Multi-Camera Light Field Capture

Synchronization, Calibration, Depth Uncertainty, and System Design

Elijs Dima

Department of Information Systems and Technology
Mid Sweden University

Licentiate Thesis No. 139
Sundsvall, Sweden
2018

Mittuniversitetet
Informationssytem och -teknologi
SE-851 70 Sundsvall
SWEDEN

ISBN 978-91-88527-56-1
ISNN 1652-8948

Akademisk avhandling som med tillstånd av Mittuniversitetet framlägges till offentlig granskning för avläggande av teknologie licentiatexamen **den 15 Jun 2018** klockan **13.00** i sal **L111**, Mittuniversitetet Holmgatan 10, Sundsvall. Seminariet kommer att hållas på engelska.

Tryck: Tryckeriet Mittuniversitetet

As geographers, Sosius, crowd into the edges of their maps parts of the world which they do not know about, adding notes in the margin to the effect, that beyond this lies nothing but sandy deserts full of wild beasts, unapproachable bogs, Scythian ice, or a frozen sea, so, in this work of mine, in which I have compared the lives of the greatest men with one another, after passing through those periods which probable reasoning can reach to and real history find a footing in, I might very well say of those that are farther off, beyond this there is nothing but prodigies and fictions, the only inhabitants are the poets and inventors of fables; there is no credit, or certainty any farther.

- Plutarch, Lives of the Noble Greeks and Romans

DON'T PANIC!

- Douglas Adams, The Hitchhiker's Guide to the Galaxy

Acknowledgements

I would like to thank my supervisors, Prof. Mårten Sjöström and Dr. Roger Olsson, for their guidance, support, and insights into the research process, and for the numerous enjoyable and sometimes downright weird "Friday discussions" that took place even on the coldest of Mondays. Of course, thanks must also go to my colleagues, Yongwei Li and Waqas Ahmad, for the invaluable assistance and friendship during the research and studies. Thank you for forming a truly enjoyable, open, and honest research group that I am glad to be a part of.

Special thanks to Jan-Erik Jonsson and Martin Kjellqvist here at IST for their help with the project and even more so for the always-enjoyable lunch break discussions. Thanks to Dr. Benny Thörnberg for the advice and feedback on this work. Thanks also to Mehrzad Lavassani, Luca Beltramelli, Leif Sundberg and Simone Grimaldi, for making the first year and all our shared courses eventful and interesting. Thanks to the past and present employees at Ericsson Research, both for hosting me in their research environment at Kista at the start of the project, and for the regular discussions of the multi-camera system's progress and development. Thanks to Lars Flodén and Lennart Rasmusson of Observit AB for their insights into the engineering goals and constraints of multi-camera applications. Thanks to Prof. Marek Domański and Prof. Reinhard Koch for hosting me in their respective research groups at Poznan and Kiel; both have been valuable sources of insight into Light Fields and camera systems, and also provided me with exposure to culturally and organizationally diverse research practices and environments. Thanks to Prof. Jenny Read of Newcastle University for the discussions on human vision and perception, and the arcane mechanisms through which we humans create a model of the 3D world. Finally, I thank all the people in the IST department for the excellent workplace atmosphere, *fika* discussions, and the administrative help.

This work has received funding through grant 6006-214-290174 from Rådet för Utbildning på Forskarnivå (FUR), Mid Sweden University, and through the LIFE project grant 20140200 from the Knowledge Foundation, Sweden.

Abstract

The digital camera is the technological counterpart to the human eye, enabling the observation and recording of events in the natural world. Since modern life increasingly depends on digital systems, cameras and especially multiple-camera systems are being widely used in applications that affect our society, ranging from multimedia production and surveillance to self-driving robot localization. The rising interest in multi-camera systems is mirrored by the rising activity in Light Field research, where multi-camera systems are used to capture Light Fields - the angular and spatial information about light rays within a 3D space.

The purpose of this work is to gain a more comprehensive understanding of how cameras collaborate and produce consistent data as a multi-camera system, and to build a multi-camera Light Field evaluation system. This work addresses three problems related to the process of multi-camera capture: first, whether multi-camera calibration methods can reliably estimate the true camera parameters; second, what are the consequences of synchronization errors in a multi-camera system; and third, how to ensure data consistency in a multi-camera system that records data with synchronization errors. Furthermore, this work addresses the problem of designing a flexible multi-camera system that can serve as a Light Field capture testbed.

The first problem is solved by conducting a comparative assessment of widely available multi-camera calibration methods. A special dataset is recorded, giving known constraints on camera ground-truth parameters to use as reference for calibration estimates. The second problem is addressed by introducing a depth uncertainty model that links the pinhole camera model and synchronization error to the geometric error in the 3D projections of recorded data. The third problem is solved for the color-and-depth multi-camera scenario, by using a proposed estimation of the depth camera synchronization error and correction of the recorded depth maps via tensor-based interpolation. The problem of designing a Light Field capture testbed is addressed empirically, by constructing and presenting a multi-camera system based on off-the-shelf hardware and a modular software framework.

The calibration assessment reveals that target-based and certain target-less calibration methods are relatively similar at estimating the true camera parameters. The results imply that for general-purpose multi-camera systems, target-less calibration is an acceptable choice. For high-accuracy scenarios, even commonly used target-based calibration approaches are insufficiently accurate. The proposed depth uncertainty model is used to show that converged multi-camera arrays are less sensitive to synchronization errors. The mean depth uncertainty of a camera system correlates to the rendered result in depth-based reprojection, as long as the camera calibration matrices are accurate. The proposed depthmap synchronization method is used to produce a consistent, synchronized color-and-depth dataset for unsynchronized recordings without altering the depthmap properties. Therefore, the method serves as a compatibility layer between unsynchronized multi-camera systems and applications that require synchronized color-and-depth data. Finally, the presented multi-camera system demonstrates a flexible, de-centralized framework where data processing is possible in the camera, in the cloud, and on the data consumer's side. The multi-camera system is able to act as a Light Field capture testbed and as a component in Light Field communication systems, because of the general-purpose computing and network connectivity support for each sensor, small sensor size, flexible mounts, hardware and software synchronization, and a segmented software framework.

Contents

List of Papers

This thesis is based on the following papers, herein referred to by their Roman numerals:

Paper I

Modeling Depth Uncertainty of Desynchronized Multi-Camera Systems
E. Dima, M. Sjöström, R. Olsson,
International Conference on 3D Immersion (IC3D), 2017 .... **??**

Paper II

Assessment of Multi-Camera Calibration Algorithms for Two-Dimensional Camera Arrays Relative to Ground Truth Position and Direction
E. Dima, M. Sjöström, R. Olsson,
3DTV-Conference: The True Vision - Capture, Transmission and Display of 3D Video (3DTV-Con), 2016 .... **??**

Paper III

Estimation and Post-Capture Compensation of Synchronization Error in Unsynchronized Multi-Camera Systems
E. Dima, Y. Gao, M. Sjöström, R. Olsson, R. Koch, S. Esquivel,
In manuscript, 2018 .... **??**

Paper IV

LIFE: A Flexible Testbed for Light Field Evaluation
E. Dima, M. Sjöström, R. Olsson, M. Kjellqvist, L. Litwic, Z. Zhang, L. Rasmusson, L. Flodén,
3DTV-Conference: The True Vision - Capture, Transmission and Display of 3D Video (3DTV-Con), 2018 .... **??**

Terminology

Abbreviations and Acronyms

2D	Two-Dimensional
3D	Three-Dimensional
3DV	3D Video
3DTV	3D Television
API	Application Programming Interface
AR	Augmented Reality
BRIEF	Binary Robust Independent Elementary Features
CCD	Charge-Coupled Device
CMOS	Complementary Metal Oxide Semiconductor
DIBR	Depth-Image Based Rendering
FAST	Features from Accelerated Segment Test
GDPR	General Data Protection Regulation
GPU	Graphics Processing Unit
HW	Hardware
LAN	Local Area Network
LIFE	Light Field Evaluation (system)
MSE	Mean Squared Error
MV	Multi-View
MVD	Multi-View plus Depth
ORB	Oriented FAST and Rotated BRIEF
PGPU	Programmable Graphics Processing Unit
PSNR	Peak Signal-to-Noise Ratio
RANSAC	Random Sample Consensus
RGB	Color-only (from Red-Green-Blue digital color model)
RGB-D	Color and Depth
SfM	Structure from Motion
SIFT	Scale-Invariant Feature Transform
SLAM	Simultaneous Localization and Mapping
SSIM	Structural Similarity Index

SURF	Speeded-Up Robust Features
ToF	Time-of-Flight (a depth camera technology)
VR	Virtual Reality

Mathematical Notation

The notation basis, using "a" as placeholder variable, is as follows:

a	Scalar "a"
$\vec{a}$	Vector "a"
$\overrightarrow{a}$	Ray "a"
$\mathbf{A}$	Matrix "A"
Δa	Variable related to "a".
$\max a$	Maximum of "a"
$a \neq A$	Likewise, $\vec{a} \neq \vec{A}$, $\overrightarrow{a} \neq \overrightarrow{A}$, $\mathbf{a} \neq \mathbf{A}$, $\Delta a \neq \Delta a$

The following terms are used in this work:

$\vec{c}$	Pixel coordinate point in the form $(x, y, 1)^{\mathrm{T}}$
$\vec{C}_i$	Spatial position of camera i (defined by camera's optical center point) in a 3D coordinate system
$\vec{E}$	A moving point (object) in 3D space, recorded by a camera or array of cameras
$\vec{E}_{i,n}$	3D position of the point $\vec{E}$, as recorded by camera i in its n-th frame
f_x, f_y	Focal lengths of a lens in the x and y axis scales, respectively
$\mathbf{H}$	Homography matrix in projective geometry
i, j	Indices of cameras recording a scene
$\mathbf{I}_n$	Image (frame) recorded at time t_n
k	Index with local meaning
$\mathbf{K}_i$	The intrinsic matrix of camera i
$\vec{m}$	Shortest vector connecting two rays $\overrightarrow{p}_j$, $\overrightarrow{p}_i$
n	Index with local meaning
$\overrightarrow{p}_i$	Ray with origin at optical center of camera i
$\vec{p}_i$	Vector with normalized magnitude and same direction as ray $\overrightarrow{p}_i$
$\mathbf{P}_i$	Projective matrix of camera i
$\mathbf{R}_i$	The rotation matrix of camera i
s	Skew factor between the x and y axes of a camera sensor
t	Time
$t_{i,n}$	Time (t) when camera i records the n-th image (frame)
$t_{i,n,n+1}$	Time between camera i's recordings of the n-th and $(n+1)$-th frames
$^{\mathrm{T}}$	Transpose operator
v	Speed
$\max v_{\vec{E}}$	Maximum possible speed of $\vec{E}$
$\mathbf{V}_{n,n+1}$	A tensor describing how an image recorded at t_n changes to an image recorded at t_{n+1}

$\mathbf{V}_{n,n+1}(x,y)$	A vector $[\Delta x, \Delta y, \Delta z]$ located at position (x,y) in the tensor $\mathbf{V}_{n,n+1}$
$\mathbf{V}_{n,n+1}(x,y,1)$	The first component of the vector given by $\mathbf{V}_{n,n+1}(x,y)$
x, y	Coordinates in a two-dimensional system
x_0, y_0	The x and y position of a camera's principal point on the camera sensor
X, Y, Z	Coordinates in a three-dimensional system
z	Value (magnitude) of a pixel
δ_n	Normalized time difference between two adjacent frames, n and $n+1$
Δd	Depth uncertainty (amplitude of possible distances between the camera and $\vec{E}$)
$\overline{\Delta d}$	Mean depth uncertainty
Δt	Synchronization offset (error) between cameras recording $\vec{E}$
$\Delta x, \Delta y$	Difference in x, y position of a moving pixel
Δz	Difference in z value of a moving pixel
θ	Angle between two rays recording $\vec{E}$
λ	Scale factor relating a coordinate system unit to a real-world distance unit
ν_i	Framerate of camera i

Chapter 1

Introduction

For humans, a fundamental way of understanding the world is through sight and observation; visual information is one of the main inputs for the human mind to interpret events of the real world. As human technology advances, so do the tools with which the real world is observed. Cameras, which serve as artificial counterparts to the eyes, have found application in all aspects of modern life - work, study, entertainment. In particular, systems of multiple cameras (*multi-camera systems*) have become prevalent in such wide-ranging fields as multimedia production, scientific research, surveillance, and robotics.

Multi-camera systems form a significant area for research. They have advanced rapidly, driven by improvements in digital camera technology, progress in computer vision, developments in computer engineering and Light Field theory, and the rising popularity of Virtual Reality (VR) and Three-Dimensional (3D) media entertainment [Zon12, Fit12]. This chapter explains why investigating multi-camera systems is important, introduces the purpose and scope of this investigation into multi-camera systems, and lists the goals and contributions of this work.

1.1 Motivation

1.1.1 Applications of Multi-Camera Systems

Multimedia, surveillance, machine vision, and behavioral science - there can be no doubt that all these fields have a significant impact on modern life. Visual media entertainment not only provides one of the primary ways to spend free time [SWR96, BBRP12], but also greatly affects how "alive" devices such as computers and television sets seem to the human mind [RN96]. Surveillance, for better or for worse, is fast becoming a de-facto standard in public spaces, affecting the social and criminal dynamics of modern cities [BAW13, KA14, Yes06]. Machine vision is set to permanently become a mainstay of everyday life, by virtue of self-driving cars

[LFP13, HD14] and face-recognizing smartphones [HHSP07]. Behavioral science is the study of human and animal interactions and behavior, and can explain daily human activities [MHT11, JWK09]. These fields provide examples for the application of multi-camera systems:

- In surveillance, the use of multi-camera systems provides multi-viewpoint capture to record events behind occlusions, improve observed area coverage, and increase the level of recorded detail [OLS$^+$15]. Virtual reconstruction of real environments is likewise a driving factor for using multi-camera systems in the context of surveillance [DBV16].
- In robot and machine vision, Simultaneous Localization and Mapping (SLAM) methods tend to use systems of imaging and range-finding sensors to both map the environment [HKH$^+$12] and localize the moving system's position [KDBO$^+$05, KSC15], thereby enabling autonomous movement or flight [HLP15, LFP13].
- In non-imaging research contexts, multi-camera systems are used to record human activities and movements in order to analyze social behavior [JLT$^+$15a] and improve human activity classification [OCK$^+$13]. In addition, multi-camera systems are employed to discreetly record the movement of animals in 3D space [SBND10, TFJ$^+$14].
- Last but not least, in entertainment and media production, multi-camera systems are used for purposes ranging from visual effects editing and cinematic capture [LMJH$^+$11, ZEM$^+$15] to producing 360-degree video content for VR via commercial products such as PanoCam3D [Pan17], Vuze 3D [tL17], Facebook Surround 360 [Fac17] and Google Jump [Goo17].

As these examples demonstrate, multi-camera systems find use in fields that have a significant and clear impact on society. Specific end-user applications may change; however the need for multi-camera systems themselves is unlikely to disappear in the foreseeable future, given the sheer variety of applications enabled by multi-camera systems. Moreover, multi-camera systems share a set of common properties and processes that can be investigated and improved upon. As long as investigations are focused on multi-camera systems themselves, the context and potential impact of the research remains connected to the broad range of end-user applications and through them, to society at large.

1.1.2 Light Fields and Plenoptic Capture

Research on multi-camera systems is most closely connected to Light Fields [LH96, GGSC96] and the plenoptic function [AB91], both of which present ways to model and represent the data recorded by multi-camera systems as a continuous whole. The plenoptic function is a light-ray based model that represents the full visual information that can be recorded about a 3D space. The plenoptic function describes the intensity of light rays at any 3D position, in any direction, at any time, and at any light wavelength. A single camera can record a subset of the plenoptic function - a range of wavelengths at specific time instants, in a range of directions, crossing a single position in space. Recorded wavelength range can be changed by using different filters and detector technologies. The rate of sampling time (camera framerate) can be varied depending on sensor and shutter technology. The range of observed light ray directions can be affected by the choice of lenses. However, multi-position recording is possible by increasing the number of cameras, i.e. by using a multi-camera system, or by using special optical structures implemented in plenoptic cameras [NLB$^+$05].

The Light Field [LH96] and the Lumigraph [GGSC96] are two similar parameterizations of a four-dimensional subset of the plenoptic function, encoding the set of light rays crossing a space between two Two-Dimensional (2D) planes. With growing commercial interest in 3D Television (3DTV) and VR, advances in Light Field research have led to advances in multi-camera system development for Light Field capture. Moreover, the focus on Light Fields has led to treating sets of multiple cameras as larger singular entities, namely, multi-camera systems.

1.2 Purpose Statement

Multi-camera systems are important tools in a wide range of research and engineering disciplines. However, the functionality of multi-camera systems covers more than just the in-camera data recording. There are a number of operations and processes that take place before and after the recording. These processes are related to designing and constructing multi-camera systems, ensuring that components in the system operate in collaboration with each other, and ensuring information consistency in the recorded data. Without such processes, there are merely sets of individual cameras, not dataset-producing multi-camera systems. The overall purpose of this work is to contribute to a more comprehensive understanding of how cameras can collaborate and produce consistent data, and how pre-recording and post-recording processes contribute to the design and operation of multi-camera systems used for Light Field capture.

