, their respective application domains and associated boundary conditions. Boundary conditions of particular importance include, desired capture volume, available time budget, required accuracy, number of cameras and type of feature detection. Commonly used calibration tools are discussed, including their specific features and target applications. A methodology to determine the suitability of a calibration tool for a specific setup and requirements using a synthetically generated environment, is presented. Finally, a multi-camera calibration tool is evaluated for its robustness in determining important calibration information under a number of envisioned camera configurations. The goal of this study is to aid the reader in the selection of the most suitable calibration techniques and tools appropriate for the task at hand.

A.1 Motivation and Related Work

Applications ranging from entertainment to scientific visualization require characterization of objects and phenomena occurring in our three-dimensional (3D) world. A broad range of image capturing devices allow the collection of such 3D information. A fundamental problem that needs to be addressed is the recovery of depth information lost during projection. Although diverse techniques have been presented to solve this problem, a primary factor in the selection of a suitable approach to perform 3D recovery is the final application of this information. Additional aspects to consider include the constraints affecting the acquisition system, such as: (i) characteristics of the environment, (ii) capture volume, (iii) time budget,

\footnote{Results described here are presented in the citation Villa-Uriol et al. (2004)}
(iv) required accuracy and (v) desired speed.

Depending on the acquisition configuration, either monoscopic images or video sequences are used (Gibson et al., 2003). Alternatively, multiple static images or video sequences may be used. If an engineering length scale is required, during the calibration stage, information regarding the characteristics of the capturing system, camera locations and camera internal features must be incorporated (Trucco and Verri, 1998; Faugeras, 1993).

The choice of a calibration technique or tool will determine relevant aspects in the posterior 3D points recovery, such as the achievable accuracy and frame rate. Depending on the camera lens, accounting for radial distortion (or not) is also decisive for accuracy. Sometimes only uncalibrated sequences are available and traditional calibration techniques can not be used, resulting in additional computational cost for the reconstruction (Mohr et al., 1995; Zhang and Sexton, 1995).

The objective of this chapter is to provide an overview of many of the available techniques and tools that can be used when a calibration stage is needed for a specific application. We intend to pinpoint the decisions that need to be made and that will affect the quality of the final outcome, critical in the use of images to describe 3D information. First, fundamental terminology is defined, followed by a classification of calibration methods from a user’s perspective and subsequently a generalization of the required steps to perform camera calibration. A subset of the most commonly available calibration packages is described, together with their advantages and disadvantages. Finally, a multi-camera calibration tool is evaluated for its robustness in determining important calibration information under a number of envisioned camera configurations.

### A.2 Terminology

To reconstruct a scene from images, the relationship between the coordinates of a set of points in 3D space with the coordinates of their corresponding image points must be established. Equations for such a relationship are written in a world reference frame. To assist with defining these relationships and relating them with camera properties, camera’s characteristics are generally grouped into *intrinsic* (internal) and *extrinsic* (external) parameters, as shown in Figure A.1.

The internal parameters describe how the camera forms an image and offer a relationship between the image and the camera. These parameters are important for linking the pixel coordinates of an image point with the corresponding coordinates in the camera reference frame. They can be defined as the set of parameters needed to characterize the optical, geometric and digital characteristics of the given camera.
The intrinsic attributes are focal lengths $f$, $f_x$, $f_y$, pixel sizes $s_x$, $s_y$ in the $x$ and $y$ directions, aspect ratio $a$ computed as $s_y/s_x$, principal point $P$, image center $(o_x, o_y)$ and radial distortion coefficients $k_1$ and $k_2$. While $f$ represents the distance between the center of projection and the retinal plane specifying the perspective projection, $f_x$ and $f_y$ are the lengths in effective horizontal and vertical pixel size units. $P$ specifies the coordinates $(p_x, p_y)$ of the center of the lens, which is the piercing point of the camera’s coordinate frame $z$-axis with the camera’s sensor plane. Finally $k_1$ and $k_2$ describe the geometric distortion introduced by the optics of the lenses.

Figure A.1: Fundamental terminology in world and camera coordinate systems.

The extrinsic parameters define the camera’s position and orientation. These relate the camera to the real world, and are defined as any set of geometric parameters that uniquely identify the transformation between the unknown reference frame and a known reference frame, called the world reference frame. The extrinsic parameters are (a) 3D translation vector $\vec{t}$, providing the translational components for the transform between the world and camera coordinate frames, and (b) $3 \times 3$ rotation matrix $R$, a product of matrices $R_x$, $R_y$, $R_z$ representing the rotation angles for the transform between the world and camera coordinate frames.

