Setting up a Computational Photography Research Studio

by

Obumneme Stanley Dukor

B.Sc., University of Lagos, 2019

Thesis Submitted in Partial Fulfillment of the
Requirements for the Degree of
Master of Science

in the
School of Computing Science
Faculty of Applied Sciences

SIMON FRASER UNIVERSITY
Summer 2024

Declaration of Committee

Name:	**Obumneme Stanley Dukor**
Degree:	**Master of Science**
Thesis title:	**Setting up a Computational Photography Research Studio**
Committee:	**Chair:** Wuyang Chen Assistant Professor, Computing Science
	Yağız Aksoy Supervisor Assistant Professor, Computing Science
	Mo Chen Committee Member Assistant Professor, Computing Science
	Ali Mahdavi-Amiri Examiner Assistant Professor, Computing Science

Abstract

AI research is transforming creative tasks, with advancements in AI tools rapidly changing post-production expectations. However, the development of these technologies is mostly driven by technologists, often without involving the creatives who will use them. This thesis presents the development of a Computational Photography Research Studio aimed at bridging this gap. The goal is to create a practical and flexible studio setup that allows collaboration between creatives and researchers, allowing production and research to occur simultaneously. This new type of research involves stakeholders, like filmmakers, to ensure the research addresses their needs and benefits creative professionals. The studio setup includes portable production cameras and lighting, enabling the capture of high-quality live-action footage and datasets necessary for developing computational photography algorithms for post-production. This environment aims to direct AI research to better serve the filmmaking community, ultimately enhancing the quality of visual storytelling.

Keywords: Computer Vision; Computational Photography; Machine Learning; Independent Filmmaking

Dedication

To my family and friends for their endless support.

Acknowledgements

I would like to express my deepest gratitude to my advisor, Professor Yağız Aksoy, for his guidance, support, and encouragement throughout my Master's program. His expertise and mentorship have been invaluable in shaping my academic journey.

I also want to extend my gratitude to Sebastian Dille and Mahdi Miangoleh, whose help and support were crucial in all the projects I worked on. A special thanks to Mahesh Reddy and Chris Carega for always being available whenever I needed help.

This research studio was funded by Canada Foundation for Innovation and B.C. Knowledge Development Fund (BCKDF) under the project "Computational Cinematography".

Table of Contents

Declaration of Committee ii

Abstract iii

Dedication iv

Acknowledgements v

Table of Contents vi

List of Tables ix

List of Figures x

1 Introduction 1

1.1 Topics in this Thesis . . . 3

1.1.1 Studio Setup . . . 3

1.1.2 Capture and Data Processing . . . 4

1.1.3 Green Screen Keying and Image Matting . . . 4

1.2 Summary . . . 4

2 Studio Setup 5

2.1 Green Screen . . . 7

2.2 Lighting . . . 8

2.2.1 Color Temperature . . . 9

2.3 Cameras and Lenses . . . 10

2.3.1 Image Formation . . . 10

2.3.2 Lenses . . . 12

2.3.3 Types of Cameras Used in Production . . . 15

2.3.4 Types of Lenses Used in Production . . . 16

2.3.5 Camera Placement and Configuration . . . 17

2.4 The Exposure Triangle . . . 18

2.4.1 ISO . . . 19

2.4.2 Shutter Speed . . . 19
2.4.3 Aperture . . . 19
2.4.4 Balancing the Exposure Triangle . . . 20
2.5 White Balance . . . 21
2.6 Microphone . . . 22
2.6.1 Types of Microphones . . . 22
2.6.2 Relevant Microphone Concepts . . . 23
2.6.3 Microphone Setup in Our Studio . . . 23
2.6.4 Microphone Placement . . . 24
2.6.5 Microphone Audio Configuration . . . 24
2.7 Timecode . . . 24
2.7.1 Types of Timecode . . . 25
2.7.2 Free Run Timecode vs. Record Run Timecode . . . 25
2.7.3 Timecode Frame Rate . . . 26
2.7.4 Drop Frame and Non-Drop Frame Timecode . . . 26
2.7.5 Timecode Configuration . . . 27
2.8 Storage . . . 27
2.8.1 Camera Video Settings . . . 29

3 Data Capture and Processing **30**
3.1 Data Capture . . . 30
3.1.1 Synchronization Process . . . 31
3.1.2 Calibration Process . . . 31
3.2 Data Transfer . . . 32
3.3 Time Synchronization . . . 33
3.3.1 Phase Shift in Cameras . . . 35
3.3.2 Mechanisms of Shutter Operation in Video Recording . . . 35
3.3.3 Camera Internal Clock . . . 37
3.3.4 Managing Phase Shifts . . . 37
3.4 Data Encoding . . . 38
3.5 Camera Calibration . . . 41
3.5.1 Camera Matrix . . . 42
3.5.2 Forward Imaging Model: From 3D to 2D . . . 43
3.5.3 Forward Imaging Model: From 3D to 3D . . . 44
3.5.4 Forward Imaging Model: Combining Intrinsics and Extrinsics . . . 46
3.6 Studio Multi-camera Calibration . . . 48
3.6.1 Calibration Using Pattern of Known Geometry . . . 48
3.6.2 Checkerboard Points Detection . . . 49
3.6.3 Single Camera Calibration . . . 51

3.6.4 Multi-Camera Calibration . . . 52
3.6.5 Analysis . . . 53

4 Automatic Green Screen Keying **56**
4.1 Related Work . . . 57
4.2 Method . . . 58
4.2.1 Trimap Generation . . . 59
4.2.2 Localized Background Matting . . . 62
4.2.3 Temporal Consistency . . . 66
4.3 Implementation . . . 66
4.3.1 Training Data . . . 67
4.3.2 Network Architectures and Training . . . 69
4.4 Experiments . . . 70
4.4.1 Detail Reconstruction . . . 70
4.4.2 Temporal Consistency . . . 71
4.4.3 Comparison Against Commercial Keying Solutions . . . 71
4.4.4 Effect of Intense Green Spill on Predicted Alpha . . . 73
4.4.5 Inference Resolution . . . 74

5 Conclusion and Future Work **75**
5.0.1 Key Considerations in Studio Setup . . . 76
5.0.2 Challenges and Limitations . . . 76
5.0.3 Future Work . . . 77

Bibliography **78**

Appendix A Code **83**

List of Tables

Table 3.1 Comparison of Detected Patterns and Reprojection Errors under Various Configurations 55

List of Figures

Figure 1.1 We present a comprehensive framework for a Computational Photography Research Studio 1

Figure 2.1 Recording session inside our studio. 5
Figure 2.2 Top-down layout of the studio room. 6
Figure 2.3 Shades of green for Backdrop Paper Green Screen. 7
Figure 2.4 Lighting setup for the green screen. 8
Figure 2.5 Various colors of a blackbody at different absolute temperatures and their correspondence light sources [49]. 9
Figure 2.6 Color temperature spectrum from 1700K to 10000K captured with Nanlite Forza 60c. 10
Figure 2.7 Bare-sensor imaging setup, where light from all scene points contributes to all sensor pixels, resulting in a blurred image [13]. 10
Figure 2.8 Pinhole camera model, where a diaphragm with a pinhole restricts light rays, allowing for a clear image formation on the sensor. The resulting image is inverted and scaled [13]. 11
Figure 2.9 Comparison of pinhole sizes. 11
Figure 2.10 Focal length impact 12
Figure 2.11 Illustration of camera lens's field of view (FOV) [43]. 13
Figure 2.12 Mirrorless Canon Camera. 15
Figure 2.13 Focal length selection for varied scenarios: 100mm close-up, 50mm presentation, and 15mm wide-angle interview. 16
Figure 2.14 Prime lenses capture stunning portraits while zoom lenses offer versatile framing in this well-equipped recording session 18
Figure 2.15 Exposure triangle, consisting of aperture, ISO and shutter speed. . 19
Figure 2.16 Illustration of how ISO, aperture, and shutter speed adjustments affect image exposure 20
Figure 2.17 Test images illuminated by different light sources without white balance [67]. 21
Figure 2.18 Audio setup: Rode mics on main cameras, Wireless GO II clipped to speaker. 24
Figure 2.19 Example of timecode labeling 25