1.3 Scope

This work is conducted within the empirical, post-positivist research paradigm, and relies on quantitative research methods. The scientific field of this work is 3D and Light Field research: an intersectional research area situated between computer engineering, computer vision, and multimedia signal processing. The surrounding context of this work is the design of a Light Field Communication System, for which this thesis considers a limited number of research problems related to 3D and Light Field acquisition. Problems related to Light Field representation, encoding, distribution and displaying are beyond the scope of this thesis. Figure 1.1 (top) illustrates the high-level structure of a 3D and Light Field communication system. The parts of

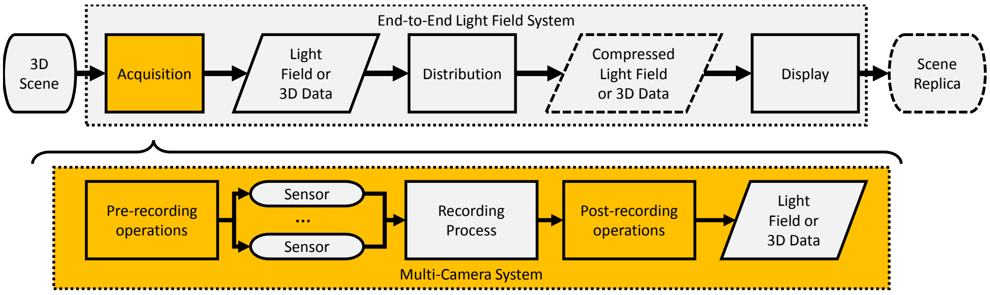


Figure 1.1: Graphical representation of end-to-end Light Field systems, where scene acquisition is performed by multi-camera systems. Color highlights show the focus of this study.

the system that are are within the scope of this work are highlighted.

This study focuses on multi-camera systems as the technology for 3D scene acquisition, specifically considering video recording with Color-only (RGB) and depth cameras. There exist alternatives for Light Field data capture, such as plenoptic cameras [NLB+05, LG09] and cameras mounted on moving gantries [VDS+15]; such alternatives are outside the scope of this thesis.

Figure 1.1 reveals how multi-camera systems fit into the context of end-to-end Light Field systems. End-to-end Light Field systems are systems that record a 3D scene and create its replica. Multi-camera systems consist of the sensors together with supporting hardware, which provide the environment for data acquisition and storage (recording). Besides the recording process, there are pre-recording and post-recording operations that enable data production with the multi-camera system.

This thesis is centered on the construction of a multi-camera system, and on specific processes within the pre-recording and post-recording blocks. Investigations into multi-camera system construction are focused on advances in the system's logical and software framework. The system hardware is limited to commonly available sensors, computers, and data transmission technologies. Investigations into the pre-recording and post-recording processes are focused on camera calibration and synchronization, due to the importance of both processes in the operation of multi-camera systems. The thesis does not seek to introduce new camera calibration methods, given the abundance of existing solutions in numerous, standardized computer vision libraries and frameworks. In this work, multi-camera calibration is addressed as a pre-recording operation, and multi-camera system synchronization is addressed through discrete pre-recording and post-recording processes.

1.4 Concrete and Verifiable Goals

In order to fulfill the purpose stated in section 1.2 and produce knowledge on multi-camera systems within the work's scope, goals are defined according within three areas of research: multi-camera system construction, multi-camera calibration, and

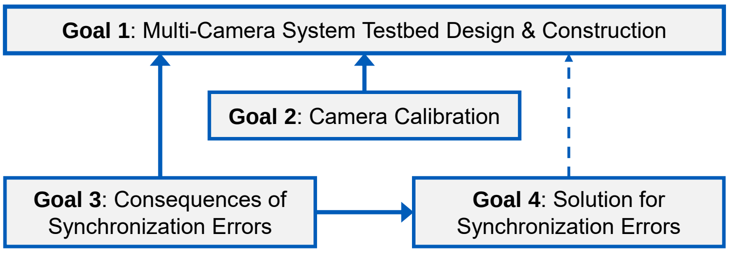

Goal 1: Multi-Camera System Testbed Design & Construction
Goal 2: Camera Calibration
Goal 3: Consequences of Synchronization Errors
Goal 4: Solution for Synchronization Errors

Figure 1.2: A graphical representation of the goals defined for this thesis. Full arrows show explicit influence between goals, and dashed arrows show indirect influence.

multi-camera synchronization. The primary goal of this work is to design and construct a Light Field Evaluation System. This system has to be a multi-camera setup that is flexible in its construction, in order to allow investigations and assessments of Light Field capture and communication. Achieving this goal fulfills the research purpose stated in section 1.2, and produces a testbed system that enables further research in Light Field capture. The primary goal is defined as follows:

- **Goal 1**: Design and construct a flexible multi-camera system testbed.

The calibration and synchronization research directions are pursued in parallel with Goal 1, and are designed to contribute to the main goal (**Goal 1)** of multi-camera system development. Figure 1.2 shows the relation between the main goal and the parallel goals (**Goal 2, Goal 3, Goal 4**). Each of the parallel goals is a separate investigation. The goals related to calibration and synchronization are defined as follows:

- **Goal 2**: Investigate the advantages and drawbacks of multi-camera calibration solutions, and assess the ability to recover the true camera parameters via calibration. This goal is addressed through the following research questions:
  - **Research question 2.1**: How good are the commonly used calibration methods at recovering the *true* camera parameters that are represented by the pinhole camera model?
  - **Research question 2.2**: Can targetless calibration methods recover the true camera parameters as effectively as target-based calibration methods?
- **Goal 3**: Investigate the consequences of inaccurate synchronization before or during recording in a multi-camera system. This goal is addressed through the following research questions:
  - **Research question 3.1**: How do errors in camera-to-camera synchronization affect the multi-camera system's ability to record scene depth?
  - **Research question 3.2**: Is the effect of synchronization errors compounded by camera positioning?

- **Goal 4**: Propose a multi-camera synchronization solution for scenarios when accurate synchronization before or during recording is not possible. This goal is addressed through the following research questions:
  - **Research question 4.1**: How accurately can the true synchronization error in a multi-camera system be estimated?
  - **Research question 4.2**: Can the re-synchronization process correct the recorded data, and thereby sufficiently approximate synchronously recorded data, by compensating the estimated synchronization error?

1.5 Outline

This thesis is structured as follows. A background to multi-camera systems is provided in Chapter 2. Investigations into selected parts of multi-camera capture - synchronization, calibration, re-synchronization - are described in Chapters 3, 4, and 5. These three chapters include the individual problem descriptions and proposed solutions that relate to the goals of this thesis and the contributions of this work. Chapter 6 details the Light Field Evaluation (LIFE) system implementation and framework. The results of the LIFE system and the three investigations are noted in Chapter 7, organized according to the respective contributions. Finally, Chapter 8 concludes the thesis, covering the outcomes, impact, and future directions of the presented work.

1.6 Contributions

The contributions on which this dissertation is based are the previously listed papers, included in full at the end of this work. As the first author of papers I, II, III and IV, I am responsible for the ideas, methods, test setup, implementations, analyses, writing, and presentation of the research work and results. For paper III, Y. Gao as the second author shared responsibility for implementation of synchronization methods, test dataset production, result analysis, and presentation of sections related to the test datasets and test setup calibration. For paper IV, M. Kjellqvist and I worked together on the software implementation. Z. Zhang and L. Litwic developed the cloud system and contributed to the communication interface definitions for the implemented system. The remaining co-authors contributed with advice and guidance throughout the research process of the respective papers. Details concerning the authors' roles and contribution are given in Chapter 7. The general purpose of each contribution is as follows:

Paper I presents a new method for modeling consequences of camera synchronization errors, and uses the new model to address general multi-camera system setup questions. **Paper II** investigates the performance of several widely available multi-camera calibration methods. **Paper III** returns to the question of camera synchronization, and presents a method for estimating and correcting the results of incorrectly synchronized multi-camera recordings. **Paper IV** introduces the high-level framework for a flexible end-to-end Light Field testbed (LIFE system), and provides the details about implementation of the LIFE system.

Chapter 2

Multi-Camera Capture

The previous chapter discussed the scope of this thesis, and mentioned how multi-camera systems are used for various applications, from surveillance and autonomous machine vision to entertainment and scientific data production. This chapter describes multi-camera systems, and the different stages of the capture process. Moreover, multi-camera systems rely on the pinhole camera model to enable geometric projection of recorded images. The pinhole camera model is therefore also described in this chapter.

2.1 Multi-Camera Systems

A multi-camera system is a collection of cameras recording the same scene from multiple viewpoints. Because the cameras are coordinated, the recorded data are consistent and the same scene is observed by all the cameras. The use and research of multi-camera systems began shortly after the introduction of consumer digital cameras in the 1990s. Two notable early multi-camera systems were the "3D Dome" [KRN97], designed to record an enclosed scene from all directions, and the "Sea of Cameras" room for virtual teleconferencing [FBA$^+$94]. These enclosed-space camera configurations were soon replaced by planar arrays of homogeneous cameras, exemplified by the Light Field video cameras of Wilburn et al. [WSLH01] and Yang et al. [YEBM02]. The change in camera layout also introduced a change in the purpose of multi-camera systems. The inward-facing multi-camera systems were designed for digitizing an enclosed scene as a 3D model, whereas the planar camera arrays were designed to record Light Fields from one general direction.

These multi-camera systems were stand-alone devices, designed to record images and video to local storage for subsequent processing and use. Another class of 3D recording systems were the *end-to-end* systems, such as [YEBM02, MP04, BK10]. These end-to-end systems combined multi-camera systems and various 3D presentation devices to show a "live" system with 3D scene input and 3D output.

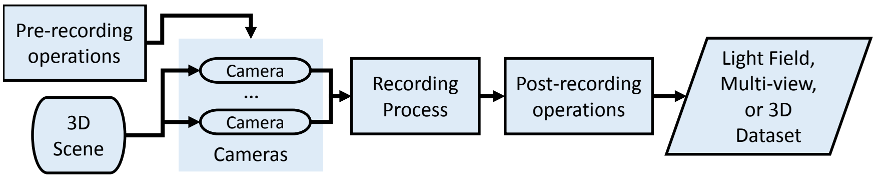


Figure 2.1: Capture process in multi-camera systems, from 3D scene to a dataset.

The next stage in the development of multi-camera systems was characterized by a greater variety in sensor types, placements, and system applications. Multi-camera systems have been created from surveillance cameras [FBLF08], 2D cameras combined with infrared-pattern and Time-of-Flight (ToF) based depth sensors [GČH12, BMNK13, MBM16], and imaging sensors mounted on mobile phones [SSS06]. The end-to-end systems were adapted for flying platforms, using lightweight, low-cost imaging sensors [HLP15]. The brief interest in 3DTV [KSM$^+$07] also fuelled the use of flat or arc-based arrays of high-quality cameras spaced at regular intervals, for multi-view video acquisition [DDM$^+$15, FBK10].

As mentioned in Section 1.1.1, multi-camera systems have applications outside of research laboratories. These systems are now embedded in smartphones [Mö18] and self-driving vehicles [HHL$^+$17], and have recently been turned into commercial products [tL17, Pan17, Inc17] and open-source design instructions [Fac17, Goo17]. This demonstrates the level of contemporary interest in multi-camera systems and the change in multi-camera system purposes. Instead of 3D object scanning and 3DTV, multi-camera systems are used in embedded applications, photography, VR, Augmented Reality (AR), 360-degree video, surveillance, and autonomous vehicles, as mentioned in section 1.1.1.

2.2 The Capture Process

The capture process is the set of operations necessary to enable the functionality of multi-camera systems. These operations can be grouped into three stages, based on multi-camera capture descriptions in [HTWM04, SAB$^+$07, NRL$^+$13, ZMDM$^+$16]. These stages are the pre-recording, recording, and post-recording stage.

Figure 2.1 shows how these three stages help convert a 3D scene into a dataset. The pre-recording stage defines how discrete cameras are combined to form a multi-camera system. A significant element of the pre-recording stage is camera calibration: a process that estimates the camera parameters using a mathematical model of the camera with ray geometry. Calibration that is more accurate implies smaller errors in the processing of data from multiple cameras, as demonstrated by Schwarz et al. [SSO14]. The recording stage is the act of capturing image sequences with the system's sensors and recording them to local camera memory. A significant part of the recording stage is camera synchronization, as indicated by Stoykova et al. [SAB$^+$07]. Synchronization during recording ensures that all cameras record images at the same

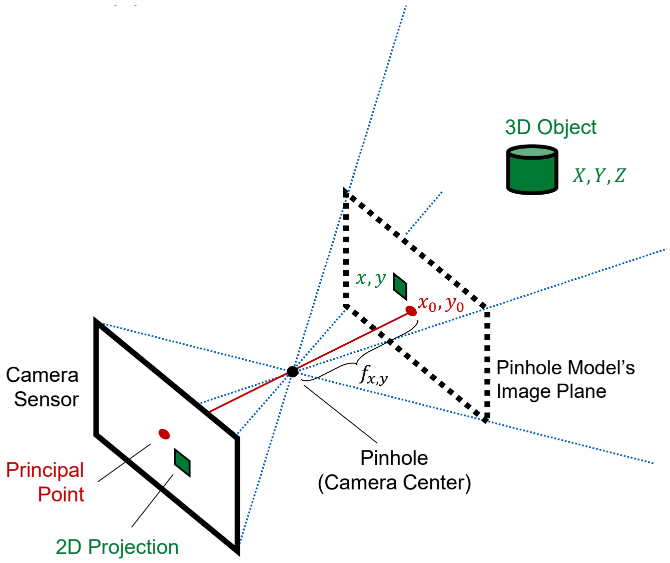


Figure 2.2: Pinhole camera model: projection from 3D scene to 2D image.

time, thereby capturing the same 3D scene. Finally, the post-recording stage consists of activities that convert the recorded sequences into datasets. A dataset is the consistent information from all cameras that can be jointly used by applications no longer part of the multi-camera system. The 3D information in the dataset can be encoded as a Light Field, as multiview sequences, as Multi-View plus Depth (MVD), or as some other format. The conversion from raw camera sequences to the selected dataset format is one example of an operation in the post-recording stage.

2.3 Pinhole Camera Model

When recording scenes from different viewpoints with multiple cameras, there is a need to map the 2D image from the camera sensor onto the 3D scene. In the context of 3D recording, this is achieved by using the mathematical framework of projective geometry [HZ03]. The projective geometry framework defines a mathematical camera model called the *pinhole camera model*. The pinhole camera models is so called because instead of describing the camera aperture or lens system, it assumes that each point on the camera sensor is projected into the world in a straight line crossing the camera optical center, as seen in Figure 2.2. The pinhole camera model describes cameras by two matrices: the intrinsic matrix and the extrinsic matrix.

The intrinsic matrix $\mathbf{K}$ describes the internal parameters of one camera. The internal parameters are the focal lengths f_x, f_y, principal point offsets x_0, y_0, and the skew factor s between the sensor's horizontal and vertical axes. The focal lengths f_x, f_y are scaled to the camera's pixel width and height, respectively, from the camera focal length f. These parameters form the intrinsic matrix:

$$\mathbf{K} = \begin{bmatrix} f_x & s & x_0 \\ 0 & f_y & y_0 \\ 0 & 0 & 1 \end{bmatrix} . \tag{2.1}$$

The principal point offset describes where the camera sensor is intersected by the optical axis: a line perpendicular to the sensor and passing through the pinhole. The focal length denotes the distance between the sensor and the optical center (pinhole) of the camera. The Gaussian lens model [Hec87] uses focal length to describe the magnification power of a lens, by matching the image size rendered by the lens with the image size produced by a pinhole camera with the given focal length. The pinhole camera model does not incorporate the Gaussian lens model.