The principal point, effective horizontal length, aspect ratio and the image centers are required for the transformation between the frame coordinates and pixels coordinates. The task to estimate the values of the intrinsic and extrinsic parameters of a single camera or a set of multiple cameras is termed camera calibration.
A.3 Classification of Calibration Methods

Accuracy of the final recovered 3D dataset depends upon the accuracy of the calibration (Lee and Jeong, 2000). Ideally, the calibration tolerance should be one or two magnitudes smaller than the desired tolerance when 3D acquisition is conducted (Trucco and Verri, 1998). For this reason, ensuring an accurate calibration is essential and it directly derives from the qualities of the acquisition system, characteristics of the captured environment and the chosen calibration technique.

Strategic placement of cameras, acquired image resolution and quality of camera lens(es) affects the acquisition system in terms of space. At a temporal level, required acquisition frame rate impacts the amount of information that needs to be stored and analyzed to obtain an accurate calibration.

Several calibration techniques are widely used in different application domains. In Figure A.2, we depict a classification of available calibration methods from an application user’s perspective. We organize them into two broad categories, based on the final usage and the importance of accuracy to these applications. In general, applications used for scientific and engineering purposes require significantly more accuracy than those in the entertainment industry. Thus, our focus will be on the scientific application domains, where a set of features such as color and shape guides the calibration process. We differentiate between systems that demand manual user intervention of features and those that are totally automated. Although manual identification of features simplifies the process, it is error-prone and may result in being more time consuming than the calibration process itself. Hence, automated or semi-automated approaches are desirable.

Figure A.2: Classification of calibration methods.
A significant contributing factor within the method of calibration is the number of cameras that are used. Calibrating multiple cameras necessitates several additional steps. First, each camera must be pointing at the capture volume of interest with significant overlap. Since calibration is based on establishing image and feature correspondences, it is desirable that a common set of features is visible from most of the cameras. Features extracted from the images can be categorized very generally as either: (i) marker-based or (ii) pattern-based. Marker-based features may be either active or passive within the scene. Patterns may include multiple or single basic geometry primitives, however, generally include very simple and well-defined color transitions (e.g. black on white checkerboards).

A.4 Techniques and Steps Involved

Most scene reconstruction and machine vision projects treat calibration as a preprocess that involves several steps, including camera placement, data acquisition, feature detection and feature matching (Ernst et al., 1999). Figure A.3 illustrates commonalities and discrete differences for the most common techniques. The calibration process generally begins with a careful definition of acquisition objectives, enabling the user to decide on the required acquisition area or volume as well as the needed spatial and temporal resolution.

These requirements aid in the selection of the required sensor configuration, varying between a single camera and multi-camera array. If a multi-camera array is used to capture time varying effects, additional complexity is introduced since precise synchronization of all cameras will be required. Ideally, this synchronization will occur at acquisition time via an external trigger signal applied to all cameras in the array or properly configured software-based barriers.

The selection of the most appropriate calibration technique itself, namely active or passive markers, a reference pattern or automatic/manual selection of matching feature points, is a function of the environment where the capture is performed and the used acquisition system. Close attention has to be paid to other conditions such as visual noise introduced through lighting, shadows or occlusion. If it is necessary to perform background substraction for a static uniform background, a very simple per-pixel operation would be enough. On the other hand, a dynamic and unpredictable background increases the complexity and has a great impact on the amount of processing that needs to be done. As expected, the reconstruction results depend on the accuracy of the calibration and accordingly, this influences the user’s choice in respect to the number of cameras, their arrangement, calibration method and pattern to be used.
A.4.1 Image Acquisition System Design

Single camera or multi-camera configurations can be used to generate the reference images needed for calibration. For a single camera, this may simply involve acquiring individual still images either from multiple viewpoints for a static scene, or from a fixed viewpoint for a time varying scene (i.e. a moving object). Camera arrays on the other hand require a mapping between features present in each of the available views of the scene. Using the corresponding points information, the implicit and explicit parameters of the cameras can be computed. Generally some image processing such as background substraction or filtering will be required to remove unwanted noise. All cameras need to be synchronized in order to ensure frame correspondence between cameras, making certain that a given frame number is captured at the same instant of time by every camera. For this type of a setup, synchronization is either done in software or in hardware and a high frames count from each camera is usually required for calibration (Barreto et al., 2004). Therefore, an automated processing sequence is desirable that facilitates time efficient processing of the reference images.