Figure 2.20 (a) A compact timecode generator for setting the timecode of each camera, and (b) a set of SD cards used for storing the recordings from each camera in our studio. 27
Figure 2.21 Chart showing the possible colors in 8-bit, 10-bit and 12-bit video [60]. 28

Figure 3.1 Step-by-Step Capture Procedure Pipeline for Each Recording Session 30
Figure 3.2 Checkerboard calibration patterns used in the studio 31
Figure 3.3 Clap Board, for marking frames using the audio waveform during recording. 32
Figure 3.4 Example of a regular recording where the phase shift is not perceptible due to the absence of fast-paced motion. 34
Figure 3.5 Illustration of a fast-paced event using a clapboard, highlighting the visible phase shift difference across camera frames. 35
Figure 3.6 Waveforms from five synchronized cameras reveal a phase shift of up to 30ms, caused by variations in mechanical shutter timing. 36
Figure 3.7 Comparison of video quality and file size across different CRF values. As the CRF value approaches 51, compression artifacts become more noticeable. 40
Figure 3.8 Mapping world coordinates to image coordinates via extrinsic and intrinsic camera parameters. 41
Figure 3.9 Mapping from 3D world coordinates to camera coordinates to pixel coordinates [41]. 42
Figure 3.10 Basic perspective projection of a 3D world point onto a 2D image plane, illustrating the center of projection **O** [8]. 43
Figure 3.11 Transformation of a 3D point from world coordinates to camera coordinates 45
Figure 3.12 Example of sparse and imprecise feature matching in our studio. The lack of distinct features results in sparse matches. 47
Figure 3.13 Some types of well-defined calibration patterns with known geometry. 48
Figure 3.14 Detected checkerboard points at the high-resolution images. 50
Figure 3.15 Challenges in fitting medium or large patterns within a narrow field of view. 54

Figure 4.1 We achieve high resolution green screen keying with our method compared to other background matting methods. 56
Figure 4.2 Our pipeline: optional trimap prediction, followed by background inpainting and image matting, ending with consistent alpha and foreground prediction. Outputs are circled in green. 59
Figure 4.3 Trimap generation input examples 60
Figure 4.4 Trimap Error Visualization 61

Figure 4.5 Discrepancies in Clean Plate Background 62
Figure 4.6 We replace the closed-form optimization with a ConvNet architecture 63
Figure 4.7 Synthetically generated composite image for training after linearization and gamma-correction (gamma=2.2). 67
Figure 4.8 Generated trimaps for training: ground truth for supervision, dilated trimaps for inpainting and matting network training. 68
Figure 4.9 Illustration of patch selection strategy during training. 69
Figure 4.10 We show an example of an artificial frame pair with their pixel difference for the video matting training [30]. 70
Figure 4.11 We show two examples of the detail reconstruction for our high-resolution studio data. 71
Figure 4.12 We show our results on our captured studio data for 5 consecutive frames. 72
Figure 4.13 We show the result of our method in comparison with a result obtained from Keylight. 72
Figure 4.14 Comparison of alpha mattes under intense green spill: our method preserves foreground integrity, outperforming FBA and RVM by avoiding spill-induced errors. 73
Figure 4.15 Comparison of alpha mattes generated at different inference resolutions 74

Figure 5.1 We are able to perform research and production simultaneously, from paper presentations and interviews to multi-illumination experiments and shadow analysis. 75
Figure 5.2 Illustration of the drift in accuracy observed in the video matting network starting from the 50th frame. 76

Chapter 1

Introduction

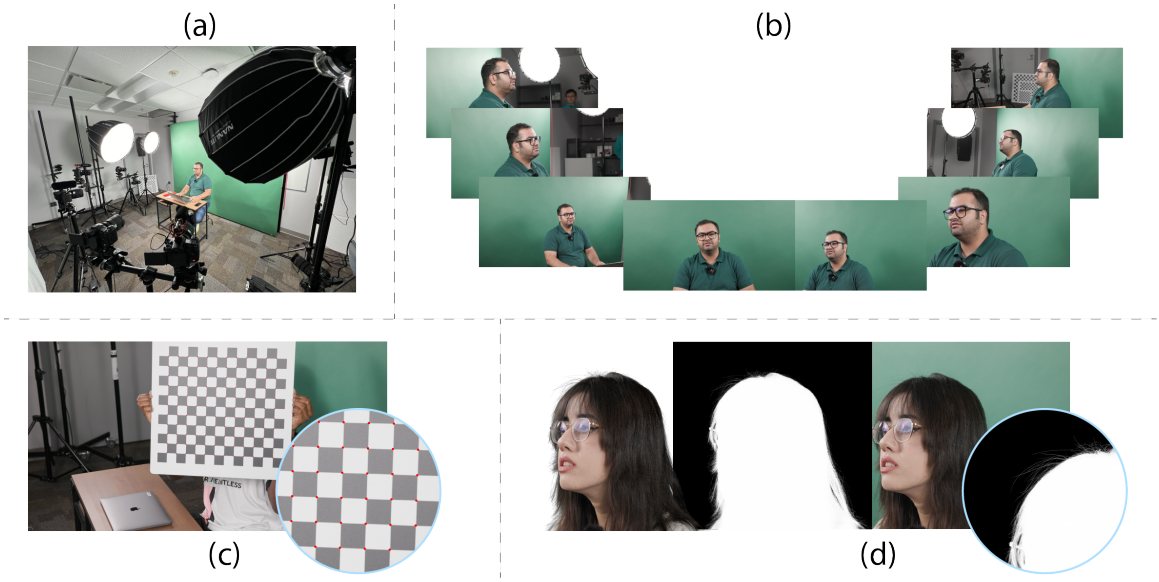


Figure 1.1: We present a comprehensive framework for a Computational Photography Research Studio, designed to integrate AI-driven methodologies with creative filmmaking. The pipeline includes four key stages: (a) Studio Setup and Data Capture, utilizing production cameras and lighting for independent environments; (b) Multicamera Synchronization, ensuring precise temporal alignment across cameras; (c) Camera Calibration, for obtaining accurate 3D representation of our studio; and (d) Green Screen Keying, to isolate the subject for both post-production workflows and computatinoal photograph research. Each stage is engineered for flexibility and ease of use in production settings.

The rapid advancements in Artificial Intelligence (AI) research are transforming creative tasks and revolutionizing the landscape of post-production. The expectations and capabilities in this field are rapidly evolving due to the development of new AI tools. However, the development of these technologies has predominantly been driven by technical experts, often excluding the creative professionals who are the end-users of these tools. This disconnect

can lead to a misalignment between the capabilities of AI tools and the actual needs of creative professionals.