The extrinsic matrix describes the 3D position and orientation of one camera. In multi-camera systems, the camera extrinsic matrices are defined in a common coordinate system. The common coordinate system may be aligned to the world coordinate system, or one of the cameras is used as the coordinate system origin and orientation reference. The camera position is encoded as the 3D point $\vec{C}$, and camera rotation is recorded in the rotation matrix $\mathbf{R}$. The extrinsic matrix is commonly denoted by the combination of the camera rotation and translation:

$$[\mathbf{R}| - \mathbf{R}\vec{C}] . \tag{2.2}$$

Together with $\mathbf{K}$, the extrinsic matrix $[\mathbf{R}| - \mathbf{R}\vec{C}]$ allows for the creation of the 4-by-3 camera matrix $\mathbf{P}$:

$$\mathbf{P} = \mathbf{K}[\mathbf{R}| - \mathbf{R}\vec{C}] . \tag{2.3}$$

The camera matrix is the projective geometry basis for projecting a 3D point with coordinates X, Y, Z to the 2D camera sensor plane at coordinates x, y:

$$\lambda \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = [\mathbf{K}|0_3] \begin{bmatrix} \mathbf{R} & -\mathbf{R}\vec{C} \\ 0_3^\mathrm{T} & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} . \tag{2.4}$$

Chapter 3

Synchronization and Depth Uncertainty Modeling

Section 2.1 mentioned that multi-camera systems are used to record consistent data from multiple perspectives. The consistency of recorded data is influenced by how well the cameras are synchronized. Perfect synchronization in a multi-camera system occurs when *all cameras take a single sample of the scene at the same time*. Perfect synchronization is not a guaranteed property of a multi-camera system due to technical or cost-based limitations of the system's components. The lack of perfect synchronization causes inconsistent sampling of a scene that changes over time. Therefore, synchronization errors affect the consistency of data recorded by a multi-camera system. Since synchronization error is an independent factor in a multi-camera system, it must be possible to model the influence of synchronization on the capabilities of a multi-camera system. This chapter describes how synchronization errors affect camera systems and geometry estimation (Section 3.1), and how this influence is expressed in a parametric model (Section 3.2).

3.1 Synchronization and the Reason for Depth Uncertainty

Synchronization between cameras can be achieved by supporting external synchronization signaling in the camera hardware, or by signaling through software instructions via the camera Application Programming Interface (API) [LZT06]. In both cases, perfect synchronization cannot be guaranteed unless the signaling bypasses all on-camera processing and directly triggers the camera shutter. Hardware support for an external control signal allows for more accurate synchronization than any other method [LHVS14], but tends to increase the unit cost of the sensors and therefore the total cost of the camera system [PM10]. Moreover, restricting a camera system to hardware-synchronized sensors can result in a lower scene sampling

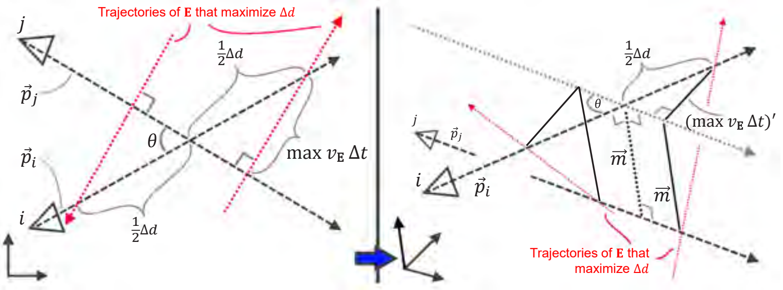

Trajectories of $\mathbf{E}$ that maximize Δd
j
$\vec{p}_j$
$\vec{p}_i$
i
$\frac{1}{2}\Delta d$
θ
$\max v_{\mathbf{E}}\,\Delta t$
$\frac{1}{2}\Delta d$
j
$\vec{p}_j$
i
$\vec{p}_i$
$\frac{1}{2}\Delta d$
θ
$(\max v_{\mathbf{E}}\,\Delta t)'$
$\vec{m}$
$\vec{m}$
Trajectories of $\mathbf{E}$ that maximize Δd

Figure 3.1: Geometric basis for deriving depth uncertainty Δd.

rate [ESH+12] or prevent the use of entire categories of cameras, such as affordable ToF depth cameras that allow capture control only through the camera API [SLK15]. Thus, any decision about the required accuracy of synchronization in a multi-camera system affects the system's design and cost. These in turn affect the system's suitability for a given application scenario.

Scenarios like motion capture [BRS+11], cinematic effect production [ZEM+15] and human activity recognition [JLT+15b] (see Section 1.1) have an implicit aim of using the scene geometry. If the scene contains moving elements, multi-camera systems with imperfect synchronization will induce errors in the geometric reconstruction of the moving elements. This occurs because the geometry recorded by the sensors is not recorded at the same time instant. The permissible range of geometry reconstruction error varies depending on the use case - for example, the pose-prediction based system in [JLT+15b] is less sensitive to geometric noise than the depth-based per-pixel cinematic lighting effects of [ZEM+15]. These errors are present in camera setups with global sensor shutters. Rolling shutters are likely to increase the error even further, since rolling shutter systems require synchronization between scanlines rather than sensors.

The specific use-cases impose requirements on maximum permitted geometric error, which in turn sets the level of the required synchronization accuracy. This influences the system design and cost. This relation between synchronization accuracy and geometric error must be modeled, in order to predict the extent of geometry errors arising from synchronization errors. To keep the model in context of multi-camera systems, the geometric error can be described via *depth uncertainty*.

3.2 Definition of Depth Uncertainty

In a multi-camera system, the 3D position of a scene point is determined by triangulation: pinpointing how far along a camera ray the scene point is located. Without perfect synchronization, triangulation produces an incorrect position; the unknown true position may lie elsewhere on the camera ray, at a different depth. Depth uncertainty is the error between the nearest and farthest possible true positions, a measure of how large the interval is in which we are *certain* that the scene point must be.

Figure 3.1 shows the principle for deriving depth uncertainty. Let i and j be two cameras that sample a scene, in which a moving element $\vec{E}$ exists. Each camera's data only states that, at the moment when i, j sample the scene, $\vec{E}$ must lie somewhere along the respective rays $\overrightarrow{p}_i$, $\overrightarrow{p}_j$. If i and j are perfectly synchronized, the 3D position $\vec{E}$ must be at the intersection of rays $\overrightarrow{p}_i$ and $\overrightarrow{p}_j$. If the synchronization is not perfect, then $\vec{E}$ has enough time (t) to move from a position on $\overrightarrow{p}_j$ to a position on $\overrightarrow{p}_i$, with neither position being the intersection of $\overrightarrow{p}_i$ and $\overrightarrow{p}_j$. The difference between the true position of $\vec{E}$ and the estimated position (intersection of $\overrightarrow{p}_i$ and $\overrightarrow{p}_j$) is the geometric error induced by the synchronization error Δt. At this point, Δt is the time between shutter activation on camera i and camera j.

While a single "true" position of $\vec{E}$ cannot be known, as long as $\vec{E}$ has a maximum speed $\max v_{\vec{E}}$, there exists a limit to how far $\vec{E}$'s true position on $\overrightarrow{p}_i$ can be from the intersection. In other words, the position of $\vec{E}$ is fixed in two lateral dimensions by the ray $\overrightarrow{p}_i$ and can vary between a minimum and maximum distance from i. The difference between these distances is the depth uncertainty Δd.

If the rays $\overrightarrow{p}_i$ and $\overrightarrow{p}_j$ are not co-planar, Δd can be found by assuming two linear trajectories of distance $\max v_{\vec{E}} \Delta t$ that maximize Δd, as shown in Fig. 3.1 (right), and calculating:

$$\Delta d = \frac{2\sqrt{\left(\max v_{\vec{E}} \Delta t\right)^2 - \|\vec{m}\|^2}}{\sin(\theta)} \, , \tag{3.1}$$

where θ is the angle between $\overrightarrow{p}_i$ and $\overrightarrow{p}_j$, given by:

$$\theta = \arccos\left(\frac{\vec{p}_i \cdot \vec{p}_j}{\|\vec{p}_i\| \, \|\vec{p}_j\|}\right) \, , \tag{3.2}$$

and $\|\vec{m}\|$ is the nearest distance between $\overrightarrow{p}_i$ and $\overrightarrow{p}_j$. The vectors $\vec{p}_i, \vec{p}_j$ denote the directions of the respective rays.

Equation (3.1) describes a discrete case involving only two rays with one possible intersection. We call the combination of rays $\overrightarrow{p_i}, \overrightarrow{p_j}$ "valid", if the rays get close enough to each other and equation (3.1) produces a real, non-negative Δd. Depth uncertainty can be used as a general property of a multi-camera system, by assessing all possible combinations of rays, for which one ray belongs to one camera and another ray to another camera. We define the general depth uncertainty $\overline{\Delta d}_{i,j}$ for cameras i, j as the mean of all valid n combinations of rays $\overrightarrow{p_i}, \overrightarrow{p_j}$ in:

$$\overline{\Delta d}_{i,j} = \frac{1}{n}\sum_{k=1}^{n} \Delta d_k \text{ , where } \Delta d_k \in \{\Delta d \mid \forall \, (\overrightarrow{p_i}, \overrightarrow{p_j} \implies \Delta d) \} \, . \tag{3.3}$$

To make the model in Equation (3.3) practical, the camera and ray definitions are expressed via a standard way of modelling cameras: the pinhole camera model [HZ03] described in Section 2.3. In the pinhole camera model, a 3-by-3 matrix $\mathbf{K}$ represents the camera sensor and lens properties, a 3-by-3 matrix $\mathbf{R}$ represents the camera rotation, and the 3D point $\vec{C}$ represents the camera position. If a ray $\overrightarrow{p}_n$ starts at the center of camera n and intersects the camera sensor at pixel coordinate $\vec{c}_n = (x, y, 1)^{\mathrm{T}}$, then $\overrightarrow{p}_n$ can be described by:

$$\overrightarrow{p}_n = \vec{C}_n + \lambda \mathbf{R}_n^{-1} \mathbf{K}_n^{-1} \vec{c}_n \ , \tag{3.4}$$

where λ is a positive, real, arbitrary scale factor. Equation (3.3) is defined for a camera pair. In a multi-camera context with n' cameras, Equation (3.3) is applied to all pairwise camera combinations, and the best pairwise result determines the system's overall depth uncertainty:

$$\overline{\Delta d} = \min_{i,j}(\overline{\Delta d}_{i,j}), \text{ where } i, j \in \{1, 2, \ldots, n'\}. \tag{3.5}$$

Thus, Equation (3.3) models the connection between a multi-camera system's synchronization accuracy and resulting geometric errors, without foreknowledge of object motion and position probabilities. The depth uncertainty model relies on a common camera model and a context-derived scene value (the maximum speed of objects in a scene). The depth uncertainty model is defined for the pinhole camera, which in synchronization terms is equivalent to a global shutter camera.

Chapter 4

Multi-Camera Calibration

Section 2.1 described multi-camera systems, and Section 2.3 explained the pinhole camera model. In addition, Section 2.2 also described the capture process and how calibration is a significant element of the pre-recording stage in multi-camera systems. This chapter covers the definition of geometric camera calibration, describes the differences between target-based and targetless geometric calibration, and discusses calibration quality.

4.1 Geometric Camera Calibration

Geometric camera calibration is a process that estimates camera positions, view directions, and lens and sensor properties [KHB08]. In multi-camera systems, calibration also ensures that the camera positions and orientations are described in the same coordinate system. The output of calibration is a set of parameters, defined by the pinhole camera model and a lens distortion model. These parameters are required for any geometric operations involving the data produced by the camera system, because they define how color and intensity values project from the 2D camera sensor into the 3D scene space. As a result, errors in these parameters have a direct effect on how well the recorded data from multiple cameras can be fused in a consistent way [SSO14].

In the context of the pinhole camera model (Section 2.3), camera calibration is separated into two discrete stages: *intrinsic* and *extrinsic* calibration. These stages are related to the intrinsic and extrinsic matrices, respectively. *Intrinsic* calibration is a process that estimates parameters describing the camera sensor and the basic optical system. In addition to the intrinsic matrix $\mathbf{K}$, intrinsic calibration methods also estimate lens distortion parameters, to better relate actual cameras to the pinhole camera model. The common calibration methods [Zha00, Bou16] and routines in computer vision libraries such as OpenCV [Bra00, Gab17] estimate radial and tangential distortion parameters of the Brown-Conrady distortion model [Bro66]. *Ex-*

trinsic calibration is a process that estimates parameters describing relative positions of the cameras. One camera is commonly selected as the coordinate system origin, although there also exist methods that place the coordinate system origin at the center of the correspondence points found during the calibration process [SMP05]. While these stages are usually distinguished from each other, in the case of multi-camera systems, intrinsic and extrinsic calibration is commonly conducted in a joint calibration process.

4.2 Target-based and Targetless Calibration

The process of calibration has been implemented by a number of methods that use the pinhole camera model. Despite differences in realization, the calibration methods tend to follow the same three-step high-level template. (1) Corresponding scene points are located in the camera images. These correspondence points are locations in the scene that can be uniquely identified in camera images, regardless of where in image the point is seen. (2) Correspondence point coordinates are used together with projective geometry to construct a system of equations. Within this system, camera parameters are the unknown variables. (3) The equation system is solved by combining an analytical solution and a maximum-likelihood-based optimization of camera parameter estimates.

A significant difference between various calibration methods lies in the first step: selection of corresponding scene points. Based on this selection, calibration methods are classified as *target-based* or *targetless* calibration. The high-level advantage of target-based methods is that the corresponding scene points provide not only relative camera parameter constraints, but also information about the world's coordinate system scale and orientation. Targetless calibration methods, on the other hand, are easier to automate, do not require a specially constructed object in the scene, and can therefore be applied to a wider variety of scenes.

4.2.1 Target-based Calibration

Target-based calibration methods assume that the scene contains an object with known dimensions and a shape or texture that highlights specific points on the object. Such an object is called a "calibration target", and is often artificially introduced into the scene. The key property of a calibration target is that it imposes additional constraints on the in-scene layout and distribution of correspondence points. Figure 4.1 presents an example of an artificially placed calibration target in a scene.

The most influential and most cited target-based calibration method is [Zha00]. This method defines the calibration target as a black-and-white checkerboard printed on a flat 2D surface. The corners of the checkerboard squares are the correspondence points. By reformulating the pinhole camera equation for 3D to 2D projection (Equa-

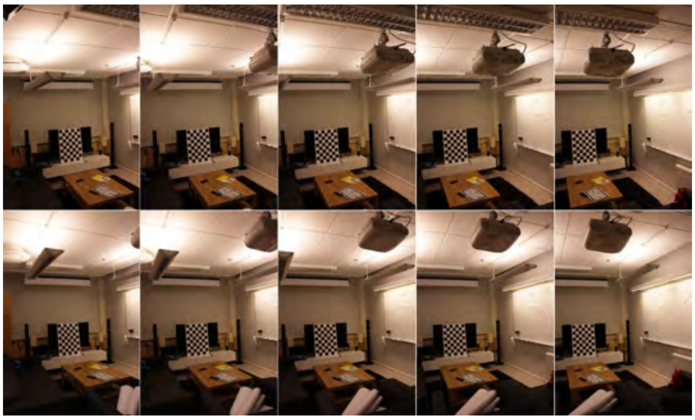


Figure 4.1: An example of a calibration scene from multiple camera views. The scene contains a calibration target (checkerboard) for target-based methods, and a sufficient number of edges and textures for targetless calibration.

tion (2.4), this calibration method establishes a homography $\mathbf{H}$:

$$\lambda \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \mathbf{K}[R] \begin{bmatrix} \mathbf{R} & -\mathbf{R}\vec{C} \end{bmatrix} \begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix} = \mathbf{H} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} . \tag{4.1}$$

The homography is based on the camera intrinsic and extrinsic matrices, and defines how a 2D surface (such as the checkerboard) is projected onto the camera's 2D image plane. Equation (4.1) allows to establish a closed-form solution. With three or more checkerboard observations at different positions, the closed-form solution has a single unique solution, up to a scale factor. The known distances between checkerboard squares are used to resolve the unknown scale factor. Once the intrinsic and extrinsic parameters are estimated, they are refined together with lens distortion parameters. This refinement is done by minimizing the distance between all checkerboard points in recorded images and their projected locations based on the estimated parameters. The lens distortion is modeled using the first few parameters in the Brown-Conrady [Bro66] distortion model.

The calibration method by Zhang et al. [Zha00] has been adapted into reusable toolboxes [Bou16] and incorporated in widely-used image processing tools such as Matlab [Mat17] and OpenCV [Bra00, Gab17]. A subset of target-based calibration methods adapts Zhang et al.'s method by specifying a different target, such as a unique, pre-generated noise pattern [LHKP13], regular patterns [LS12], 3D corners [GMCS12], and spheres [RK12]. The change of calibration target allows for better identification of the correspondence points, or provides more constraints on camera parameters by adding more relations between the correspondence points and their possible projections.

4.2.2 Targetless Calibration

Targetless calibration methods, also known as self-calibration methods, use the three-step approach described in Section 4.2. There are two significant differences between targetless and target-based calibration methods. First, targetless calibration methods use random, uniquely identifiable scene features as correspondence points. Second, due to the absence of a known reference for distances, the targetless calibration methods can estimate camera parameters up to a scale factor. If necessary, the scale factor is resolved based on an additional constraint on the camera parameters provided by the context in which the method is applied.