Passive or active markers can be introduced in the scene to simplify the identification of feature cor-
respondences between images (Figure A.4). This involves introducing a known pattern into the scene and to subsequently identify its features, such as known edges and colors to establish correspondence between different images. The reference markers can be useful beyond calibration, to accomplish object tracking itself, as illustrated in Figure A.4.

(a) Wand for dynamic calibration.  
(b) L-frame for static calibration.  
(c) Using passive markers.  
(d) In a video sequence (image courtesy of Gibson et al. (2003)).

Figure A.4: Marker-based (a, b) and Feature (c, d) tracking.

A.4.2 Selection of Calibration Patterns

Camera calibration usually relies on the extraction of features from the available input images. A simple method to allow unit-based calibration, consists of having a reference object of known dimensions present in all the views. This reference simplifies the task of feature identification and allows for efficient calibration.
The approach for single camera calibration uses planar objects with a very precisely known geometrical pattern. In this scenario, the calibration process and setup are relatively simple, as opposed to multiple camera calibration, where three dimensional cuboidal (Tsai, 1986) or spherical (Agrawal and Davis, 2003) reference objects are used, as shown in Figure A.5.

Patterns are frequently in black and white to take advantage of the high contrast during feature detection. The simplest and most common is the use of grid-based (Jeong et al., 2002) or checkerboard patterns. However, conical patterns such as circles, have also been successfully applied (Heikkilä, 2000). Static markers, including bright color stickers, may also be introduced in the scene to add distinct reference points.

A different approach which does not utilize any reference object is called self-calibration and provides a unit-less calibration result. Its operation is based in establishing image-to-image correspondences of a set of markers or features (Gibson et al., 2003). One option consists of using a laser pointer as a passive dynamic marker and to track its position between frames (Sbovoda, 2004).

A.5 Common Calibration Packages

A number of common calibration packages exist, that mainly differ in the level of accuracy that can be obtained. Some of those are discussed in the following sections.

A.5.1 Static Pattern Calibration

The calibration tools discussed in this section use a static, regular chessboard pattern as a reference. One example is the Bouguet camera calibration toolbox (Bouguet, 2004). This toolbox was developed in C/C++,

(a) Circular pattern (image courtesy of Heikkilä (Heikkilä, 2004)).
(b) Square pattern.
A.5. COMMON CALIBRATION PACKAGES

using Intel’s Open Source Computer Vision Library (OpenCV). This tool can calibrate a video camera accurately in a matter of just a few seconds. A flat checkerboard pattern in combination with knowledge about the dimensions of each square are used to enable automatic edge detection and computation of the intrinsic camera parameters (focal length, principal point, distortion coefficients) as well as the extrinsic parameters (3D position of the pattern for each image). Once calibration is complete, image distortion can be removed in real-time. The lens distortion model consists of two terms: a radial distortion term (up to the fourth order) and a tangential distortion term consisting of two scalars for encoding the angular orientation of the focal plane with respect to the sensor plane. Consequently, the lens distortion model is parameterized using four scalar coefficients. One drawback of this toolbox is that it only supports the calibration of one camera at any given time. A Matlab implementation is available as well but is limited by performance bottlenecks.

Heikkilä and Silven have developed another Matlab based open-source calibration tool (Heikkilä and Silvén, 1997; Heikkilä, 2000; Heikkilä, 2004). The algorithm is based on a circular pattern and uses a new bias correction procedure for circular control points and a non-recursive method to reverse the distortion model. A drawback of this approach is that it does not support automatic corner detection, requiring that users specify values, identifying the corners of the checkered pattern. Zhang (1999, 2000) developed a flexible camera calibration technique. However, it does not have corner extraction and hence, user input is required to provide this information. The Tsai camera calibration application is considered one of the classic tools and is based on Tsai’s algorithm for the perspective projection camera model (Tsai, 1986, 1987). The foundation for this algorithm is the pinhole model for perspective projection. This tool as well does not include an automatic corner extractor.