To address this gap, the goal of this thesis is to establish a practical and flexible studio setup that allows both production and research to be conducted simultaneously, fostering a collaborative environment between creatives and researchers. This studio setup is designed to support a new type of research that involves stakeholders—such as filmmakers—who will potentially benefit from the research outcomes. By integrating the input of these stakeholders, the research can be directed to address their specific needs, aligning AI advancements with the benefits of creative professionals.

Traditional setups for computational photography research often involve the use of computer vision cameras and controlled lighting environments. While effective, these setups are typically expensive and require extensive engineering efforts, making them impractical for active production environments. They also tend to be rigid and complex, limiting their usability for non-technical professionals. In contrast, this thesis proposes a flexible and portable studio setup that utilizes production cameras and familiar lighting equipment commonly found in independent production environments. This approach not only makes the setup more accessible and user-friendly for non-AI engineers but also ensures high-quality production and efficient data capture for computational photography research. The setup enables the capture of live-action footage suitable for high-quality production as well as datasets necessary for developing advanced computational photography algorithms.

The broader vision for the studio involves creating a comprehensive pipeline that extends beyond the immediate scope of this thesis. A crucial theme of this vision is illumination analysis and manipulation, which plays a critical role in both production and post-production processes. The way a scene is illuminated has a profound impact on its perception, influencing mood, atmosphere, and narrative. This studio setup aims to not only support the capture of different lighting conditions accurately but also provide tools for analyzing and manipulating these conditions in post-production. By using multiple cameras and varied lighting setups, we seek to develop methods for changing scene illumination in a physically consistent manner, allowing filmmakers greater creative control in post-production.

Another key theme is the analysis and manipulation of camera positioning and movement. The precise placement and motion of cameras are crucial elements in storytelling, defining what the audience sees and experiences. The goal is to develop neural representations of the scene that can accommodate changes in camera movements during post-production. This capability is particularly valuable in refining complex shots that might not be perfectly executed on set, such as tracking or dolly-zoom shots. By providing a mechanism to adjust these elements in post-production, filmmakers are offered more flexibility and precision in crafting their narratives.

These themes are not isolated but interconnected within the studio's overall pipeline. The accurate capture and representation of illumination and scene structure lay the foundation

for more advanced applications, such as 3D reconstruction and mixed-reality environments. By integrating these elements, the studio setup aspires to create a full 3D representation of scenes using a single camera, enhancing the scope and quality of both traditional filmmaking and emerging mediums like virtual and augmented reality.

1.1 Topics in this Thesis

This thesis contributes to the foundation of a practical Computational Photography Research Studio aimed at bridging the gap between technical research and creative production. We focus on three main aspects: studio setup, data capture and processing, and automatic green screen keying. Each part of the thesis addresses specific challenges and proposes solutions that enhance both production efficiency and research capabilities.

1.1.1 Studio Setup

In Chapter 2, we explore the process of setting up a versatile studio environment. The chapter begins with a detailed discussion on the selection and practical positioning of the appropriate green screen, cameras, lenses, lighting equipment, and audio devices, taking into account the trade-offs and considerations essential for both research and production purposes in our studio environment. We introduce fundamental concepts such as image formation, camera lens field of view (FOV), the exposure triangle, and white balance and how these concepts affect the quality of captured data.

A critical aspect of our setup is the choice of audio devices, which are carefully selected to ensure high-fidelity sound recording that matches the visual quality of the video. We discuss the criteria for selecting microphones and audio interfaces, including considerations such as frequency response, dynamic range, and placement within the studio to minimize noise and ensure clear audio capture.

We also discuss the importance of timecode in maintaining temporal coherence across multiple video recordings. We explore the various considerations involved in selecting an appropriate timecode generation method, including the accuracy required for synchronization, compatibility with different camera systems, and the potential challenges in a production environment. Given the need for a setup that is both compact and portable, as well as user-friendly for non-technical experts, we explore the available timecode generator device that prioritizes these aspects.

Furthermore, we explore storage solutions, particularly the use of SD cards for capturing and storing high-resolution video and audio data, and discuss configuration settings that optimize both performance and workflow efficiency.

1.1.2 Capture and Data Processing

Chapter 3 focuses on the procedures and methodologies we employ during data capture and processing. This chapter outlines our systematic approach to capturing high-quality data, starting with the synchronization of timecodes across all cameras. We detail the calibration process, where each camera is exposed to a calibration pattern from various orientations, ensuring precise camera calibration.

One bottleneck in our workflow is the transfer of large video files from the camera SD cards to the studio's storage server. We address this challenge by exploring various data transfer techniques and optimizing the process to minimize delays. Furthermore, this chapter discusses our approach to encoding video files, striking a balance between file size and quality. We provide a comprehensive analysis of camera parameters obtained through calibration and propose an optimal calibration process to achieve an efficient multi-camera calibration system with low reprojection error.

1.1.3 Green Screen Keying and Image Matting

In Chapter 4, we turn our attention to background matting techniques, with a focus on green screen keying and extending these techniques to video processing. This chapter begins with a review of current approaches to green screen keying, highlighting their limitations, especially regarding resolution and temporal coherence.

We then introduce our pipeline for achieving high-resolution green screen keying, which incorporates temporal supervision to maintain consistency across video frames. The chapter provides an in-depth discussion of the details of the implementation, including the algorithms and system design for efficient inference of our high-resolution studio data. We rigorously compare our approach to existing methods and commercial software, demonstrating significant improvements in both quality and temporal coherence.

1.2 Summary

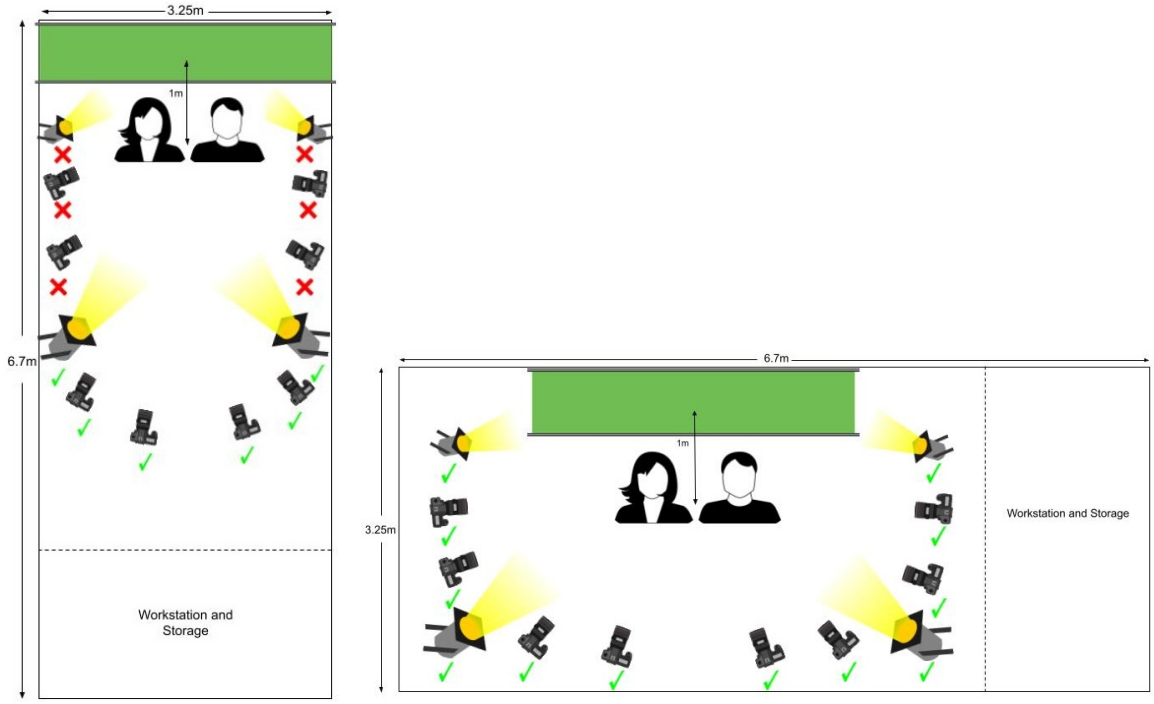
Together, these chapters establish a collaborative and adaptable studio environment that is not only technically robust but also creatively empowering. By integrating AI research with production techniques, we aim to create a studio setup that serves as a versatile tool for both researchers and filmmakers. This thesis offers a comprehensive guide to setting up such a studio, providing detailed insights into the challenges and solutions encountered.