In targetless calibration methods, the correspondence points in camera images are generated by detecting locations in the scene that generate local maxima responses from a target-detection algorithm. A number of targetless calibration methods [BEMN09, SSS06, GML^{+}14, DEGH12] use generic feature detection and description algorithms such as the Scale-Invariant Feature Transform (SIFT) [Low99], Speeded-Up Robust Features (SURF) [BETVG08], Oriented FAST and Rotated BRIEF (ORB) [RRKB11] algorithms. In part, the use of such generic feature-detection algorithms is motivated by scenarios where pre-seeding the scene with artificial features is impossible. Alternatively, self-calibration methods such as [SMP05] pre-seed the scene with artificial features, such as a small, manually-moved light source. This allows for the use of a custom-made feature detection algorithm, thereby reducing the likelihood of incorrect correspondence identification. However, such features do not constitute a calibration target, because the relative positions of all such artificial points are not known. This means that such feature points do not provide a real-world scale, nor a reference to a "correct" correspondence point structure. Therefore, in targetless calibration methods, the closed-form analytical solution is constructed based on the rigidity of the correspondence points, when observed from several viewpoints. In targetless calibration methods, Random Sample Consensus (RANSAC) is usually incorporated in the parameter optimization step, in order to reject incorrectly detected correspondence points that act as outliers during reprojection.

4.3 Calibration Quality and Reprojection Error

Calibration methods have to self-assess the quality of their camera parameter estimations, because these methods often rely on likelihood-based optimization. This optimization is necessary because input errors are bound to be present in the data on which cameras are calibrated, i.e. camera images. One source of input errors is the incorrect detection and matching of correspondence points. Another input error source is the non-linear distortion caused by the camera lens system. Calibration methods commonly use the Brown-Conrady lens model [Bro66], which does not represent such lens properties as defocus, chromatic aberration [ESGMRA11], coma, field curvature, astigmatism [Mac06], flare, glare and ghosting [TAHL07, RV14]. Moreover, the architecture of digital sensors leads to noise in the camera [HK94, SKKS14] which affects the scene sampling and therefore the accuracy of feature detection in the scene. The common sensor types in visible-spectrum cameras are Charge-Coupled Device (CCD) and Complementary Metal Oxide Semiconductor (CMOS) sensors. CMOS and CCD sensors suffer from temporally fluctuating noise and fixed-pattern noise [BCFS06, HK94]. Examples of the temporal noise sources in CMOS and CCD sensors are the quantum uncertainty of light, the free electron generation from thermal energy in silicon, the gain and analog-to-digital conversion during sensor readout. CCD sensors also suffer from charge overflow between nearby pixels [BCFS06], and CMOS sensors are affected by thermal MOS device noise [HK94]. In addition to temporal noise, CCD and CMOS sensors are affected by fixed pattern noise, which is a fixed variation of output between pixels, given the same input, and is caused by variations of each pixel's quantum efficiency [HK94].

The existing calibration methods estimate camera parameters up to a certain threshold of accuracy, since input errors are unavoidable in the current calibration processes. Since constraints on camera parameters are given by equation systems based on projective geometry, the corresponding quality assessment is also usually based on projective geometry. The accuracy of calibration is often measured by the correspondence point reprojection error: the difference in positions between where a correspondence point is observed in one image, and where the same point is projected into the image from another camera's observation.

As Schwarz et al. demonstrate in [SSO14], processes that depend on calibration data, such as reprojection of image points, are highly sensitive to errors in both intrinsic and extrinsic camera parameters. Thus, it is important to know how accurately these parameters can be identified using different methods. Using the correspondence point reprojection error as the quality metric for camera calibration is a fundamentally problematic, because this metric is not directly based on the real-world parameters that the calibration process is supposed to recover. Correspondence points are affected by input errors at the capture stage, therefore the evaluation metric is also affected by these errors. Multi-camera calibration methods rely on the pinhole camera model and Brown-Conrady distortion model, and thus do not model all input error sources in the camera system. While commonly used calibration methods compute the calibration accuracy from the reprojection error, the accuracy of these methods remains unknown with respect to the ground truth - the physical properties of the camera system.

Chapter 5

Re-Synchronization of Recorded Data

Chapters 3 and 4 described camera calibration and theoretical modeling of the consequences of synchronization. Both these topics relate to the pre-recording stage of the capture process (see Figure 1.1 in Chapter 1) and not to direct manipulation of the recorded data. Non-synchronized multi-camera systems exist because of technology, cost, or other limitations, as demonstrated by [TTN08, YEBM02, YTJ$^+$14, HBNF15]. In particular, the use of any camera that cannot be synchronized, such as the low-cost Kinect ToF camera, leads to non-synchronized multi-camera systems. Since such non-synchronized systems exist, there is an implicit need to address synchronization errors. These errors can be addressed in the post-recording stage of the capture process (see Section 2.2), by applying an error-compensation solution on the recorded data, rather than affecting the multi-camera system design.

5.1 Synchronization and Camera Models

Synchronization of cameras is commonly considered as a separate aspect of multi-camera systems, not as in integral part of the multi-camera system model. It is not a parameter in the pinhole camera model (see Section 2.3) nor in dedicated multi-camera models such as [GNN15, LLZC14, SSL13, SFHT16, LSFW14, WWDG13, Ple03]. Surveys on multi-camera system pipelines tend to avoid explicit discussion of synchronization [NRL$^+$13, ZMDM$^+$16]. Moreover, standard applications using multi-camera data assume that the data are synchronous (i.e. have been recorded by synchronous cameras). For example, the expectation of perfect synchronicity is evident in the treatment and formulation of multi-view geometry [HZ03]: no parameter exists to describe the temporal difference between the involved cameras. Likewise, the fundamental methods of Depth-Image Based Rendering (DIBR) [Feh04] do not parametrize the difference of capture times between camera images and depthmaps.

The requirement for synchronized data is a property assumed to be true by default, unless explicitly stated otherwise. Therefore, non-synchronized camera systems require post-recording synchronization in order to satisfy this default assumption of synchronicity in datasets.

5.2 Post-Recording Synchronization

Research on post-recording synchronization can be split into two categories - *video sequence alignment* and *implicit synchronization*. Video sequence alignment is commonly achieved by determining a temporal offset between two frames with the same identifier in the sequences. In such cases, the temporal offset is equal to the synchronization error. Implicit synchronization is achieved by modifying data consumer applications to explicitly compensate for synchronization errors.

Sequence alignment methods use various cues to establish a temporal correspondence between video sequences. A few of these methods rely on meta-information such as audio tracks [SBW07], encoded bit-rate patterns [SSE$^+$13], or environmental in-scene signals [SWBS06]. These methods assume the same frame speed between sequences. Most methods, however, operate on arbitrary scene content by minimizing differences in either image intensity [DPSL11, CI02] or feature point trajectories [LY06, TVG04, LM13, EB13, PM10, DZL06, PCSK10]. The common factor among all forms of video sequence alignment is that the final output is the synchronization offset parameter, not a synchronized dataset. As a result, the problem of using an estimated offset is deferred to an as-yet undefined later-stage process.

Implicit synchronization has been treated in the literature as a side component of solutions to other research problems related to multi-camera systems. Works such as [KSC15, RKLM12, AKF$^+$17a, NK07, NS09] all define late-stage applications (solutions) that explicitly address the lack of synchronization in input data. In particular, [RKLM12] avoids synchronization by treating non-synchronized depthmap sequences as a low-resolution guide to reduce search space for image-to-image correspondence mapping. High-resolution refinement and new image rendering is conducted only from the synchronous image data. While [RKLM12] defines a rendering process, [KSC15] uses two non-synchronized cameras to reconstruct the system's movement trajectory through a static environment. Synchronization errors are compensated for during the sensor-to-sensor reprojection process, wherein the origin of one sensor is displaced from the origin of the other sensor along the estimated trajectory. Finally, [AKF$^+$17a, NK07, NS09] describe methods to estimate the camera geometry (the extrinsic parameters mentioned in Section 4.1). Non-synchronicity between the cameras is treated by using feature point trajectories instead of discrete positions as the basis for the polynomial reprojection equation system. In addition to estimating extrinsic parameters, [AKF$^+$17a, AKF$^+$17b] can also be used as a video sequence alignment method to only output the synchronization offset.

Implicit synchronization methods are application-specific, and cannot easily be transferred from the context of one problem to another. However, these methods do address the effects of synchronization error within the context of their applications.

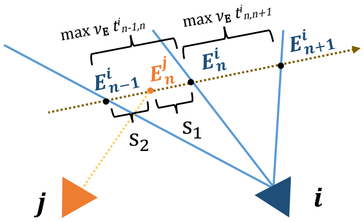

$\max v_{\mathbf{E}}\, t^{i}_{n-1,n}$
$\max v_{\mathbf{E}}\, t^{i}_{n,n+1}$
E^{i}_{n+1}
E^{i}_{n}
E^{i}_{n-1}
E^{j}_{n}
s_1
s_2
j
i

Figure 5.1: A moving point $\vec{E}$ observed by two non-synchronized cameras.

On the other hand, video sequence alignment methods stop short of applying a corrective change on the recorded data. While these methods are used to estimate the synchronization error, they do not manage or compensate for its effect. A process that *does* apply a correction on non-synchronized datasets would serve as a bridge between non-synchronized multi-camera systems and applications that are based on the default assumption of synchronicity.

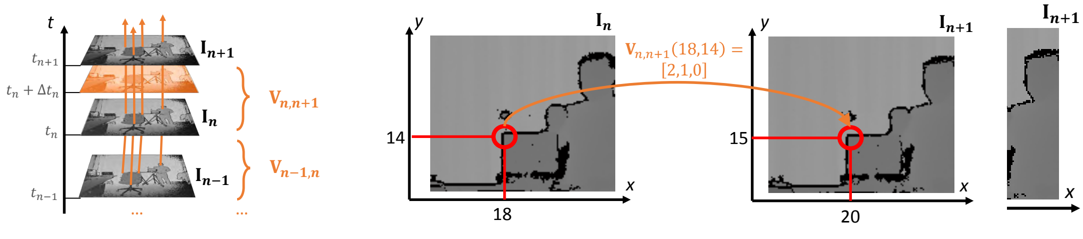
5.3 Re-Synchronization

Synchronization errors matter only if there is movement in the recorded scene, as described in Section 3.1. Therefore, re-synchronization is defined in the context of at least two non-synchronized cameras recording the same moving object. Re-synchronization is a two-part process: (1) estimation of the synchronization error, and (2) compensation of the synchronization error. The first part is equivalent to video sequence alignment from Section 5.2, and the second part addresses the gap between video sequence alignment and implicit synchronization. Estimation of the synchronization error is mentioned here because the compensation step requires this error to be a known quantity.

5.3.1 Synchronization Error Estimation

In order to compensate for the synchronization error Δt, the error must first be a known quantity. This section demonstrates how to determine Δt using an alternative to video sequence alignment methods. The same assumption is used as in [TVG04, PM10, AKF^{+}17b]: under small timescales, objects in real scenes have an approximately linear movement along a constant direction. This assumption allows for the modeling of Δt from the observed difference in position of the moving object $\vec{E}$, as illustrated in Figure 5.1.

If $\vec{E}_{i,n}$ is the 3D position of the object $\vec{E}$ recorded in the n-th frame of camera i, then given two sensors i and j and a three-frame recording window, the synchronization error Δt_n between the n-th frame of camera i and n-th frame of camera j



(b)

Figure 5.2: a) Correcting synchronization error by tensor-based interpolation. b) Constructing the tensor between adjacent frames.

can be found as follows:

$$\frac{s_1}{s_1 + s_2} = \frac{\max v_{\vec{E}} \Delta t_n}{\max v_{\vec{E}} \|t_{i,n} - t_{i,n-1}\|} = \frac{\Delta t_n}{1/\nu_i} \implies \Delta t_n = \frac{s_1}{(s_1 + s_2)\nu_i} \ , \tag{5.1}$$

$$\text{... where } s_1 = \|\vec{E}_{j,n} - \vec{E}_{i,n}\| \ , \ s_2 = \|\vec{E}_{j,n} - \vec{E}_{i,n-1}\| \ .$$

In this equation, ν_i is the recording framerate of camera i, and Δt_n^i indicates the time when the n-th frame was recorded in camera i. Equation (5.1) works as long as the motion of the object is not directly toward or away from a camera. In other words, *both* cameras need to observe that the object is moving. The 3D positions of $\vec{E}$ can be determined by finding such trajectory that satisfies the following constraints:

$$\frac{\|\vec{E}_{i,n-1} - \vec{E}_{i,n}\|}{\|\vec{E}_{i,n} - \vec{E}_{i,n+1}\|} = \frac{\|t_{i,n-1} - t_{i,n}\|}{\|t_{i,n} - t_{i,n+1}\|} \text{ and } \frac{\|\vec{E}_{j,n-1} - \vec{E}_{j,n}\|}{\|\vec{E}_{i,n-1} - \vec{E}_{i,n}\|} = \frac{\|t_{j,n-1} - t_{j,n}\|}{\|t_{i,n-1} - t_{i,n}\|} . \tag{5.2}$$

The constraints given in Equation (5.2) are related to the projections of the object in the recorded frames, described by the 3D to 2D pinhole camera projection Equation (2.4) in Section 2.3.

5.3.2 Synchronization Error Compensation

Figure 5.2 demonstrates the compensation principle applicable to a depthmap sequence. The positions of all objects need to be compensated, which can be achieved by creating an image synchronized at time $t_n + \Delta t_n$ from the recorded images $\mathbf{I}_n, \mathbf{I}_{n+1}$. The time $t_n + \Delta t_n$ is when a corresponding n-th reference frame was recorded in another camera, to which this sequence is being synchronized. From this point onward, the synchronization error Δt_n is assumed to be known.

The synchronization error Δt_n is expressed as a time-based ratio δ_n. The changes between the recorded images $\mathbf{I}_n$ and $\mathbf{I}_{n+1}$ are encoded at a pixel-by-pixel level in the tensor $\mathbf{V}_{n,n+1}$:

$$\delta_n = \frac{\Delta t_n}{\Delta t_n + \frac{1}{\nu_i} - \Delta t_{n+1}} \ ; \mathbf{V}_{n,n+1}(x,y) = [\Delta x, \Delta y, \Delta z] \ . \tag{5.3}$$

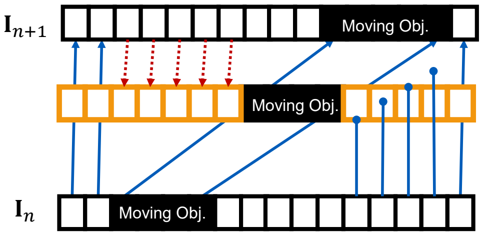


Figure 5.3: A 2D view of tensor interpolation between two frames with a static background and a moving foreground segment. Solid lines display how scene points are mapped from image to image by the tensor. Dashed lines indicate mappings that do not exist in the tensor.

The tensor $\mathbf{V}_{n,n+1}$ describes how the position (x, y) and value (z) changes for every single scene point from image $\mathbf{I}_n$ to $\mathbf{I}_{n+1}$. These changes are represented by $\Delta x, \Delta y, \Delta z$, respectively. The tensor $\mathbf{V}_{n,n+1}$ can also be considered as a matrix of size x by y by 3. Each frame in the non-synchronized sequence is warped according to Equation (5.4):

$$\mathbf{I}_{t_n+\Delta t_n}(x,y) = \begin{cases} \mathbf{I}_n(x',y') + \delta_n \mathbf{V}_{n,n+1}(x',y',3), & \text{if } \triangle; \\ \mathbf{I}_{n+1}(x,y), & \text{otherwise.} \end{cases}$$
$$\triangle\text{: } \exists \begin{pmatrix} x' \\ y' \end{pmatrix} \text{ s.t. } \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} x' + \delta_n \mathbf{V}_{n,n+1}(x',y',1) \\ y' + \delta_n \mathbf{V}_{n,n+1}(x',y',2) \end{pmatrix} \quad (5.4)$$

The conditional nature of Equation (5.4) is due to the background overlap caused by moving foreground segments. Figure 5.3 illustrates a one-line example of an interpolation with a static background and a moving foreground segment, with blue lines demonstrating how points are mapped from image $\mathbf{I}_n$ to $\mathbf{I}_{n+1}$ in the tensor $\mathbf{V}_{n,n+1}$. In the interpolated image, there are pixels which do not have an associated mapping from $\mathbf{I}_n$. Since these areas correspond to a revealed background, the value for these pixels must be taken directly from $\mathbf{I}_{n+1}$.

The tensor $\mathbf{V}_{n,n+1}$ is equivalent to a dense optical flow map between n and $n+1$ frames. An optical flow map for feature-rich images can be created by using optical flow estimation algorithms such as [Far03]. However, these algorithms become unreliable in images with few features and textures, such as depthmaps. An approach for estimating $\mathbf{V}_{n,n+1}$ in depthmaps is outlined in Figure 5.4. This approach consists of (1) depthmap segmentation, (2) coarse segment motion mapping, (3) dense pixel motion mapping, and (4) the depthmap correction through interpolation.

1: Segmentation is begun by detecting object edges as depth discontinuities in both $\mathbf{I}_n$ and $\mathbf{I}_{n+1}$, using the difference map $\mathbf{I}_{n+1} - \mathbf{I}_n$. The pixels at the edge are used as seed points for iteratively adding neighbouring pixels that belong to the same object surface as these seed points.