A.5.2 Dynamic Calibration - Single Camera

Dynamic calibration encompasses the scenario where either a single moving camera (or video camera) is used to collect a sequence of images over a period of time. Commonly termed video sequence calibration, this approach involves identifying a number of features that are shared between consecutive frames. These features are then tracked from frame to frame, and thus form the points of correspondence for calibration. A drawback of this method is that commonly, extended image sequences (hundreds of frames) are required. If the user must identify the different features in each frame, then the delay in the process is dominated by the user’s input time. Hence, an automated system is highly desirable in such video sequence calibration approaches and one example of such an application is discussed in Gibson et al. (2003).
A.5.3 Multi-Camera Calibration

In this section, we describe selected multi-camera calibration application frameworks. One such package, developed within MatLab and C is the EasyCal (Barreto et al., 2004) calibration software. The engine behind EasyCal is an extension of the camera calibration toolbox developed by Bouguet (2004). The main advantage of this tool is the modular multi-image feature, while a primary drawback is the large number of images required. The authors note that a sequence of at least 1000 images at a relatively high frame rate ($\geq 15$fps) are required for suitable results (Barreto et al., 2004). In addition, images of a basic chessboard design pattern from at least two cameras are needed. A second drawback is the approach expects the room lighting to be turned off while acquiring the point images as calibration data. This may not be practically possible for many applications.

The Multi-Camera Self-Calibration Toolbox (Sbovoda, 2004), developed at the Computer Vision Laboratory of the Swiss Federal Institute of Technology, incorporates each calibration phase within the MatLab toolbox. A minimum of three cameras and a diffused laser pointer, which is tracked during calibration, are required for this algorithm. This software is well suited for calibrating multiple cameras at the same time. An important advantage of this toolbox is that it is completely automated. The light source is moved throughout the anticipated field of view during the calibration phase, generating a set of images of the sources’ path. The light source does not necessarily need to be visible from all cameras during the entire calibration capture, as the algorithm compensates for occluded or missing points. The only drawback of this application is that it requires a relatively dark surrounding, such that the light source maintains a good contrast with the background for tracking purposes.

A.5.4 Case Study of a Calibration Tool

To test the feasibility of the Multi-Camera Self-Calibration Toolbox (Sbovoda, 2004), we evaluate its capabilities considering a variety of envisioned cameras and locations. The purpose of this work is to test if this tool is suitable for a given application, in terms of volume to be captured, achievable accuracy, robustness, storage needs and calibration time budget. These parameters are used as a basis to determine if the combination of camera setup and calibration tool studied meets certain requirements. The camera setup is defined by a set of intrinsic and extrinsic parameters (camera placement) of all the cameras in consideration. A general methodology is presented to test different pairs of camera setups and calibration tools to help in the decision process when designing a multiple camera capture system. Our approach utilizes a synthetically generated
A.5. COMMON CALIBRATION PACKAGES

multi-camera capture setup recreating real world conditions to test the calibration tool, in combination with a synthetic calibration sequence.

A three-dimensional modeling tool was used to generate arbitrary scenes with a variable number of \( n \) cameras (3 to 12) with known intrinsic and extrinsic parameters. In this section we present an analysis of 12 setups: (a) four setups of four cameras each, (b) two setups of eight cameras and (c) six different setups of 3, 5, 6, 7, 10 and 12 cameras each. A set of snapshots showing top and perspective views of a subset are included in Figure A.6. More specifically, there is one case with \( n = 12 \) placed in two concentric circles at two different heights and at 30 degrees angular intervals (Figure A.6 (c) and (d)). Two different cases with \( n = 4 \) are also included, corresponding to cases (ii) and (iii). Case (ii) shows an arbitrary placement of cameras, while in case (iii) they are placed in a single half of the hemisphere covering the scene.

The multi-camera self-calibration toolbox requires a calibration sequence in which each of the cameras captures a colored-light source moving through the capture volume. In our tests, each of these calibration sequences consisted of 300 frames with a resolution of 640x480 pixels, covering a capture volume of 10x10x10 units with a red-light source of radius 0.5 units. The viewing volumes for all studied camera configurations were of similar size.