Chapter 2

Studio Setup

Figure 2.1: Recording session inside our studio.

In order to efficiently set up the studio to achieve a practical and flexible environment that allows both production and research to be conducted simultaneously, various considerations and decisions must be carefully made. This chapter focuses on the choices and decisions made in order to set up the studio in a way that aligns with the objectives and goals of the research laboratory.



(a) Portrait (b) Landscape

Figure 2.2: Top-down layout of the studio room.

To set up the studio efficiently, we started by selecting an appropriate space, as shown in Figure 2.2. The room layout plays a role in accommodating both the recording equipment and non-recording equipment. Our setup requires space for 8 cameras and 4 light sources around a green screen and the subject being recorded. Additionally, we needed room for non-recording equipment, such as a workstation and storage for camera equipment.

Initially, we considered a portrait layout for the room as shown in Figure 2.2a. This arrangement offers significant flexibility in the distance between the main cameras and the subject, making it well-suited for production videos. However, this layout presents challenges when using multiple cameras, especially more than four, as it limits the ability to capture multiple views due to lack of available space, as seen in Figure 2.2a. This limitation is problematic for research purposes, where capturing different angles is essential for computer vision analysis, such as 3D reconstruction. Moreover, the portrait layout tends to become crowded when additional equipment, such as lighting sources, is added or when multiple subjects are involved in the recording.

On the other hand, the landscape layout as in Figure 2.2b, despite its limited distance range between the main cameras and the subject, offers sufficient space to accommodate multiple cameras. This layout is advantageous as it provides more room for both production video recording and research purposes. It allows for a more organized arrangement of

(a) Veri Green/Savage Tech Green (b) Jade/Savage Evergreen

Figure 2.3: Shades of green for Backdrop Paper Green Screen.

cameras, lighting, and other equipment. The landscape layout aligns well with the studio's goals of flexibility and versatility, making it an ideal choice for our setup.

2.1 Green Screen

The use of a green screen in our studio setup is crucial for both production and research purposes. They facilitate the process of chroma keying, where a specific color (in this case, green) is digitally removed from a scene, allowing for the background to be replaced with another image or video. This technique is valuable in post-production, enabling the smooth integration of different visual elements.

In our studio, we have made specific choices regarding the type and color of the green screen material. We opted for a 9' x 36' roll seamless paper green screen due to its smooth surface and uniform color, which minimizes the risk of unwanted reflections and inconsistencies in color. This uniformity is essential for achieving a clean keying effect, as any irregularities can complicate the keying process and reduce the quality of the final output.

Moreover, there has been a consideration regarding the shade of green to be used. Traditionally, a bright green color such as the one in Figure 2.3a is employed for chroma keying due to its high saturation and brightness, which provides a strong signal that is easily distinguishable from the foreground elements. However, in our setup, we have a unique context to consider. Our studio utilizes neural networks for background matting, a process that differs from standard chroma keying. This method allows for a more complex separation of the foreground and background, potentially mitigating the concerns associated with using a darker green such as the one in Figure 2.3b.

One of the key advantages of using a darker green is the reduction of color spill, which simplifies the problem for the neural network. Color spill, where the green color reflects onto the subject, can create halos or artifacts in the keyed image, complicating the removal

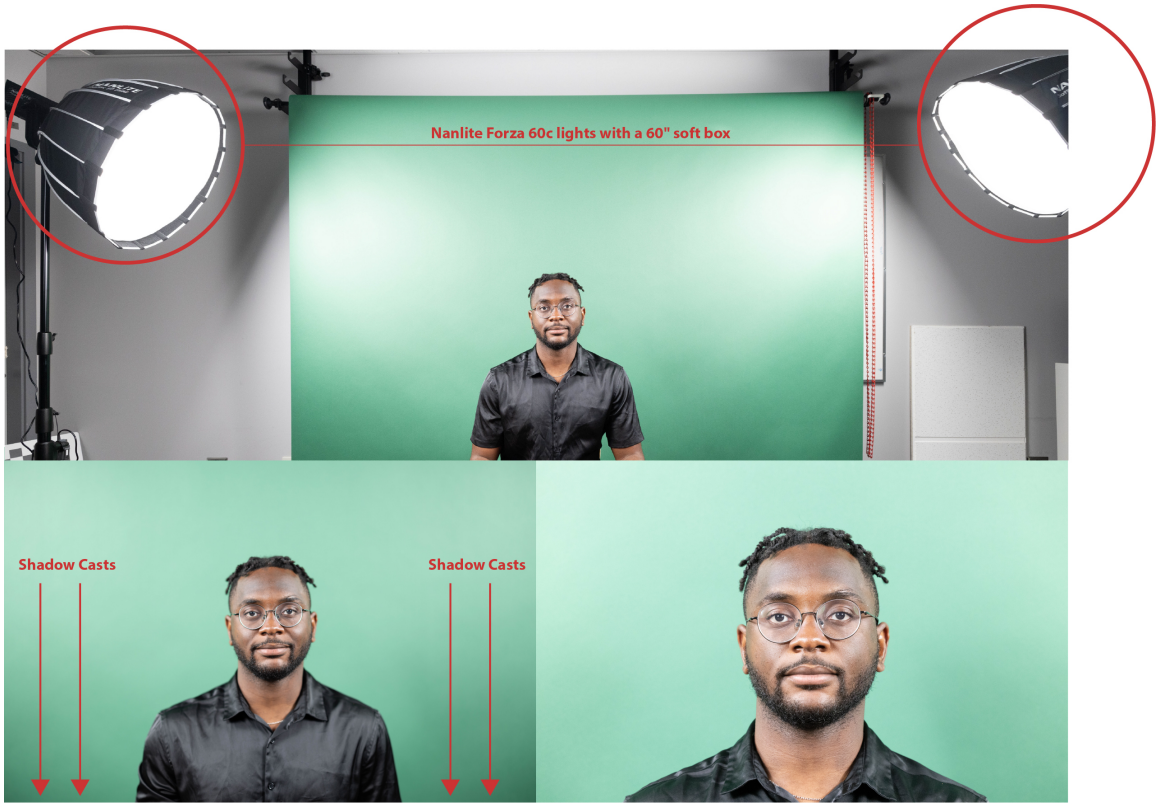


Figure 2.4: Lighting setup for the green screen. (Top) Positioning of Nanlite Forza 60c lights with 36" soft boxes for uniform illumination. (Bottom left) Shadows cast to the sides. (Bottom right) Main camera view showing a uniformly lit green background without shadows.

process and degrading the final visual quality. We perform an analysis on this specific scenario later in section 4.4.4 when comparing our green screen keying method with others. By choosing a darker shade, we aim to minimize this issue, enhancing the clarity of the foreground subject. The effectiveness of this approach will be further validated through our experiments and analysis.