2: After the depthmap is segmented, the segments from $\mathbf{I}_n$ are matched with segments in $\mathbf{I}_{n+1}$ based on overlap, position, and similarity in size. Coarse motion is estimated as the difference between center positions of the matched segments, and

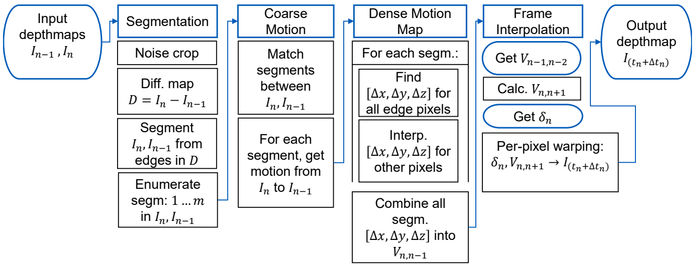


Figure 5.4: A process to correct synchronization errors.

does not describe how each pixel of the segment has moved from frame n to frame $n + 1$.

3: Dense motion is the individual motion of pixels in each segment, between frames n and $n + 1$. This motion is estimated based on the coarse motion of each segment. For each segment, an edge correspondence search is performed to find the opposite edges of a segment along a search direction. All pixels along the search direction that lie between the corresponding edges, must belong to the segment. The direction of the correspondence search is determined by the coarse motion of the segment. Each pixel of each segment thus gets the $[\Delta x, \Delta y, \Delta z]$ vector, thereby producing $\mathbf{V}_{n,n+1}$.

4: Finally, when the tensor $\mathbf{V}_{n,n+1}$ is known, the synchronized depthmap can be generated using Equation (5.4). Steps 1 to 3 use $\mathbf{I}_n$ and $\mathbf{I}_{n+1}$ to estimate $\mathbf{V}_{n,n+1}$. In case $\mathbf{I}_{n+1}$ is not available, $\mathbf{V}_{n,n+1}$ is instead predicted from the preceding frames, as shown in Figure 5.4, using Equation (5.5):

$$\mathbf{V}_{n,n+1} = -(\frac{\mathbf{V}_{n,n-1}}{2} + \frac{\mathbf{V}_{n,n-2}}{2}) \,. \tag{5.5}$$

Chapter 6

The LIFE System Framework and Testbed

The preceding chapters covered multi-camera synchronization and calibration - important and highly specific processes within multi-camera systems. This chapter covers the design and implementation of the LIFE system, which is a multi-camera based end-to-end testbed for 3D and Light Field recording, streaming and presentation. The LIFE system was developed at Mid Sweden University in conjunction with the theoretical work described in the preceding chapters. The LIFE system provides for the recording, processing, distribution and presentation of 3D data.

6.1 Overview

The concept behind the LIFE system is that it should serve as a testbed for evaluating aspects of Light Field capture. As such, flexibility and ease of modification were important goals throughout the development of this system. A number of existing multi-camera based systems [MP04, YEBM02, BK10] are designed to support a specific Light Field processing chain with homogeneous camera arrays and direct connection between cameras and rendering computers. The LIFE system, in contrast, supports diverse camera configurations and distributed processing, and separates recording of data from streaming, processing and presenting.

The system is based on a high-level framework, illustrated in Figure 6.1. The framework describes an end-to-end Light Field system, which includes recording, sending and presenting of 3D information. The LIFE framework follows a segmented scheme, isolating all devices and software processes into component blocks and system domains. This segmentation ensures that the system is implemented in a modular way. The modular implementation of domains also ensures that the system can adapt various configurations and represent not only fixed multi-camera systems, but also smart-camera and distributed camera networks.

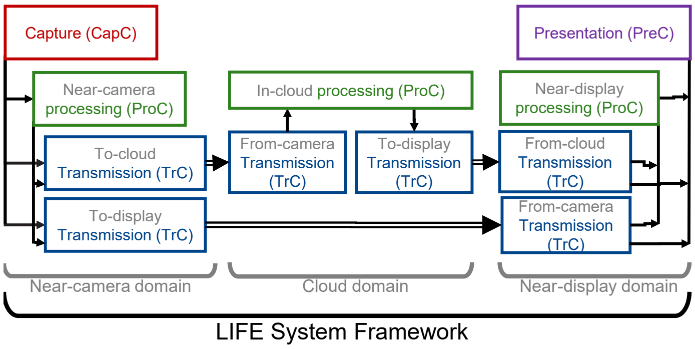

Capture (CapC)
Presentation (PreC)
Near-camera processing (ProC)
In-cloud processing (ProC)
Near-display processing (ProC)
To-cloud Transmission (TrC)
From-camera Transmission (TrC)
To-display Transmission (TrC)
From-cloud Transmission (TrC)
To-display Transmission (TrC)
From-camera Transmission (TrC)
Near-camera domain
Cloud domain
Near-display domain
LIFE System Framework

Figure 6.1: High-level view of the LIFE framework, presenting its three high-bandwidth domains and five component types.

6.2 High-Level Framework

The framework shown in Figure 6.1 consists of three domains. Each domain is an environment where devices and processes can exchange data over high bandwidth with low latency. Between domains, communication bandwidth is reduced, and latency is increased. The three domains are identified by their purposes and the devices they contain. The *near-camera domain* is focused on recording, and includes the cameras and other sensors. The *cloud domain*, which focuses solely on processing, includes larger computational resources than the other two domains. The *near-display domain* focuses on presentation, and includes the displays and rendering solutions.

The software and hardware modules in each domain are grouped into one of five possible components: Capture Component (CapC), Transmission Component (TrC), Processing Component (ProC), and Presentation Component (PreC). The CapC is unique to the near-camera domain, and includes all the cameras, as well as the software and hardware necessary to control the cameras and access the recorded frames. The TrC contains the software and hardware required for distributing data across the domains. The ProC contains processes that either generate new information from the recorded data, or change the recorded data. As seen in Figure 6.1, all three domains share ProC and TrC. Finally, the PreC is unique to the near-display domain, and contains the presentation devices, the software and hardware used to control these devices, and the rendering process. The division of responsibilities and hardware dependencies among components ensures that software or hardware changes within one component do not affect the other components.

6.3 Testbed Implementation

The testbed covers all three domains of the LIFE framework. The focus of this section is on the implementation of the near-camera domain, as shown in Figure 6.2.

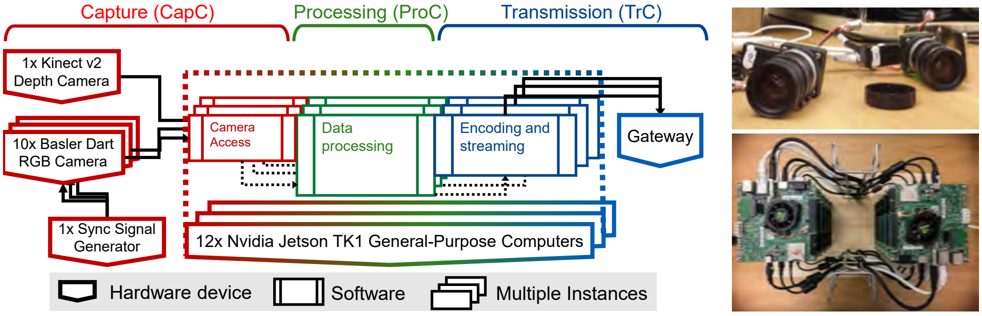


Figure 6.2: Implementation of a multi-camera system in the LIFE testbed. Architecture of the near-camera domain (left), and the camera and computer hardware (right).

This domain is implemented as a multi-camera system with color and depth cameras, dedicated computers, and a flexible software stack. The multi-camera system is described in Section 6.3.1, and the implementation of the other two LIFE framework domains is briefly discussed in Section 6.3.2.

6.3.1 Multi-Camera System

The multi-camera system makes up the near-camera domain of the LIFE framework. In order to support the processing and transmission components, the multi-camera system consists of cameras paired to dedicated single-board computers. This allows each computer to serve as the host for processing and transmission software. Moreover, the camera and computer pairing allows for encapsulating the camera API into a generic interface for configuring cameras and retrieving images. This approach also provides larger bandwidth for each camera, unlike in multicamera systems such as [YEBM02, MP04, BK10] which force several cameras to share bandwidth to a connected computer. The processing power of the camera-paired computers in this multi-camera system allows to compress the raw camera images for further streaming over shared network connections. Streaming over the Internet is enabled by embedding the open-source GStreamer framework [Dev18] into the LIFE framework's transmission component.

The system uses ten Basler daA1600-60uc cameras [AG17], which have a 1.92 megapixel sensor and a USB 3.0 interface in a compact (7.2 mm x 27 mm x 27 mm) housing. The global shutter sensor and a synchronization signal input mean that the cameras can be perfectly synchronized from an external periodic signal generator. By using several external generators, a controlled synchronization can be introduced into the system. The CS-type lens mount allows for a wide variety of lenses with different apertures, focal lengths, and resolutions. The system also uses a Microsoft Kinect v2 depth camera [Mic18], which provides a 2.07 megapixel 2D color image and a 0.22 megapixel depthmap. The cameras are each connected to one of eleven Nvidia Jetson TK1 single-board computers [Cor18]. These computers are chosen because of their compact size, high-bandwidth interfaces, Linux operating system, and the on-board processor and Programmable Graphics Processing Unit (PGPU). A twelfth Jetson TK1 is included for software component testing purposes. Figure 6.2 shows the implementation's logical structure (left) and physical hardware (right). The hardware choices ensure that the already supported cameras are reconfigurable, and that adding a new camera into the system only requires adding another Linux-capable computer and providing the camera API encapsulation to the generic LIFE framework interface. In this way, adding or replacing cameras does not affect the processing and streaming software on the paired computers.

6.3.2 Distribution and Presentation Systems

The cloud domain is implemented as a cloud-based video stream distribution system, using virtual instances in a parallel computing data center. The primary purpose of this distribution system is to provide a video processing environment and a video transcoding service. This cloud-based system relies on one master instance to receive and decode incoming video streams, and on a pool of slave instances for re-encoding, processing, and sending video streams to the presentation system. This presentation system is implemented on a regular computer connected to a display device, receiving video data via GStreamer's standard components. The presentation system can receive streams from both the multi-camera system and the distribution system, because the same stream format is used for all cross-system interfaces.

Chapter 7

Contributions

In previous chapters, components of the multi-camera capture process have been described, including problems, research questions, and proposed theories related to synchronization, calibration, and depthmap correction. These problems and research questions have been addressed by the contributions that underpin this dissertation. This chapter presents the novelties and evaluation results of the following contributions, which investigate components of the capture process and present a framework for Light Field system evaluation. Each contribution is a separate paper with its own experimental setup, aims, and results. In total, the contribution list is as follows:

I. Contribution I - Modeling Depth Uncertainty of Desynchronized Multi-Camera Systems.

II. Contribution II - Assessment of Multi-Camera Calibration Algorithms for Two-Dimensional Camera Arrays Relative to Ground Truth Position and Direction.

III. Contribution III - Estimation and Post-Capture Compensation of Synchronization Error in Unsynchronized Multi-Camera Systems.

IV. Contribution IV - LIFE: A Flexible Testbed for Light Field Evaluation.

7.1 Contribution I

Modeling Depth Uncertainty of Desynchronized Multi-Camera Systems

Contribution I introduces the concept of depth uncertainty in unsynchronized multi-camera systems. A model is presented to describe depth uncertainty, using general parameters of the camera system and the recorded scene. This contribution covers the concepts introduced in Chapter 3.

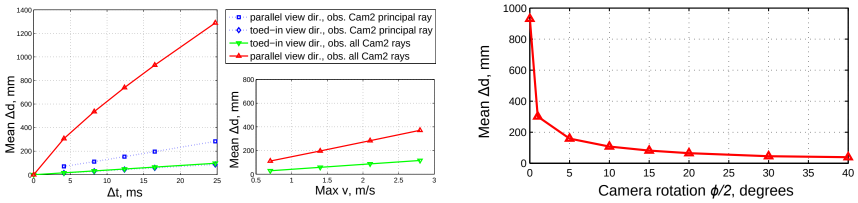


Figure 7.1: Left: Depth uncertainty Δd, given varying camera desynchronization and varying maximum speed of scene elements for parallel and $\phi = 20^\circ$ -convergent view directions. Right: Mean Δd along all rays of camera 1, for varying convergence ϕ of both cameras (indicated rotation $\phi/2$ for camera 1, with simultaneous negative rotation $-\phi/2$ on camera 2).

7.1.1 Novelty

The novelty of this contribution lies in (1) revealing and using a new model that relates camera system parameters and synchronization errors to the extent of possible errors in depth estimation, and (2) introducing the concept of depth uncertainty - a quantifiable metric describing a property of a multi-camera system.

7.1.2 Evaluation and Results

Two types of experiments were performed as part of this work. The first experiment type focused on using the depth uncertainty model to demonstrate how synchronization and rotation affect the depth uncertainty of a multi-camera system. The second experiment type examined the depth uncertainty model for computational shortcuts, i.e. viability of using a subset of camera rays instead of the whole set. For the experiments, a two-camera setup was modeled using realistic parameters (sensor resolution, view angle, camera placement, synchronization error) that described a basic multi-camera system. The two-camera scenario was used because the model in this contribution relies on depth uncertainty minimization across camera pairs.

The results of the simulations, illustrated in Fig. 7.1, showed that synchronization error and scene element speed have a linear relation to the system's depth uncertainty. Convergence between cameras was seen to have a non-linear effect on mean depth uncertainty. It was concluded that a multi-camera system with parallel camera view directions is significantly more affected by synchronization errors, compared to a camera system with converged camera view directions. This conclusion also served as a reason for using the depth uncertainty model to examine the capabilities of a multi-camera capture system.

During depth uncertainty simulations, the computational shortcut tests showed significant differences in the depth uncertainty results. The ray-based depth uncertainties generated during the test had both different distributions and different magnitudes between the whole ray set and the ray subset scenarios. This implies that, in order to compute the depth uncertainty, the proposed depth uncertainty model requires that all ray interactions in the camera system are computed.

7.1.3 Author Contribution

Elijs Dima originally proposed the idea of the depth uncertainty model, derived the model, and initiated the idea to investigate computational shortcuts by using ray set reduction. He is the main author of the article. Prof. Mårten Sjöström and Dr. Roger Olsson provided feedback during the process of deriving the model, suggested corrections to the manuscript, and contributed advice on creating article content and addressing reviewer feedback.

7.2 Contribution II

Assessment of Multi-Camera Calibration Algorithms for Two-Dimensional Camera Arrays Relative to Ground Truth Position and Direction

Contribution I described a model for which camera parameters are the input data. Typically, camera parameters are obtained by performing camera calibration. Contribution II examines a number of publicly available multi-camera system calibration tools that characterize target-based and targetless calibration. The aim of this contribution is to verify the accuracy of calibration tools by using a custom dataset with externally obtained ground truth for camera parameters, and to assess whether self-calibration tools can match the accuracy of checkerboard based calibration tools. This contribution utilizes the concepts presented in Chapter 4.

7.2.1 Novelty

The novelty of this research work lies in the following: (1) the assessment of several freely available and popular calibration tools based on their correspondence to ground truth camera parameters instead of self-reported reprojection errors of opportunistically selected points; (2) the introduction and use of a calibration dataset that has strict *a priori* constraints on the ground truth of camera parameters. These constraints are produced as a result of the dataset generation method, are fundamental to the dataset, and allow for clear classification of correct and incorrect parameter estimates from calibration methods.

7.2.2 Evaluation and Results

The dataset used for evaluation was produced by setting three cameras on a calibrated motorized dolly, to take images from 15 different viewpoints. For the calibration methods, the dataset was presented as if recorded by a different camera at each viewpoint, and the camera positions were verified with a laser rangefinder. This provided ground truth constraints for camera lens parameters, principal point, and

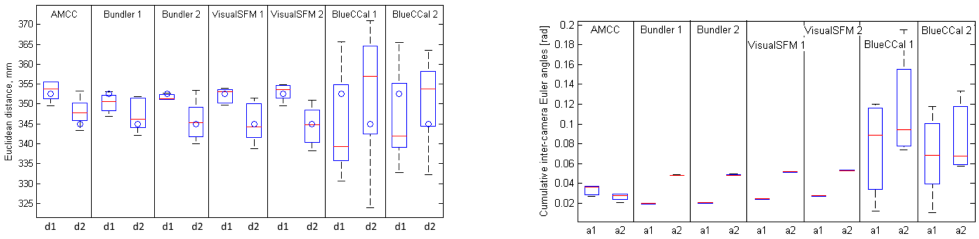


Figure 7.2: Comparison of target-based (AMCC [Zha00]) and targetless (Bundler, VisualSFM, BlueCCal [SSS06, Wu13, SMP05]) camera calibration methods, measured on a rigid 3-camera rig. Left: estimated distances between camera centers. Circle shows ground truth. Right: estimated rotation difference a_1 between rigidly mounted cameras 1 and 2, and a_2 between cameras 2 and 3. Box plots show median, 25th and 75th percentile, whiskers show minimum and maximum.

positions, all of which are key contributors to rendering errors in multi-camera systems [SSO14]. The target-based calibration was represented by the method of Zhang et al. [Zha00], which forms the basis for camera calibration processes in Matlab and OpenCV software. Zhang et al's target-based method was compared with Snavely's [SSS06], Svoboda's [SMP05] and Wu's [Wu13] targetless calibration methods.