To begin our study, the calibration accuracy of each camera setup needs to be determined. To perform this task, we analyzed the differences between the calibration results obtained from the calibration tool and the known, synthetically generated setup. Intrinsic and extrinsic camera parameters of each of the cameras are obtained and compared, namely, focal length \((f_x, f_y)\) and camera location expressed as a rotation \((\text{rot})\) and a translation vector \((\text{trans})\). In the case of focal length the error between the calibrated case and the synthetic setup is computed as:

\[
\text{error}\% (f_x) = \frac{f_{x, \text{cal}} - f_{x, \text{syn}}}{f_{x, \text{syn}}} \times 100 \\
\text{error}\% (f_y) = \frac{f_{y, \text{cal}} - f_{y, \text{syn}}}{f_{y, \text{syn}}} \times 100
\]

where \(f_{x, \text{cal}}\) and \(f_{y, \text{cal}}\) are the computed values for \(f_x\) and \(f_y\) by the calibration tool, and \(f_{x, \text{syn}}\) and \(f_{y, \text{syn}}\) are the known values for \(f_x\) and \(f_y\) obtained from the synthetic scene setup. The square error is used for camera locations:

\[
square \text{error} (\text{rot}) = \frac{1}{9} \left( \sum_{i=1, j=1}^{3,3} (r_{\text{cal}, ij}^2 - r_{\text{syn}, ij}^2) \right)
\]

-109-
\[
\text{square error(\text{trans})} = \frac{1}{3} \left( \sum_{i=1}^{3} (t_{\text{cal}}^{i^2} - t_{\text{syn}}^{i^2}) \right) \tag{A.4}
\]

where \(r_{\text{cal}}^{ij}\) correspond to the \(i\) by \(j\) elements of the rotation matrix \(R\) and \(t_{\text{cal}}^{i}\) to the translational components of the normalized vector \(\vec{t}\), characterizing the transformation between the world and camera coordinate frames, obtained from the calibration process. \(r_{\text{syn}}^{ij}\) and \(t_{\text{syn}}^{i}\) are the known values extracted from the synthetic scene.

The plots in Figure A.7 show a sample of the computed errors for each of the individual cameras in the twelve camera setup shown in Figure A.6(c) and (d). The impact each camera has on the computation of the mean error for the camera setup (mean \(f_x\), mean \(f_y\), mean trans and mean rot) are also noted. Results presented in Figure A.7 illustrate the high accuracy obtained, in terms of focal length and translation and rotation matrices. The mean error (amongst all cameras in this setup) was very small 0.25\%. In addition, the maximum error observed, in terms of focal length was 0.8\%.

A comparison among all twelve tested setups in terms of the mean error for mean \(f_x\) and mean \(f_y\) and the mean square error for mean trans and mean rot are presented in Figure A.8. These are presented as the mean of all cameras used in the setup considered. Among the four different setups with four cameras each, there are some differences in the observed error. For example, the mean focal length error ranges from 0.2\% to 1.6\% for cases (ii) and (iii). This behavior can be explained with the different camera placement used for each setup (Figure A.6). Case (iii) of the four-camera setup was designed with all cameras located on the same side of the hemisphere, resulting in more overlap of the viewing volume. As one might expect, as the number of cameras increase in a setup, the computed mean error decreases.

It also can be observed that a high correlation exists in the error in focal lengths among the different setups, which is in the order of 0.4\%. In addition, rotation and translation square errors are very small, with general magnitude below the range of \(10^{-5}\). Analysis of different camera configurations demonstrates that the expected calibration errors introduced by the calibration tool can be bounded. As such, this methodology provides a platform to test and compare arbitrary calibration tools and camera setups.

### A.6 Summary Remarks

The objective of this chapter is to provide an overview of many of the available techniques and tools that can be used when a calibration stage is needed to recover 3D information from a set of two-dimensional images. This is well recognized as a critical, yet tedious and non-trivial part of the processing pipeline.
However, until now, a unified discussion of this topic, where the advantages and disadvantages of a variety of techniques are explored, has not been presented in the literature. Therefore, we first provide a classification of calibration methods from a user's perspective and subsequently generalize the required steps to perform camera calibration. A subset of the most commonly available calibration packages are summarized. The goal of this literary review is to aid the reader in the selection of the most suitable calibration techniques and tools appropriate for the task at hand. Finally, a methodology to determine the suitability of a calibration tool for a specific setup and requirements, is presented. It is concluded that with the tested tool, low errors for intrinsic and extrinsic camera properties may be obtained, for the range of camera configurations considered.
Figure A.6: Different views of the spatial configuration for an $n$ cameras setup.
A.6. SUMMARY REMARKS

Figure A.7: Computed errors for a 12 cameras setup.

Figure A.8: Computed mean errors for all tested setups.