In an ideal setup, the subject should be positioned at least 2 meters (approximately 8 feet) away from the green screen to avoid shadows on the green screen that could interfere with the keying process. However, due to space constraints in our studio, we have positioned the subject 1 meter away from the green screen, as shown in Figure 2.2.

2.2 Lighting

Effective green screen lighting is helpful in achieving high-quality chroma key results and minimizing post-production challenges. We uniformly illuminate the green screen using two Nanlite Forza 60c lights, each equipped with a Nanlite Forza 36" softbox, positioned symmetrically on either side of the green screen. These lights are set at an equal distance

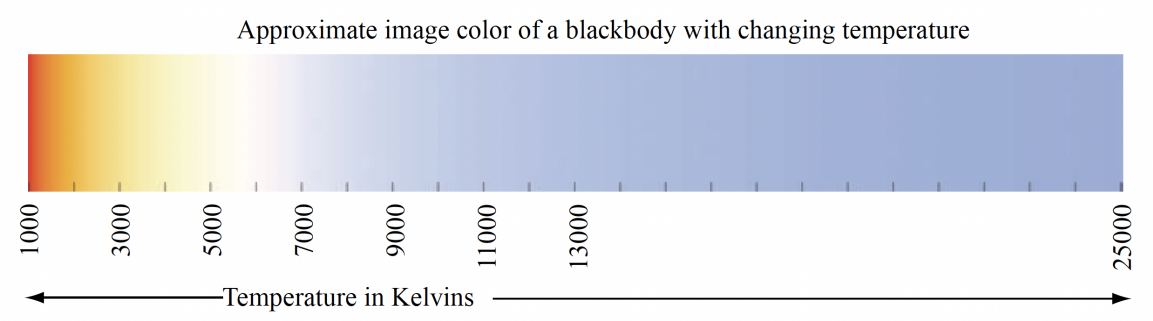


Figure 2.5: Various colors of a blackbody at different absolute temperatures and their correspondence light sources [49].

from the green screen and angled similarly to ensure even lighting across the entire surface. This setup helps eliminate shadows directly behind the subject, as any shadows cast will fall off to the sides, out of the main camera's view. The camera captures a uniformly lit green background, which is ideal for chroma keying later.

For lighting the subject, we employ two Nanlite Forza 60c lights, each fitted with a Nanlite Forza 60" softbox. These lights are versatile and can be adjusted based on the shooting scenario to function as either the key light or the fill light. The key light serves as the primary light source for the subject, simulating natural or artificial light sources such as the sun or interior lighting. This light is essential for defining the subject's features and should be positioned to match the intended light direction in the scene. The fill light is used to soften the shadows created by the key light, reducing the contrast between light and shadow to create a more balanced and natural look. Although we have not yet incorporated a backlight in our setup, it can be used to create a soft halo effect around the subject's hair, further separating them from the background and adding depth to the scene.

2.2.1 Color Temperature

The color temperature of a light source is defined as the temperature, measured in Kelvin, of a blackbody radiator that emits light with the same chromaticity (or nearly the same chromaticity) as the light source [49]. The color of a blackbody shifts from a reddish to a bluish hue as the temperature increases, as illustrated in Figure 3.2. This serves as a model for the varying shades of natural light that are observed in everyday life.

Since our studio is a controlled environment, for standard recording, we use a color temperature of *5600k*, representing daylight, and set all the light sources to the same color temperature, ensuring uniformity.

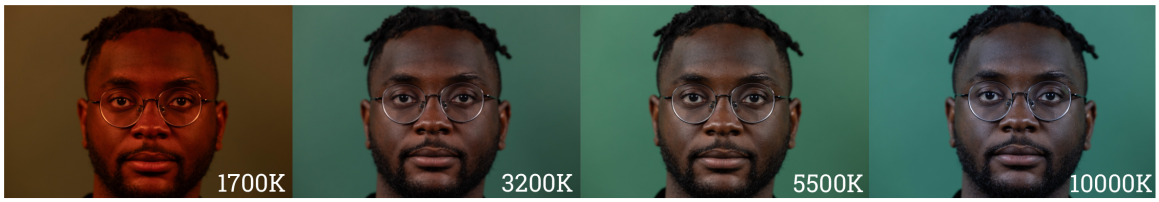


Figure 2.6: This figure displays the range of color temperature settings available on the Nanlite Forza 60c lights in our studio. All images were captured with a 200mm focal length lens and white balance of 5600k. The first image (left) shows a temperature of 1700K (candlelight), the second image represents 3200K (tungsten/incandescent), the third image represents 5500K (daylight), and the fourth image represents 10000K (clear blue sky).

2.3 Cameras and Lenses

Cameras and lenses are the most important components of our studio, serving as the primary tools for capturing motion pictures. Before going into the specifics of our camera positioning and how we utilize them to meet our studio needs, it is crucial to understand the fundamental process of image formation.

A camera, or more broadly, an imaging system, is a device that allows the projection of light from three-dimensional (3D) points in the real world onto a medium that records this light pattern. This medium could be traditional photographic film or a modern electronic sensor, which captures the intensity and sometimes the color of light to produce a digital image.

2.3.1 Image Formation

In the simplest case, we can consider a scenario known as *bare-sensor imaging*. Imagine a sensor placed in front of an object we wish to photograph, as shown in Figure 2.7. In this setup, all scene points contribute light to all sensor pixels, resulting in a completely blurred image. This happens because every location on the sensor receives light from multiple sources, making it impossible to distinguish specific features of the object.

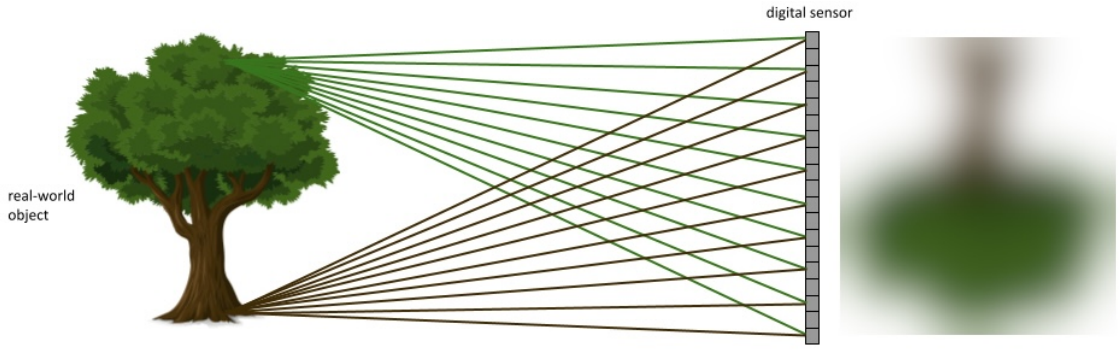


Figure 2.7: Bare-sensor imaging setup, where light from all scene points contributes to all sensor pixels, resulting in a blurred image [13].

To address this issue, a barrier with a small aperture, commonly referred to as a *pinhole*, can be introduced between the object and the sensor. This setup, as shown in Figure 2.8, allows only a single ray of light from each point in the scene to pass through the pinhole and reach a specific location on the sensor. As a result, each scene point contributes to only one sensor pixel, ensuring that the sensor captures a clear and sharp image of the object. The barrier ensures that the light reaching the sensor plane comes from only one direction per point, thereby preventing the overlap of light from different sources.