An analysis of the calibration results (shown in Figure 7.2) indicated that the two targetless Structure from Motion (SfM) calibration methods [SSS06, Wu13] outperformed the third targetless calibration method [SMP05], especially when estimating camera position and rotation. Moreover, the accuracy levels of estimates for camera position and rotation were quite similar in the target-based method and the two SfM methods. Furthermore, the estimation of lens distortion coefficients was equally accurate in both the checkerboard and SfM-based methods. However, the target-based method did estimate an additional parameter - the location of the principal point on the camera sensor plane.

Overall, the results indicate that the tested SfM methods perform equally well with and without a checkerboard target present in the scene. The scene is shown in Chapter 4.2, Figure 4.1. The scene contains objects with texture and clear edges at multiple depths and positions. Checkerboard placement is the only factor that changes across multiple captures of the scene. Moreover, except for estimating the principal point, the SfM methods perform as well as the checkerboard calibration method.

7.2.3 Author Contribution

Elijs Dima is the main author of the article. Prof. Mårten Sjöström proposed the idea of comparing calibration methods. Elijs Dima proposed and produced the dataset with ground truth constraints, selected and applied the calibration methods, conducted the experiments, analyzed the results, and wrote the article. Prof. Mårten Sjöström provided advice on organizing the tests and analysing the results, and

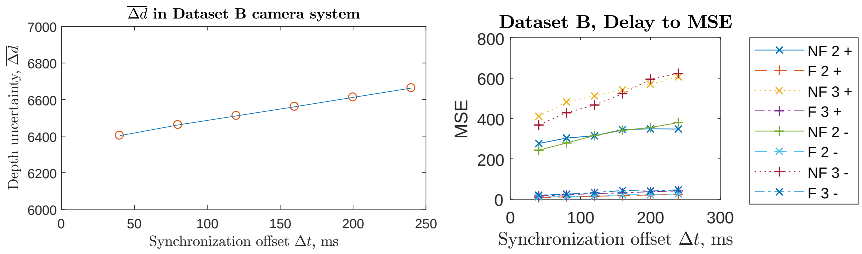


Figure 7.3: Left: Relation between depth uncertainty $\overline{\Delta d}$ and synchronization error Δt. Right: DIBR rendering quality, given synchronization error Δt between depthmap and texture. "F" indicates disocclusion filling, "NF" indicates no filling, numbers indicate render target view, "$+, -$" indicates positive or negative Δt, respectively.

played a correctional role in manuscript writing and addressing reviewer feedback. Dr. Roger Olsson contributed advice on the content and structure of the article.

7.3 Contribution III

Estimation and Post-Capture Compensation of Synchronization Error in Unsynchronized Multi-Camera Systems

Contribution III addresses the need for correcting depth data in a multi-camera system with incorrect synchronization between cameras, on the basis of the depth uncertainty model introduced in Contribution I. This contribution further introduces a method for detecting and compensating synchronization errors in multi-camera systems. The primary focus is on multi-camera systems consisting of RGB and depth cameras. The contribution utilizes the concepts presented in Chapters 3 and 5.

7.3.1 Novelty

The novelty of this work is twofold: (1) the depth uncertainty model from Contribution I is extended and demonstrated to correlate with rendered image quality; (2) a new method for estimating and compensating synchronization errors in Color and Depth (RGB-D) systems is introduced, consisting of two discrete parts: a correspondence-based estimation of synchronization error, and a weighted frame interpolation approach for non-destructive depthmap re-synchronization.

7.3.2 Evaluation and Results

Three experiments were performed in this work. The first experiment aimed to validate the depth uncertainty model by examining whether there is a correlation between depth uncertainty and objective measurements of rendered virtual images.

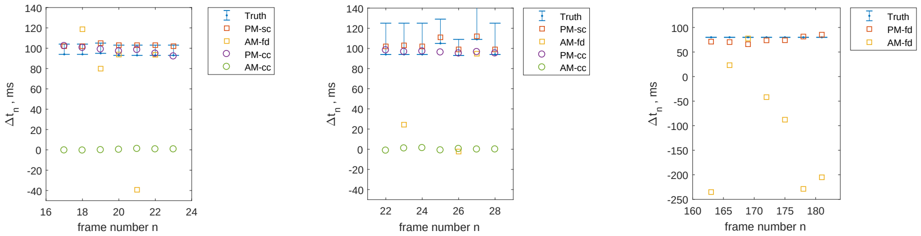


Figure 7.4: Synchronization error estimation accuracy for datasets with synchronization error Δt_n between a color camera and a depth camera at frame n. Left: both cameras have constant framerate at 7.5 Hz. Ground truth known with 10 ms accuracy. Middle: same camera system with longer exposure time. Right: synchronized multiview-plus-depth dataset, with $\Delta t = 80$ ms obtained by frame offset between color and depth images. "Truth": ground truth interval. "PM": proposed method. "AM": method from [AKF$^+$17b].

Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index (SSIM) and Mean Squared Error (MSE) were used as objective image quality metrics.

Results from the first experiment, shown in Figure 7.3, revealed that an increase in content-independent depth uncertainty correlates with a decrease in quality of reprojected images. The experiment relies on the assumption that the camera calibration parameters are correct. Besides calibration, the reprojection quality is affected by four other factors: synchronization error, reprojection baseline, disocclusion inpainting, and the direction (future or past) of the temporal offset between reprojection target and source. Of these factors, synchronization error affects the depth uncertainty. On the tested datasets, synchronization error had a greater effect (by 2 to 3 dB PSNR) on image quality than reprojection baseline or temporal offset direction, but a smaller effect than disocclusion inpainting. Disocclusion inpainting has a greater effect than synchronization error due to the pixel-based nature of objective quality metrics such as PSNR, SSIM, and MSE.

The second experiment assessed the proposed method for estimating synchronization error on RGB-D datasets, and compared it against a state-of-the-art existing method. According to the results shown in Figure 7.4, the proposed method was more reliable, with an average offset estimation accuracy of 83% to 95% of the true offset in the datasets. Unlike the state-of-the-art method, the proposed method did not require temporally long trajectories or rapidly changing in-scene motion.

The third experiment assessed the proposed method for synchronization error compensation together with four other methods based on optical flow estimation [KRN97, Far03], image morphing [SD96] and displacement field interpolation [Thi98]. The improvement gained from the proposed synchronization error compensation process was also demonstrated by comparing the reprojection and foreground isolation results, as shown in Figure 7.5. The depthmaps corrected using the proposed compensation method produced as good results as synchronous depthmaps, when used for foreground texture isolation and view reprojection. In contrast, uncorrected (unsynchronized) depthmaps caused notable errors in foreground-to-texture align-

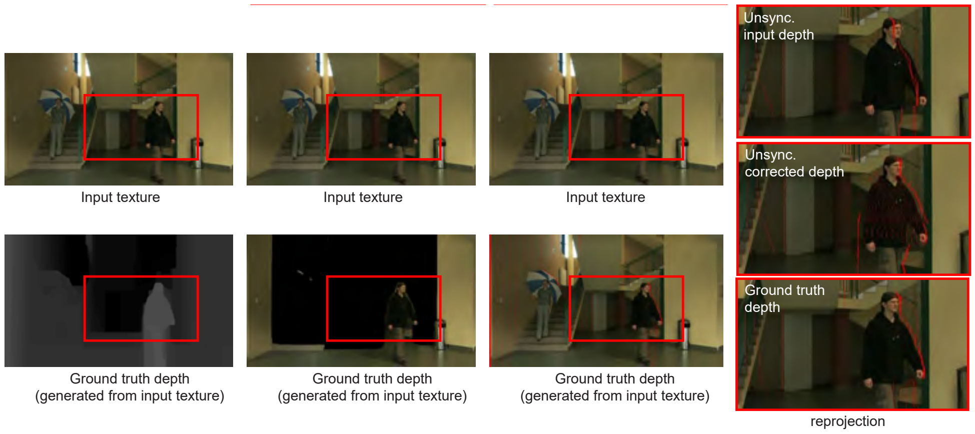


Figure 7.5: Impact of using unsynchronized, compensated, and synchronized depth in depth-based foreground isolation and view reprojection.

ment and incorrect disocclusion positions in reprojected images.

7.3.3 Author Contribution

Elijs Dima is the main author of the article, responsible for the idea of compensating the synchronization error in RGB-D systems, proposing the correction method, and implementating it. He is also responsible for running the experiments, analyzing the results, writing the article, and addressing reviewer feedback. Yuan Gao, the second author of the article, contributed calibration data for a test dataset and is responsible for initial implementation of synchronization estimation, and writing about motivation and test data production in an early version of the manuscript. Prof. Mårten Sjöström and Prof. Reinhard Koch contributed with feedback on the overall idea and experimentation requirements, and suggested corrections to the manuscript. Prof. Sjöström also provided advice regarding the necessity for additional development and experimentation, and on addressing reviewer feedback. Dr. Roger Olsson and Dr. Sandro Esquivel contributed with advice on the article content and suggested corrections to the manuscript.

7.4 Contribution IV

LIFE: A Flexible Testbed for Light Field Evaluation

Contribution IV introduces the LIFE framework - a high-level framework for organizing hardware and software components into a flexible end-to-end Light Field system. The contribution also presents implementation details and preliminary evaluation of the system's real-time ability.

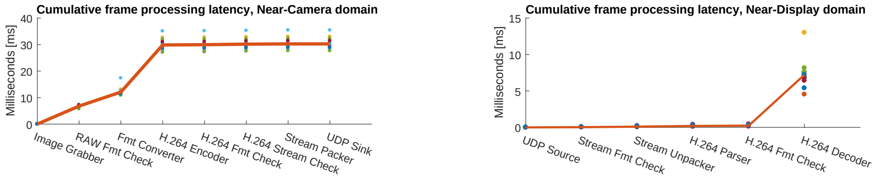


Figure 7.6: Cumulative latency for frame processing. Red line shows average frame latency, dots show individual frame latency measurements.

7.4.1 Novelty

This work is novel in two ways. First, the new LIFE framework is designed to encourage modularity and flexibility in the system, and can be implemented to cover a wide range of capture, distribution, and presentation system configurations. Second, the presented system is the first existing implementation of the LIFE framework and is used to both develop the framework and demonstrate the framework's features and viability. The implementation is also a novel testbed, designed specifically to enable the evaluations of Light Field capture, distribution, and presentation.

7.4.2 Evaluation and Results

To evaluate the framework, the system's real-time ability was measured for implementation of two system domains: the near-camera domain and the near-display domain. Both domains are described in Chapter 6. The real-time limit was set at 40 ms, based on the maximum camera recording rate of 25 Hz at the highest supported sensor resolution. The processing latency was measured for each component of a live stream from camera to display over a public IP network.

The results, shown in Figure 7.6, revealed that video stream coding, processing, and network encapsulation had a cumulative latency of less than 40 ms in each domain. In the near-display domain, this included sending the frame to the display driver. In the near-camera domain, this included accessing the camera buffer, but did not include the time that the camera itself took to expose the sensor (9.5 ms) and de-bayer the recorded image (20 ms). Within both domains, the formatting and coding of the frame had the largest impact on latency (approximately 10 to 20 ms), and the stream formatting components added negligible latency.

7.4.3 Author Contribution

Elijs Dima contributed to the planning and design of the LIFE framework; the planning, component selection, construction, and software development for the near-camera prototype; interface planning between the near-camera and cloud prototypes; and performance of the preliminary recording and streaming tests. M. Kjellqvist contributed to the planning and software development for implementation of the near-camera domain. Z. Zhang and L. Litwic contributed to the planning, development and tests of the connected cloud transcoding prototype, and to drafting the section on cloud domain in the article. M. Sjöström and R. Olsson managed the resource allocation and prioritization of implementation work, and provided feedback on writing the article. L. Rasmusson and L. Flodén contributed with advice on smart camera systems and distributed, networked surveillance camera technologies.

Chapter 8

Conclusion and Outlook

The previous chapters summarized the contributions of research papers and provided an overview of the developed multi-camera system. This chapter presents an overview of this thesis, reviews the link between the completed work and the initial purpose and goals, and discusses directions for future research and the potential impact of the contributions of this work.

8.1 Overview

This work and its contributions are aimed at extending the knowledge of 3D scene capture using multi-camera systems. To this end, investigations were conducted on camera calibration and synchronization methods, in conjunction with the development of a multi-camera system for future Light Field evaluations.

Investigations into camera calibration identified the discrepancy between the usual assessment of calibration quality, and the ability of widely-used calibration methods to recover camera parameters. One of the contributions of this work was to compare the ability of various calibration methods to estimate the true camera parameters, and to compare the ability of self-calibration methods to match target-based calibration methods.

Investigations into synchronization led to the proposal of a new model of depth uncertainty, detailing how synchronization error affects the capture results in multi-camera systems. The model was experimentally tested and used to reach conclusions on how a camera system's layout and synchronization affects its ability to accurately estimate the positions of moving objects. Furthermore, a re-synchronization method was proposed for solving synchronization errors in RGB-D camera systems during the post-recording stages instead of during recording. The proposed method was experimentally shown to improve the fusion of depth and color data in datasets with real and simulated synchronization errors.

The development of a multi-camera system led to the introduction of the LIFE system framework and testbed implementation, which includes the multi-camera system as the component for scene acquisition. The multi-camera system was designed to be easily re-configurable and modifiable. Each camera was paired with a dedicated computer and a modular software stack, to allow for the general-purpose processing and streaming of recorded data. The system was able to record, encode, and stream out the camera views within real-time latency constraints.

8.2 Outcome

The specific goals of this work were outlined in Section 1.4. Presented here is a summary of the outcomes with respect to each goal.

Goal 1: Design and construct a flexible multi-camera system testbed.

The LIFE framework and the multi-camera based implementation from Contribution 4 were developed in parallel with the other goals. The resulting system contains not only a multi-camera system, but also a cloud processing prototype and a light-weight presentation system. The multi-camera system is able to record and also process and stream live camera data over the Internet.

Early findings on synchronization in Contribution 1 led to the decision of using cameras with support for an external synchronization signal, due to a lack of ready-to-use solutions for software synchronization of data. During system planning and construction, it was determined that available ToF sensors do not support synchronization, thereby reinforcing the need to compensate synchronization errors specifically in recorded depth data. This requirement was addressed in Contribution 3.

Investigations on calibration in Contribution 2 resulted in the decision to separate calibration into a stand-alone software module within the system's control layer. This enables seamless transition between calibration methods and non-autonomous tools that could not be deployed on the general-purpose computers connected directly to the cameras. The LIFE framework's design and the computational power of the general-purpose computers allow for the future development of autonomous, on-device calibration.

The flexibility of the developed multi-camera system is evident through several aspects: (1) Processes and devices of the multi-camera system are separated into framework domains and components. This allows for the change of system capabilities at a hardware and software level with low implementation overhead. (2) The presence of per-camera computers and full Internet connectivity allows for the distribution of processing operations. Video and image processing can occur in the camera system itself, in a connected cloud, or in a connected end-user system. It can also be distributed among all three equally. (3) The multi-camera system uses off-the-shelf cameras, computers, and an open source streaming framework. This increases the potential for compatibility between this multi-camera system and third-party processing or utility applications. (4) The multi-camera system supports hardware and software synchronization, per-camera computing, and has adjustable camera mounts and small camera sizes. Therefore, it is possible to use the LIFE multi-camera system as a stand-in for various acquisition systems, ranging from sparse surveillance camera networks to small and dense 360-video camera systems. The developed multi-camera system is not only flexible, but also able to serve as a testbed for Light Field capture.

Goal 2: Investigate the advantages and drawbacks of multi-camera calibration solutions, and assess the ability to recover the true camera parameters via calibration.

The field of camera calibration was found to be relatively mature, with several widely accepted and widely used methods already built into standard image processing tool collections. A research gap was identified by the lack of strict comparisons between self-calibration and target-calibration methods, and by the lack of calibration result comparisons against ground truth of camera parameters. A comparison of methods was conducted based on ground truth, and the results were published in Contribution 2. The results revealed the relative capabilities of target-based and targetless calibration methods, and the performance of freely available calibration methods with regard to their accuracy in estimating camera parameters. Further development of new calibration methods was considered to be outside the scope of this work, given the maturity of the field and the abundance of existing methods being commonly used.

Goal 3: Investigate the consequences of inaccurate synchronization before or during recording in a multi-camera system.

The field of camera synchronization was investigated and it was found that synchronization is a less explored field than calibration, with notable research gaps in relating synchronization to geometric multi-camera models. A model for mapping synchronization error to depth estimation error was presented in Contribution 1. The proposed model links synchronization error to the geometric pinhole camera model. This facilitates the modeling of synchronization error consequences in multi-camera system design and construction stages.

Goal 4: Propose a multi-camera synchronization solution for scenarios when accurate synchronization before or during recording is not possible.