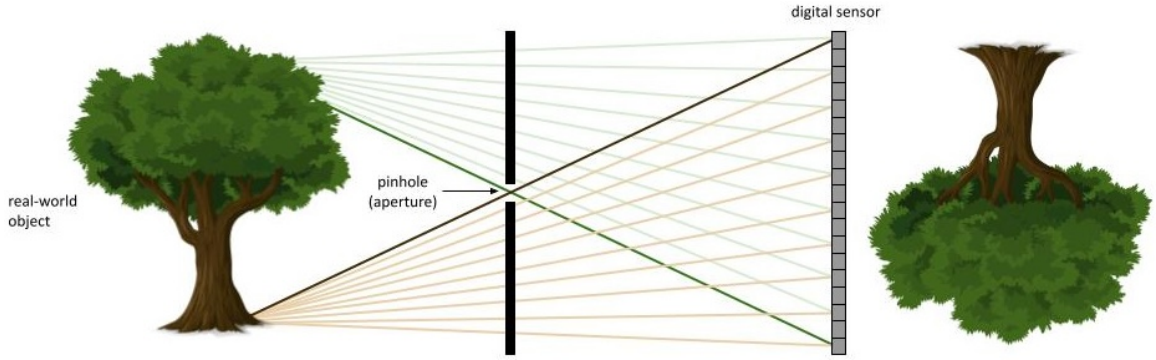


Figure 2.8: Pinhole camera model, where a diaphragm with a pinhole restricts light rays, allowing for a clear image formation on the sensor. The resulting image is inverted and scaled [13].

However, the sharpness of the image depends significantly on the size of the pinhole. If the pinhole is too large, multiple rays from a single point in the scene can pass through it and reach different locations on the sensor. Conversely, if the pinhole is too small, diffraction effects become significant, causing the image to lose sharpness. Thus, there is an optimal pinhole size that balances these two effects to produce the clearest image.

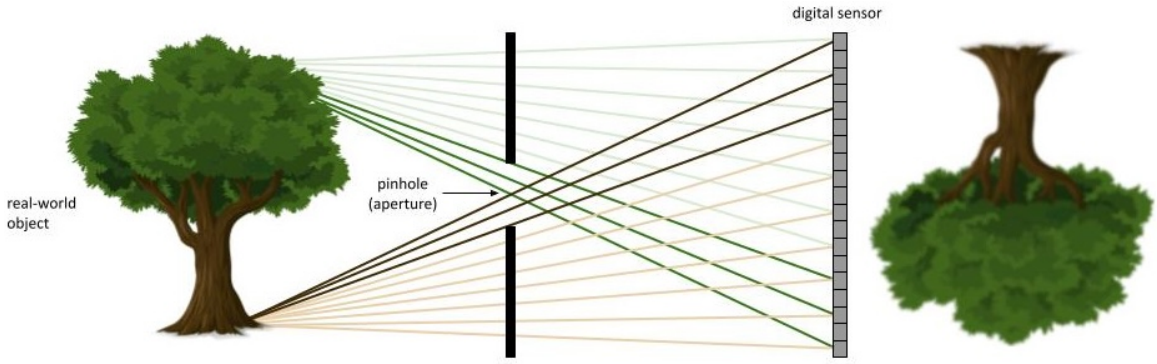


Figure 2.9: Illustration of pinhole size and its effect on image sharpness and brightness. A smaller pinhole results in a sharper image with less light, while a larger pinhole increases brightness but can cause blurriness [13].

In addition to the pinhole size, another critical parameter in image formation is the *focal length*. The focal length is the distance from the pinhole (or lens) to the sensor plane. It determines the size of the projected image on the sensor: a longer focal length results in a larger image, while a shorter focal length reduces the image size. This relationship occurs because a longer focal length allows the rays of light to spread out more before reaching the sensor, thereby enlarging the projection.

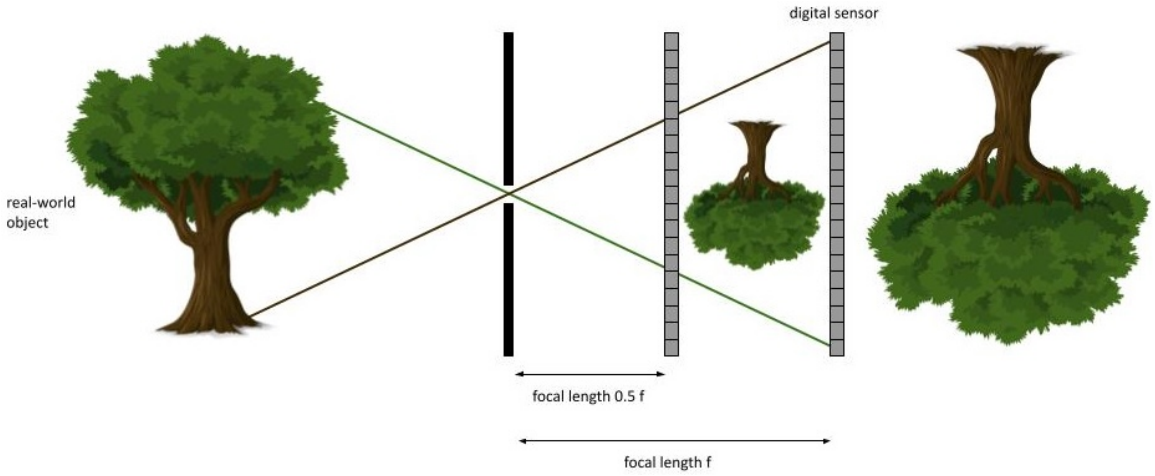


Figure 2.10: Illustration of focal length and its effect on image size. A longer focal length increases the size of the projected image on the sensor, while a shorter focal length decreases it [13].

This setup, known as the *pinhole camera model*, produces an image that is an inverted and scaled copy of the real-world object. The simplicity of this model makes it a foundational concept in understanding more complex imaging systems, including those that use lenses to further refine and enhance image quality.

2.3.2 Lenses

Lenses are crucial components in imaging systems that greatly enhance the quality and efficiency of light capture compared to the pinhole model. Unlike the simple pinhole, which restricts light to a single ray per point, lenses allow bundles of rays from each point in the scene to converge at corresponding points on the sensor. This convergence not only improves the brightness of the image but also sharpens the details, allowing for clearer and more detailed captures.

The primary advantage of using lenses is their ability to focus more light onto the sensor, making the imaging system more light-efficient. This efficiency is achieved because lenses gather and direct more photons from each scene point compared to a pinhole, which severely limits the amount of light that reaches the sensor. As a result, lenses enable the

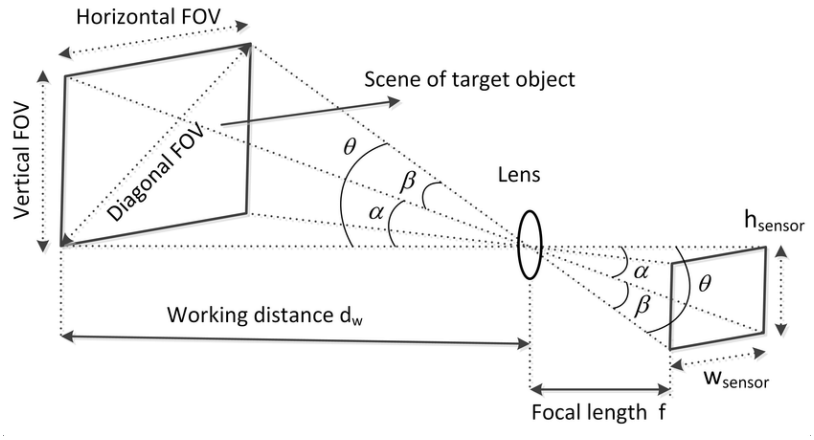


Figure 2.11: Illustration of camera lens's field of view (FOV) [43].

capture of well-exposed images even under low-light conditions, which is essential for various applications, including filmmaking and photography.