A research gap was identified in connecting outputs of unsynchronized camera systems to the inputs of standard multi-view geometry applications. In particular, depth-sensing cameras were found to often be a source of synchronization error in multi-camera systems. Contribution 3 presented a new method for ensuring post-recording synchronization between depth and color data recorded by separate cameras. The proposed model estimates the synchronization error between depth and color cameras, and compensates the error in depth data in order to bring the depth data to a synchronized state.

Purpose: Contribute to the knowledge and understanding of multi-camera systems within the context of Light Field acquisition.

This purpose has been achieved in two ways. First, through completion of Goals 2, 3, and 4, new knowledge has been contributed to the pre-recording and post-recording stages of the process of multi-camera Light Field capture. Second, through completion of Goal 1, the constructed LIFE system serves not only as an example of a decentralized, distributed-processing camera system, but also as a research testbed using which further investigations into Light Field acquisition can be conducted.

8.3 Impact

Since this work is grounded in computing research, there is a responsibility to seriously consider both the positive and negative impact of the produced knowledge [HWB+18]. This work likely has both direct and indirect consequences. The direct consequences mainly relate to the scientific impact of this work, and the indirect consequences mainly encompass the social and ethical impact of the knowledge developed in this work.

8.3.1 Scientific Impact

The potential scientific impact relates to the immediate contributions to multi-camera systems, which are detailed in terms of multi-camera system construction, depth uncertainty estimation, camera calibration, and synchronization. The proposed framework for a Light Field evaluation system serves as a template and a reference for developing flexible multi-camera based systems. Such systems can be used as testbeds for Light Field research and therefore accelerate such research. The proposed depth uncertainty model parametrizes the cost of synchronization error in a multi-camera system. This cost can be used in the planning stages of future multi-camera systems, to ensure that the camera system implementation matches the context requirements. As regards calibration, the difference between the ground truth of camera parameters and the standard quality metric for calibration (the reprojection error) has been highlighted. This can lead to an evaluation of the approaches used in calibration methods and the development of metrics more closely correlated with the true camera parameters. Finally, the proposed re-synchronization method can be used to make non-synchronized multi-camera systems compatible with a wide range of applications that require synchronized datasets.

8.3.2 Ethical and Social Impact

The indirect consequences of this work are potential changes in any field and application that relies on multi-camera systems. Section 1.1.1 described how multi-camera systems are applied in surveillance [OLS+15, DBV16], robot and machine vision [HKH+12, KDBO+05, KSC15, HLP15, LFP13], human and group behavior analysis [JLT+15a, OCK+13], and multimedia entertainment production [LMJH+11, ZEM+15, Pan17, tL17, Fac17, Goo17]. At a higher abstraction level, any system with a camera may gain *more information about what is being observed* if more cameras are added into the system. The trend of turning single-camera systems into multi-camera systems is evident from the use of multi-camera based photography effects [Mö18] and depth-based facial profiling [deA17] in modern smartphones. It is evident that contributions to multi-camera systems can have far-reaching consequences.

The overall purpose of this work is to expand the existing knowledge on multi-camera systems. More importantly, parts of this work contribute to the use of multi-camera systems and on-camera processing, and to the ability to recover highly synchronized multi-view and depth data from unsynchronized cameras.

On the one hand, these aspects can benefit surveillance camera systems, since these systems often rely on low-cost, unsynchronized cameras while being used for vehicle and people tracking. Moreover, on-camera processing allows surveillance camera systems to avoid privacy laws: while it may be illegal to record and store video footage of people in public spaces, it may be legal to record a video, analyze it on a camera, and only store results of the analysis, such as the total number of people seen in an area or group behavior patterns at different times. As multi-camera systems and surveillance camera networks become increasingly present, communities and governments will need to strictly define which kinds of image analysis results are too specific, and whether the benefit of using surveillance systems is justified. It may be necessary to ensure that implementations of on-camera processing do not breach the European Union's General Data Protection Regulation (GDPR) [Alb16].

On the other hand, these aspects have a positive effect in several contexts. The investigations into the consequences of synchronization error may improve the design of multi-camera systems, and reduce the use of needlessly expensive cameras. This will improve the cost efficiency of multi-camera systems used in manufacturing and entertainment, without a reduction in human jobs. The multi-camera systems will still need to be designed and constructed, just with more specific hardware requirements. Multi-camera systems are already widely used in systems such as self-driving vehicles, where the accuracy of recorded data has a direct effect on human safety. Such systems could benefit from an increased focus on camera synchronization and accuracy in 3D estimation of recorded objects. Moreover, the LIFE testbed can be used to accelerate research on multi-camera and Light Field systems. This implies that improvements in current multi-camera systems and Light Fields may occur sooner and be incorporated into multimedia production, VR, AR, and industrial monitoring applications.

8.4 Future Work

Given the definition of the capture process and the applications of multi-camera systems mentioned in Section 1.1, the scope for future work is vast. With a functioning multi-camera based Light Field evaluation testbed, directions for potential research are numerous. They include optimization of camera placement, management of inter-system and system-to-client communications, and distributed image-and-depth processing on compute modules. Likewise, the proposed synchronization model can be extended by parametrizing the camera shutter speed and motion blur, or using the model in a cost function for multi-camera layout optimization.

Furthermore, due to the LIFE testbed's Internet-connected design and capability to record and transmit data simultaneously, it can be used as a component in a larger Light Field communication system, enabling research on Light Field transmission, coding, and presentation. Moreover, the flexibility of the testbed opens possibilities for research on other 3D-related applications, such as 360 video production, depth-enhanced photography, and augmented reality.

List of Figures

List of Tables

Bibliography

[AB91] Edward H Adelson and James R Bergen. *The Plenoptic Function and the Elements of Early Vision*. Vision and Modeling Group, Media Laboratory, Massachusetts Institute of Technology, 1991.

[AG17] Basler AG. Basler dart usb 3.0 - product documentation. Technical report, Basler AG, 2017. `https://www.baslerweb.com/en/sales-support/downloads/document-downloads/basler-dart-usb-3-0-users-manual/`, Retrieved 05/04/2018.

[AKF$^+$17a] Cenek Albl, Zuzana Kukelova, Andrew Fitzgibbon, Jan Heller, Matej Smid, and Tomas Pajdla. On the two-view geometry of unsynchronized cameras. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, July 2017.

[AKF$^+$17b] Cenek Albl, Zuzana Kukelova, Andrew Fitzgibbon, Jan Heller, Matej Smid, and Tomas Pajdla. On the two-view geometry of unsynchronized cameras. *arXiv preprint arXiv:1704.06843*, 2017.

[Alb16] Jan Philipp Albrecht. How the GDPR will change the world. *Eur. Data Prot. L. Rev.*, 2:287, 2016.

[BAW13] Tuba Bakıcı, Esteve Almirall, and Jonathan Wareham. A smart city initiative: the case of barcelona. *Journal of the Knowledge Economy*, 4(2):135–148, 2013.

[BBRP12] Beverly A Bondad-Brown, Ronald E Rice, and Katy E Pearce. Influences on tv viewing and online user-shared video use: Demographics, generations, contextual age, media use, motivations, and audience activity. *Journal of Broadcasting & Electronic Media*, 56(4):471–493, 2012.

[BCFS06] M Bigas, Enric Cabruja, Josep Forest, and Joaquim Salvi. Review of CMOS image sensors. *Microelectronics Journal*, 37(5):433–451, 2006.

[BEMN09] Rune H Bakken, Bjørn G Eilertsen, Gustavo U Matus, and Jan H Nilsen. Semi-automatic camera calibration using coplanar control points. In *Proceedings of NIK Conference*, pages 37–48, 2009.

[BETVG08] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-up robust features (surf). *Computer Vision and Image Understanding*, 110(3):346–359, 2008.

[BK10] Tibor Balogh and Péter Tamás Kovács. Real-time 3D light field transmission. In *Real-Time Image and Video Processing*, volume 7724, page 772406. International Society for Optics and Photonics, 2010.

[BMNK13] Kai Berger, Stephan Meister, Rahul Nair, and Daniel Kondermann. A state of the art report on kinect sensor setups in computer vision. In *Time-of-Flight and Depth Imaging. Sensors, Algorithms, and Applications*, pages 257–272. Springer, 2013.

[Bou16] Jean-Yves Bouguet. Camera calibration toolbox for matlab. *URL http://www. vision. caltech. edu/bouguetj/calib_doc*, 2016.

[Bra00] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.

[Bro66] Duane C Brown. Decentering distortion of lenses. *Photogrammetric Engineering and Remote Sensing*, 1966.

[BRS$^+$11] Kai Berger, Kai Ruhl, Yannic Schroeder, Christian Bruemmer, Alexander Scholz, and Marcus A Magnor. Markerless motion capture using multiple color-depth sensors. In *VMV*, pages 317–324, 2011.

[CI02] Yaron Caspi and Michal Irani. Spatio-temporal alignment of sequences. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(11):1409–1424, 2002.

[Cor18] NVIDIA Corporation. Jetson TK1, 2018. `http://www.nvidia.com/object/jetson-tk1-embedded-dev-kit.html`, Retrieved 05/04/2018.

[DBV16] Ruofei Du, Sujal Bista, and Amitabh Varshney. Video fields: Fusing multiple surveillance videos into a dynamic virtual environment. In *Proceedings of the 21st International Conference on Web3D Technology*, pages 165–172. ACM, 2016.

[DDM$^+$15] Marek Domański, Adrian Dziembowski, Dawid Mieloch, Adam Łuczak, Olgierd Stankiewicz, and Krzysztof Wegner. A practical approach to acquisition and processing of free viewpoint video. In *Picture Coding Symposium (PCS), 2015*, pages 10–14. IEEE, 2015.

[deA17] Michael deAgonia. Apple's face ID [The iPhone X's facial recognition tech] explained. `https://www.computerworld.com/article/3235140/apple-ios/apples-face-id-the-iphone-xs-facial-recognition-tech-explained.html`, Retrieved 02/04/2018, 2017.

[DEGH12] Deepak Dwarakanath, Alexander Eichhorn, Carsten Griwodz, and Pål Halvorsen. Faster and more accurate feature-based calibration for widely spaced camera pairs. In *Second International Conference on Digital Information and Communication Technology and it's Applications (DICTAP)*, pages 87–92. IEEE, 2012.

[Dev18] GStreamer Developers. GStreamer: open source multimedia framework, 2018. https://gstreamer.freedesktop.org/.

[DPSL11] Ferran Diego, Daniel Ponsa, Joan Serrat, and Antonio M López. Video alignment for change detection. *IEEE Transactions on Image Processing*, 20(7):1858–1869, 2011.

[DZL06] Congxia Dai, Yunfei Zheng, and Xin Li. Subframe video synchronization via 3d phase correlation. In *IEEE International Conference on Image Processing (ICIP)*, pages 501–504. IEEE, 2006.

[EB13] Georgios D Evangelidis and Christian Bauckhage. Efficient subframe video alignment using short descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(10):2371–2386, 2013.

[ESGMRA11] FC Estrada-Silva, J Garduño-Mejía, and M Rosete-Aguilar. Third-order dispersion effects generated by non-ideal achromatic doublets on sub-20 femtosecond pulses. *Journal of Modern Optics*, 58(10):825–834, 2011.

[ESH$^+$12] Ahmed Elhayek, Carsten Stoll, Nils Hasler, Kwang In Kim, H-P Seidel, and Christian Theobalt. Spatio-temporal motion tracking with unsynchronized cameras. In *2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1870–1877. IEEE, 2012.

[Fac17] Facebook. facebook Surround 360 Open Edition. `https://facebook360.fb.com/facebook-surround-360/`, 2017.

[Far03] Gunnar Farnebäck. Two-frame motion estimation based on polynomial expansion. *Image Analysis*, pages 363–370, 2003.

[FBA$^+$94] Henry Fuchs, Gary Bishop, Kevin Arthur, Leonard McMillan, Ruzena Bajcsy, Sang Lee, Hany Farid, and Takeo Kanade. Virtual space teleconferencing using a sea of cameras. In *Proc. First International Conference on Medical Robotics and Computer Assisted Surgery*, volume 26, 1994.

[FBK10] Anatol Frick, Bogumil Bartczack, and Reinhard Koch. 3D-TV LDV content generation with a hybrid ToF-multicamera rig. In *3DTV Conference: The True Vision-Capture, Transmission and Display of 3D Video (3DTV-Con)*, pages 1–4, 2010.

[FBLF08] Francois Fleuret, Jerome Berclaz, Richard Lengagne, and Pascal Fua. Multicamera people tracking with a probabilistic occupancy map. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(2):267–282, 2008.

[Feh04] Christoph Fehn. Depth-image-based rendering (dibr), compression, and transmission for a new approach on 3d-tv. In *Electronic Imaging 2004*, pages 93–104. International Society for Optics and Photonics, 2004.

[Fit12] Bruce Fitter. Big s small 3d: What makes stereoscopic video so compelling? In *International Conference on 3D Imaging (IC3D)*, pages 1–5. IEEE, 2012.

[Gab17] Bernat Gabor. Camera calibration with opencv. `https://docs.opencv.org/2.4/doc/tutorials/calib3d/camera_calibration/camera_calibration.html`, 2017.

[GČH12] Vineet Gandhi, Jan Čech, and Radu Horaud. High-resolution depth maps based on ToF-stereo fusion. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 4742–4749, 2012.

[GGSC96] Steven J Gortler, Radek Grzeszczuk, Richard Szeliski, and Michael F Cohen. The lumigraph. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 43–54. ACM, 1996.

[GMCS12] Andreas Geiger, Frank Moosmann, Ömer Car, and Bernhard Schuster. Automatic camera and range sensor calibration using a single shot. In *International Conference on Robotics and Automation (ICRA)*, St. Paul, USA, May 2012.

[GML$^+$14] Patrik Goorts, Steven Maesen, Yunjun Liu, Maarten Dumont, Philippe Bekaert, and Gauthier Lafruit. Self-calibration of large scale camera networks. In *IEEE International Conference on Signal Processing and Multimedia Applications (SIGMAP)*, pages 107–116. IEEE, 2014.

[GNN15] Ginni Grover, Ram Narayanswamy, and Ram Nalla. Simulating multi-camera imaging systems for depth estimation, enhanced photography and video effects. In *Imaging Systems and Applications*, pages IT3A–2. Optical Society of America, 2015.

[Goo17] Google. Google VR | Jump. `https://vr.google.com/jump/`, 2017.

[HBNF15] Si Ying Hu, James Baldwin, Armand Niederberger, and David Fattal. I3. 2: Invited paper: A multiview 3d holochat system. In *SID Symposium Digest of Technical Papers*, volume 46, pages 286–289. Wiley Online Library, 2015.

[HD14] Daniel Howard and Danielle Dai. Public perceptions of self-driving cars: The case of berkeley, california. In *Transportation Research Board 93rd Annual Meeting*, volume 14, 2014.

[Hec87] Eugene Hecht. *Optics*. Addison Wesley, 2 edition, 1987.

[HHL+17] Christian Häne, Lionel Heng, Gim Hee Lee, Friedrich Fraundorfer, Paul Furgale, Torsten Sattler, and Marc Pollefeys. 3D visual perception for self-driving cars using a multi-camera system: Calibration, mapping, localization, and obstacle detection. *Image and Vision Computing*, 68:14–27, 2017.

[HHSP07] Abdenour Hadid, JY Heikkila, Olli Silvén, and M Pietikainen. Face and eye detection for person authentication in mobile phones. In *Distributed Smart Cameras, 2007. ICDSC'07. First ACM/IEEE International Conference on*, pages 101–108. IEEE, 2007.

[HK94] Glenn E Healey and Raghava Kondepudy. Radiometric CCD camera calibration and noise estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(3):267–276, 1994.

[HKH+12] Peter Henry, Michael Krainin, Evan Herbst, Xiaofeng Ren, and Dieter Fox. Rgb-d mapping: Using kinect-style depth cameras for dense 3d modeling of indoor environments. *The International Journal of Robotics Research*, 31(5):647–663, 2012.

[HLP15] Lionel Heng, Gim Hee Lee, and Marc Pollefeys. Self-calibration and visual slam with a multi-camera system on a micro aerial vehicle. *Autonomous Robots*, 39(3):259–277, 2015.

[HTWM04] Weiming Hu, Tieniu Tan, Liang Wang, and Steve Maybank. A survey on visual surveillance of object motion and behaviors. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 34(3):334–352, 2004.

[HWB+18] Brent Hecht, Lauren Wilcox, Jeffrey P. Bigham, Johannes Schöning, Ehsan Hoque, Jason Ernst, Yonatan Bisk, Luigi De Russis, Lana Yarosh, Bushra Anjum, Danish Contractor, and Cathy Wu. It's Time to Do Something: Mitigating the Negative Impacts of Computing Through a Change to the Peer Review Process. Technical report, ACM Future of Computing Blog, Association for Computing Machinery (ACM), 2018. `https://acm-fca.org/2018/03/29/negativeimpacts/`.

[HZ03] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2003.