Field of View (FOV)

The field of view (FOV) of a camera lens determines the extent of the scene captured by the camera. The FOV is influenced by several factors, including the focal length of the lens and the size of the sensor.

From Figure 2.11, α and β are angles that represent the extent of the scene captured along the horizontal and vertical axes, respectively. θ is the total field of view, which can be thought of as the diagonal coverage of the lens. It is defined by the angle between the extreme rays passing through the center of the lens to the edges of the sensor. To understand how the FOV is related to the lens and sensor characteristics, we consider the following geometric relationship.

- h be the height of the sensor,
- f be the focal length of the lens,
- θ be the diagonal field of view.

Using the small angle approximation, the FOV can be derived as:

$$\tan\left(\frac{\theta}{2}\right) = \frac{h}{2f} \tag{2.1}$$

Solving for θ, we get:

$$\theta = 2\arctan\left(\frac{h}{2f}\right) \tag{2.2}$$

This equation shows that the FOV θ is determined by both the sensor size h and the focal length f of the lens. From this relationship, we can conclude that:

- Longer focal lengths (f) result in a narrower field of view. This is because a longer focal length reduces the angle θ, thus capturing a smaller portion of the scene. In Figure.., we show the effect of different focal lengths on the FOV for the same camera shot.
- Larger sensor sizes (h) result in a wider field of view. This is because a larger sensor allows for a greater angle θ, capturing more of the scene.

We can deduce that the choice of lens and sensor size directly affects the composition and framing of the captured image. This is crucial for achieving the desired visual outcome in both creative and technical photography applications.

Lens Focus

Focus determines the sharpness of the image captured. In simple terms, focusing adjusts the lens to ensure that light rays from a particular point in the scene converge at a corresponding point on the sensor.

Aside from the FOV of the image, the focal length also plays a significant role in determining the depth of field (DoF), which is the range of distances within a scene that appears acceptably sharp. A longer focal length results in a shallower depth of field, meaning that only a narrow plane of the scene will be in focus, while areas closer or further away from this plane will appear blurred. Conversely, shorter focal lengths provide a deeper depth of field, allowing more of the scene to be in focus.

The aperture of a lens, often denoted by the f-number (e.g., f/2.8, f/16), controls the amount of light entering the lens and also affects the depth of field. A larger aperture (smaller f-number) allows more light to pass through, resulting in a shallow depth of field. This effect is often used in portrait photography to create a blurred background while keeping the subject sharp. A smaller aperture (larger f-number) reduces the amount of light and increases the depth of field, making more of the scene appear in focus.

The relationship between the aperture and focal length can be described using the concept of the *circle of confusion*, which represents the largest blur spot that is still perceived as a point by the human eye. The equation governing the depth of field, taking into account these factors, is:

$$\text{DoF} = \frac{2 \cdot u^2 \cdot N \cdot c}{f^2} \tag{2.3}$$

where:

- u is the distance to the subject,

- N is the f-number (aperture),
- c is the circle of confusion diameter,
- f is the focal length.

This equation shows that for a given subject distance u, increasing the f-number N (smaller aperture) or decreasing the focal length f increases the depth of field.

Focusing Mechanism

Focusing on modern cameras can be manual or automatic. Manual focus involves adjusting the lens elements by hand to bring the subject into focus. Automatic focus (autofocus) systems use sensors, control systems, and motors to adjust the lens elements automatically, ensuring that the subject is sharp. Autofocus systems can operate based on contrast detection, phase detection, or hybrid methods, each with its strengths and limitations depending on the shooting scenario.

2.3.3 Types of Cameras Used in Production

The most commonly used types of cameras in production environments are DSLR, mirrorless, and cinema cameras, each offering unique advantages and catering to specific production needs.

DSLR (Digital Single-Lens Reflex) cameras are renowned for their durability, extensive range of interchangeable lenses and robust build quality. These cameras utilize a mirror mechanism that directs light from the lens to an optical viewfinder, allowing photographers to see through the lens with high clarity. The optical viewfinder is particularly beneficial in bright outdoor conditions, where electronic displays may be less effective. However, the mirror mechanism can be bulky and noisy, making DSLRs less suitable for discreet or lightweight setups.

Figure 2.12: Mirrorless Canon Camera.

Figure 2.13: This figure illustrates the selection of focal lengths based on different shooting scenarios. The first image (left) uses a 100mm lens for a close-up shot of a Lego. The second image (center) utilizes a 50mm lens, suitable for presentation videos. The third image (right) features a two-person interview captured with a 15mm lens, providing a wider field of view to accommodate both subjects in the frame.

Mirrorless cameras, in contrast, lack the mirror mechanism found in DSLRs, leading to a more compact and lightweight design. This absence of a mirror allows for quieter operation and often faster autofocus in live view mode. Mirrorless cameras typically feature electronic viewfinders (EVFs) that provide real-time previews of exposure and settings, which can be advantageous for video production. Their smaller size and advanced features make them highly versatile and suitable for both still photography and video recording in diverse environments.

Cinema cameras are specifically designed for professional video production, offering unparalleled image quality, dynamic range, and manual control options. These cameras often support high-resolution recording, extensive color grading capabilities, and compatibility with various cinema lenses. While they excel in producing cinematic visuals, cinema cameras are generally larger, heavier, and more complex to operate than other camera types. They are predominantly used in high-end filmmaking, where precise control over the image is of high importance.

Given the importance of high-quality recordings for both production and research, as well as flexibility, we choose to utilize eight Canon EOS R5C cameras for our studio setup. The Canon EOS R5C is a mirrorless camera capable of recording up to 8K video at high frame rates, offering cinematic features for both still photography and video production. Its mirrorless design makes it flexible and portable, ideal for a wide range of creative and research applications.

2.3.4 Types of Lenses Used in Production

The primary types of lenses include prime lenses, zoom lenses, wide-angle lenses, telephoto lenses, and speciality lenses, each offering distinct advantages.

Prime lenses are characterized by a fixed focal length, which typically results in superior optical quality, wider apertures, and better low-light performance. The simplicity of prime lenses, with fewer moving parts, often translates into sharper images with minimal optical

distortions. However, their fixed focal length prevents changing lenses to achieve different compositions, which can be a limitation in dynamic shooting situations.

Zoom lenses, with their variable focal lengths, provide flexibility in framing without the need to switch lenses. This makes them very useful in fast-paced environments where rapid changes in perspective are required. However, zoom lenses often have smaller maximum apertures compared to prime lenses, which can compromise low-light performance. Additionally, they may exhibit optical distortions, especially at extreme focal lengths.

Wide-angle lenses are designed with short focal lengths, typically ranging from 10mm to 35mm, allowing them to capture expansive views. This capability is ideal for landscapes, architecture, and tight indoor spaces. While wide-angle lenses can introduce perspective distortion, making objects at the edges of the frame appear stretched, this effect can be creatively exploited or corrected during post-production.