[Inc17] Lytro Inc. Lytro Reality Experience & Lytro Immerge Updates. `http://blog.lytro.com/lytro-reality-experience-lytro-immerge-updates/`, 2017.

[JLT+15a] Hanbyul Joo, Hao Liu, Lei Tan, Lin Gui, Bart Nabbe, Iain Matthews, Takeo Kanade, Shohei Nobuhara, and Yaser Sheikh. Panoptic studio: A massively multiview system for social motion capture. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3334–3342, 2015.

[JLT$^+$15b] Hanbyul Joo, Hao Liu, Lei Tan, Lin Gui, Bart Nabbe, Iain Matthews, Takeo Kanade, Shohei Nobuhara, and Yaser Sheikh. Panoptic studio: A massively multiview system for social motion capture. In *IEEE International Conference on Computer Vision (ICCV)*, 2015.

[JWK09] Fan Jiang, Ying Wu, and Aggelos K Katsaggelos. Detecting contextual anomalies of crowd motion in surveillance video. In *Image Processing (ICIP), 2009 16th IEEE International Conference on*, pages 1117–1120. IEEE, 2009.

[KA14] Francisco R Klauser and Anders Albrechtslund. From self-tracking to smart urban infrastructures: Towards an interdisciplinary research agenda on big data. *Surveillance & Society*, 12(2):273, 2014.

[KDBO$^+$05] Niklas Karlsson, Enrico Di Bernardo, Jim Ostrowski, Luis Goncalves, Paolo Pirjanian, and Mario E Munich. The vSLAM algorithm for robust localization and mapping. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 24–29. IEEE, 2005.

[KHB08] Juho Kannala, Janne Heikkilä, and Sami S Brandt. Geometric camera calibration. *Wiley Encyclopedia of Computer Science and Engineering*, 2008.

[KRN97] Takeo Kanade, Peter Rander, and PJ Narayanan. Virtualized reality: Constructing virtual worlds from real scenes. *IEEE Multimedia*, 4(1):34–47, 1997.

[KSC15] Christian Kerl, Jorg Stuckler, and Daniel Cremers. Dense continuous-time tracking and mapping with rolling shutter rgb-d cameras. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2264–2272, 2015.

[KSM$^+$07] Akira Kubota, Aljoscha Smolic, Marcus Magnor, Masayuki Tanimoto, Tsuhan Chen, and Cha Zhang. Multiview imaging and 3DTV. *IEEE Signal Processing Magazine*, 24(6):10–21, 2007.

[LFP13] Gim Hee Lee, Friedrich Faundorfer, and Marc Pollefeys. Motion estimation for self-driving cars with a generalized camera. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 2746–2753. IEEE, 2013.

[LG09] Andrew Lumsdaine and Todor Georgiev. The focused plenoptic camera. In *IEEE International Conference on Computational Photography (ICCP)*, pages 1–8. IEEE, 2009.

[LH96] Marc Levoy and Pat Hanrahan. Light field rendering. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, pages 31–42. ACM, 1996.

[LHKP13] Bo Li, Lionel Heng, Kevin Koser, and Marc Pollefeys. A multiple-camera system calibration toolbox using a feature descriptor-based calibration pattern. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1301–1307. IEEE, 2013.

[LHVS14] Richard Latimer, Jason Holloway, Ashok Veeraraghavan, and Ashutosh Sabharwal. Socialsync: Sub-frame synchronization in a smartphone camera network. In *European Conference on Computer Vision*, pages 561–575. Springer, 2014.

[LLZC14] Xinzhao Li, Yuehu Liu, Shaozhuo Zhai, and Zhichao Cui. A structural constraint based dual camera model. In *Chinese Conference on Pattern Recognition*, pages 293–304. Springer, 2014.

[LM13] Cheng Lu and Mrinal Mandal. A robust technique for motion-based video sequences temporal alignment. *IEEE Transactions on Multimedia*, 15(1):70–82, 2013.

[LMJH$^+$11] Jorge Lopez-Moreno, Jorge Jimenez, Sunil Hadap, Ken Anjyo, Erik Reinhard, and Diego Gutierrez. Non-photorealistic, depth-based image editing. *Computers & Graphics*, 35(1):99–111, 2011.

[Low99] David G Lowe. Object recognition from local scale-invariant features. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, volume 2, pages 1150–1157. IEEE, 1999.

[LS12] Yuankun Liu and Xianyu Su. Camera calibration with planar crossed fringe patterns. *Optik-International Journal for Light and Electron Optics*, 123(2):171–175, 2012.

[LSFW14] Junbin Liu, Sridha Sridharan, Clinton Fookes, and Tim Wark. Optimal camera planning under versatile user constraints in multi-camera image processing systems. *IEEE Transactions on Image Processing*, 23(1):171–184, 2014.

[LY06] Cheng Lei and Yee-Hong Yang. Tri-focal tensor-based multiple video synchronization with subframe optimization. *IEEE Transactions on Image Processing*, 15(9):2473–2480, 2006.

[LZT06] Georgios Litos, Xenophon Zabulis, and Georgios Triantafyllidis. Synchronous image acquisition based on network synchronization. In *2006 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'06)*, pages 167–167. IEEE, 2006.

[Mac06] Chris A Mack. *Field Guide to Optical Lithography*, volume 6. SPIE Press Bellingham, WA, 2006.

[Mat17] MathWorks. Matlab | camera calibration. `https://se.mathworks.com/help/vision/camera-calibration.html`, 2017.

[MBM16] Matteo Munaro, Filippo Basso, and Emanuele Menegatti. Openptrack: Open source multi-camera calibration and people tracking for rgb-d camera networks. *Robotics and Autonomous Systems (RAS)*, 75:525–538, 2016.

[MHT11] Mehdi Moussaïd, Dirk Helbing, and Guy Theraulaz. How simple rules determine pedestrian behavior and crowd disasters. *Proceedings of the National Academy of Sciences*, 108(17):6884–6888, 2011.

[Mic18] Microsoft. Kinect for windows, 2018. `https://developer.microsoft.com/en-us/windows/kinect`, Retrieved 06/04/2018.

[MP04] Wojciech Matusik and Hanspeter Pfister. 3D TV: a scalable system for real-time acquisition, transmission, and autostereoscopic display of dynamic scenes. In *ACM Transactions on Graphics (TOG)*, volume 23, pages 814–824. ACM, 2004.

[Mö18] Stefan Möllenhoff. Beautiful blur for smartphone portraits: How bokeh effect works. `https://www.androidpit.com/how-bokeh-effect-works-with-smartphones`, 2018.

[NK07] Mami Noguchi and Takekazu Kato. Geometric and timing calibration for unsynchronized cameras using trajectories of a moving marker. In *IEEE Workshop on Applications of Computer Vision (WACV)*, pages 20–20. IEEE, 2007.

[NLB$^+$05] Ren Ng, Marc Levoy, Mathieu Brédif, Gene Duval, Mark Horowitz, and Pat Hanrahan. Light field photography with a hand-held plenoptic camera. *Computer Science Technical Report CSTR*, 2(11):1–11, 2005.

[NRL$^+$13] Rahul Nair, Kai Ruhl, Frank Lenzen, Stephan Meister, Henrik Schäfer, Christoph S Garbe, Martin Eisemann, Marcus Magnor, and Daniel Kondermann. A survey on time-of-flight stereo fusion. In *Time-of-Flight and Depth Imaging. Sensors, Algorithms, and Applications*, pages 105–127. Springer, 2013.

[NS09] Michael Nischt and Rahul Swaminathan. Self-calibration of asynchronized camera networks. In *IEEE 12th International Conference on Computer Vision Workshops (ICCV Workshops)*, pages 2164–2171. IEEE, 2009.

[OCK$^+$13] Ferda Ofli, Rizwan Chaudhry, Gregorij Kurillo, René Vidal, and Ruzena Bajcsy. Berkeley MHAD: A comprehensive multimodal human action database. In *IEEE Workshop on Applications of Computer Vision (WACV)*, pages 53–60. IEEE, 2013.

[OLS$^+$15] Shun-Hsing Ou, Chia-Han Lee, V Srinivasa Somayazulu, Yen-Kuang Chen, and Shao-Yi Chien. On-line multi-view video summarization for wireless video sensor network. *IEEE Journal of Selected Topics in Signal Processing*, 9(1):165–179, 2015.

[Pan17] Panocam3d.com. 3D 360 Cameras. `http://www.panocam3d.com/camera.html`, 2017.

[PCSK10] Flavio Padua, Rodrigo Carceroni, Geraldo Santos, and Kiriakos Kutulakos. Linear sequence-to-sequence alignment. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(2):304–320, 2010.

[Ple03] Robert Pless. Using many cameras as one. In *Proceedings of the 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages II–587. IEEE, 2003.

[PM10] Dmitry Pundik and Yael Moses. Video synchronization using temporal signals from epipolar lines. In *European Conference on Computer Vision*, pages 15–28. Springer, 2010.

[RK12] Taufiqur Rahman and Nicholas Krouglicof. An efficient camera calibration technique offering robustness and accuracy over a wide range of lens distortion. *IEEE Transactions on Image Processing*, 21(2):626–637, 2012.

[RKLM12] Kai Ruhl, Felix Klose, Christian Lipski, and Marcus Magnor. Integrating approximate depth data into dense image correspondence estimation. In *Proceedings of the 9th European Conference on Visual Media Production*, pages 26–31. ACM, 2012.

[RN96] Byron Reeves and Clifford Ivar Nass. *The media equation: How people treat computers, television, and new media like real people and places.* Cambridge university press, 1996.

[RRKB11] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. Orb: An efficient alternative to sift or surf. In *IEEE International Conference on Computer Vision (ICCV)*, pages 2564–2571. IEEE, 2011.

[RV14] Dikpal Reddy and Ashok Veeraraghavan. Lens flare and lens glare. In *Computer Vision*, pages 445–447. Springer, 2014.

[SAB$^+$07] Elena Stoykova, A Ayd, Philip Benzie, Nikos Grammalidis, Sotiris Malassiotis, Joern Ostermann, Sergej Piekh, Ventseslav Sainov, Christian Theobalt, Thangavel Thevar, et al. 3-D time-varying scene capture technologies — a survey. *IEEE Transactions on Circuits and Systems for Video Technology*, 17(11):1568–1586, 2007.

[SBND10] Andrew D Straw, Kristin Branson, Titus R Neumann, and Michael H Dickinson. Multi-camera real-time three-dimensional tracking of multiple flying animals. *Journal of The Royal Society Interface*, page rsif20100230, 2010.

[SBW07] Prarthana Shrstha, Mauro Barbieri, and Hans Weda. Synchronization of multi-camera video recordings based on audio. In *Proceedings of the 15th ACM International Conference on Multimedia*, pages 545–548. ACM, 2007.

[SD96] Steven M Seitz and Charles R Dyer. View morphing. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, pages 21–30. ACM, 1996.

[SFHT16] Chris Sweeney, Victor Fragoso, Tobias Hollerer, and Matthew Turk. Large scale SfM with the distributed camera model. *arXiv preprint arXiv:1607.03949*, 2016.

[SKKS14] Tamara Seybold, Marion Knopp, Christian Keimel, and Walter Stechele. Beyond standard noise models: Evaluating denoising algorithms with respect to realistic camera noise. *International Journal of Semantic Computing*, 8(02):145–167, 2014.

[SLK15] Hamed Sarbolandi, Damien Lefloch, and Andreas Kolb. Kinect range sensing: Structured-light versus time-of-flight kinect. *Computer Vision and Image Understanding (CVIU)*, 139:1–20, 2015.

[SMP05] Tomáš Svoboda, Daniel Martinec, and Tomáš Pajdla. A convenient multicamera self-calibration for virtual environments. *PRESENCE: Teleoperators and Virtual Environments*, 14(4):407–422, 2005.

[SSE$^+$13] Florian Schweiger, Georg Schroth, Michael Eichhorn, Anas Al-Nuaimi, Burak Cizmeci, Michael Fahrmair, and Eckehard Steinbach. Fully automatic and frame-accurate video synchronization using bitrate sequences. *IEEE Transactions on Multimedia (TMM)*, 15(1):1–14, 2013.

[SSL13] Hooman Shidanshidi, Farzad Safaei, and Wanqing Li. A method for calculating the minimum number of cameras in a light field based free viewpoint video system. In *2013 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6. IEEE, 2013.

[SSO14] Sebastian Schwarz, Mårten Sjöström, and Roger Olsson. Multivariate sensitivity analysis of time-of-flight sensor fusion. *3D Research*, 5(3):18, 2014.

[SSS06] Noah Snavely, Steven M Seitz, and Richard Szeliski. Photo tourism: Exploring photo collections in 3d. In *ACM Transactions on Graphics (TOG)*, volume 25, pages 835–846. ACM, 2006.

[SWBS06] Prarthana Shrestha, Hans Weda, Mauro Barbieri, and Dragan Sekulovski. Synchronization of multiple video recordings based on still camera flashes. In *Proceedings of the 14th ACM International Conference on Multimedia*, pages 137–140. ACM, 2006.

[SWR96] Janet E Stockdale, Andrew J Wells, and Marilyn Rall. Participation in free-time activities: a comparison of london and new york. *Leisure Studies*, 15(1):1–16, 1996.

[TAHL07] Eino-Ville Talvala, Andrew Adams, Mark Horowitz, and Marc Levoy. Veiling glare in high dynamic range imaging. In *ACM Transactions on Graphics (TOG)*, volume 26, page 37. ACM, 2007.

[TFJ+14] Diane H Theriault, Nathan W Fuller, Brandon E Jackson, Evan Bluhm, Dennis Evangelista, Zheng Wu, Margrit Betke, and Tyson L Hedrick. A protocol and calibration method for accurate multi-camera field videography. *Journal of Experimental Biology*, pages jeb–100529, 2014.

[Thi98] J-P Thirion. Image matching as a diffusion process: an analogy with maxwell's demons. *Medical image analysis*, 2(3):243–260, 1998.

[tL17] Humaneyes technologies LTD. Vuze Plus Camera. `https://vuze.camera/camera/vuze-plus-camera/`, 2017.

[TTN08] Yuichi Taguchi, Keita Takahashi, and Takeshi Naemura. Real-time all-in-focus video-based rendering using a network camera array. In *3DTV Conference: The True Vision-Capture, Transmission and Display of 3D Video (3DTV-Con)*, pages 241–244. IEEE, 2008.

[TVG04] Tinne Tuytelaars and Luc Van Gool. Synchronizing video sequences. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages I–I. IEEE, 2004.

[VDS+15] Suren Vagharshakyan, Ahmed Durmush, Olli Suominen, Robert Bregovic, and Atanas Gotchev. Accuracy evaluation of a linear positioning system for light field capture. In *Asian Conference on Intelligent Information and Database Systems*, pages 388–397. Springer, 2015.

[WSLH01] Bennett S Wilburn, Michal Smulski, Hsiao-Heng Kelin Lee, and Mark A Horowitz. Light field video camera. In *Media Processors 2002*, volume 4674, pages 29–37. International Society for Optics and Photonics, 2001.

[Wu13] Changchang Wu. Towards linear-time incremental structure from motion. In *3DTV Conference: The True Vision-Capture, Transmission and Display of 3D Video (3DTV-Con)*, pages 127–134. IEEE, 2013.

[WWDG13] Huogen Wang, Jiachen Wang, Zhiyong Ding, and Fei Guo. Self-converging camera arrays: Models and realization. In *2013 Ninth International Conference on Natural Computation (ICNC)*, pages 338–342. IEEE, 2013.

[YEBM02] Jason C Yang, Matthew Everett, Chris Buehler, and Leonard McMillan. A real-time distributed light field camera. *Rendering Techniques*, 2002:77–86, 2002.

[Yes06] Bilge Yesil. Watching ourselves: Video surveillance, urban space and self-responsibilization. *Cultural Studies*, 20(4-5):400–416, 2006.

[YTJ+14] Su Jeong You, Phuc H Truong, Sang Hoon Ji, Sang Moo Lee, Chang Eun Lee, and Young Jo Cho. A cooperative multi-camera system for tracking a fast moving object. In *4th Annual International Conference on Cyber Technology in Automation, Control, and Intelligent Systems (CYBER)*, pages 141–145. IEEE, 2014.

[ZEM+15] Matthias Ziegler, Andreas Engelhardt, Stefan Müller, Joachim Keinert, Frederik Zilly, Siegfried Foessel, and Katja Schmid. Multi-camera system for depth based visual effects and compositing. In *Proceedings of the 12th European Conference on Visual Media Production*, page 3. ACM, 2015.

[Zha00] Zhengyou Zhang. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11):1330–1334, 2000.

[ZMDM+16] Pietro Zanuttigh, Giulio Marin, Carlo Dal Mutto, Fabio Dominio, Ludovico Minto, and Guido Maria Cortelazzo. Data fusion from depth and standard cameras. In *Time-of-Flight and Structured Light Depth Cameras*, pages 161–196. Springer, 2016.

[Zon12] Ray Zone. *3-D Revolution: The History of Modern Stereoscopic Cinema*. University Press of Kentucky, 2012.