Telephoto lenses, with their long focal lengths, generally between 70mm and 300mm or more, are essential for capturing distant subjects and are commonly used in wildlife, sports, and portrait photography. These lenses compress the perceived distance between objects, creating a flattering and focused perspective. However, their size and weight can make them heavy, often requiring stabilization equipment to avoid camera shakes.

Specialty lenses, such as macro lenses and fisheye lenses, serve specific purposes. Macro lenses are optimized for close-up photography, allowing for high magnification and capturing fine details. Fisheye lenses offer an ultra-wide field of view with significant barrel distortion, producing a unique, curved perspective. These lenses are particularly useful for creative effects and specialized scientific applications.

For our lenses, we select lenses with focal lengths ranging from 15mm to 200mm and f/stop (aperture) ranging from f/1.2 to f/22. This allows us to capture scenes at varying distances, offering both close-up and wide-angle perspectives. The different f/stops allow for a broad spectrum of depth of field options. This versatility allows us to capture different kinds of scenes for different purposes, such as Legos for research (see Figure 1.1) and multi-person interviews for production.

We also disable auto-focus in the lenses, as this feature changes the focus during recording. This is problematic for most of our recording scenarios, as we want a fixed focus on the subject throughout the recording.

2.3.5 Camera Placement and Configuration

In our studio setup, careful consideration is given to the positioning of the eight cameras around the green screen and the subject. We arrange the cameras in a way that meets the needs of both production and computer vision research. For a recording session, we utilize two main cameras equipped with prime lenses, specifically the 50mm or 85mm lens, to capture high-quality portrait shots of the subject, which is typical for production videos. These lenses are chosen for their ability to deliver sharp and pleasing images, which is

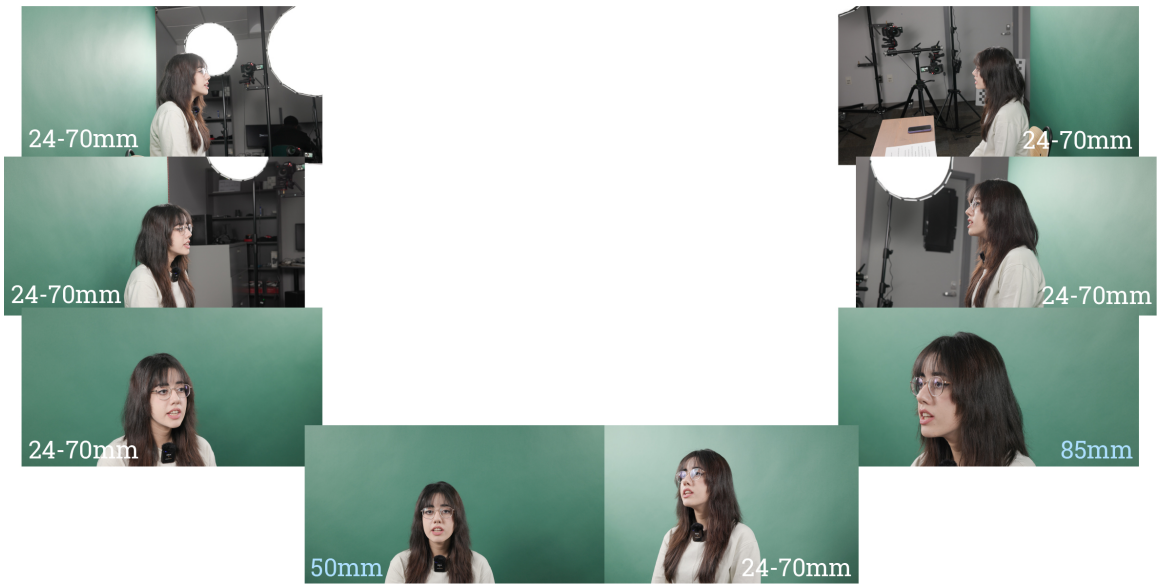


Figure 2.14: Example of a recording session with the camera setup and lenses utilized. The main cameras are equipped with prime lenses (50mm and 85mm) for high-quality portrait shots, while the remaining cameras use zoom lenses (24-70mm) for versatile framing.

essential for professional video production. The remaining cameras are equipped with zoom lenses, offering flexibility in changing the focal length as required. This setup is important for capturing a variety of perspectives and ensuring that no detail is missed during recording.

With computer vision research and analysis tasks like 3D reconstruction in mind, the cameras are strategically positioned at varying horizontal and vertical locations. We also use varying focal lengths for the cameras equipped with zoom lenses, providing both close-up and wide-angle views. This arrangement ensures that we capture a wide range of views, providing the necessary data for accurate multi-view analysis.

We minimize the horizontal displacement between adjacent cameras to ensure that they are not positioned too far apart to create overlapping views. This is very important for our multi-camera calibration process, and this prevents potential 3D alignment issues such as mismatched perspectives or alignment errors.

Each camera is indexed from A to H, representing cameras 1 to 8. This indexing system helps identify each camera's recording during post-production, such as identifying cameras with external audio sources.

2.4 The Exposure Triangle

Exposure refers to the total amount of light that reaches the camera's sensor during the process of capturing an image. It directly affects the brightness and detail visible in the final image. In the context of our studio, achieving the correct exposure is essential for

maintaining high and consistent image quality. This ensures that the captured videos meet the desired artistic and technical standards.

The Exposure Triangle is a fundamental concept in photography that describes the relationship between three key elements: ISO, shutter speed, and aperture. These components work together to determine an image's exposure. Proper control and balance of these elements are important for capturing high-quality images and videos, especially in a controlled environment such as the studio.

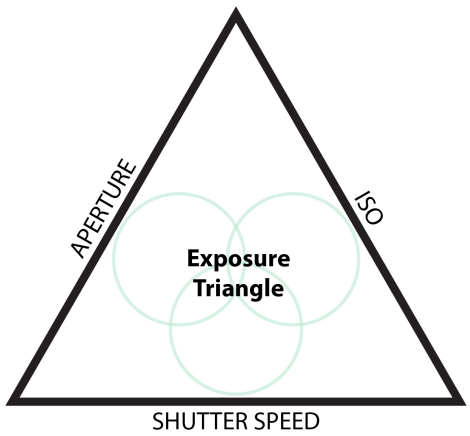


Figure 2.15: Exposure triangle, consisting of aperture, ISO and shutter speed.

2.4.1 ISO

ISO measures the sensitivity of the camera's sensor to light. A lower ISO value (e.g., 100 or 200) indicates lower sensitivity, which is ideal for bright lighting conditions as it produces cleaner images with less noise. Conversely, a higher ISO value (e.g., 1600 or 3200) increases the sensor's sensitivity, making it suitable for low-light situations. However, higher ISO settings can introduce noise, degrading image quality. In our studio, since we have artificial light sources bright enough to illuminate the scene, we utilize ISO values within the range of 100 to, at most, 1600 to maintain clear and high-quality captures.

2.4.2 Shutter Speed

Shutter speed determines the duration for which the camera's sensor is exposed to light. It is measured in fractions of a second (e.g., 1/60, 1/1000). A faster shutter speed (shorter duration) is used to freeze motion, making it ideal for capturing fast-moving subjects without blurring. A slower shutter speed (longer duration) allows more light to reach the sensor, which is useful in low-light conditions or when a motion blur effect is desired.

2.4.3 Aperture

Aperture refers to the size of the opening in the lens through which light enters the camera. A lower f-stop number corresponds to a larger aperture, allowing more light to enter the

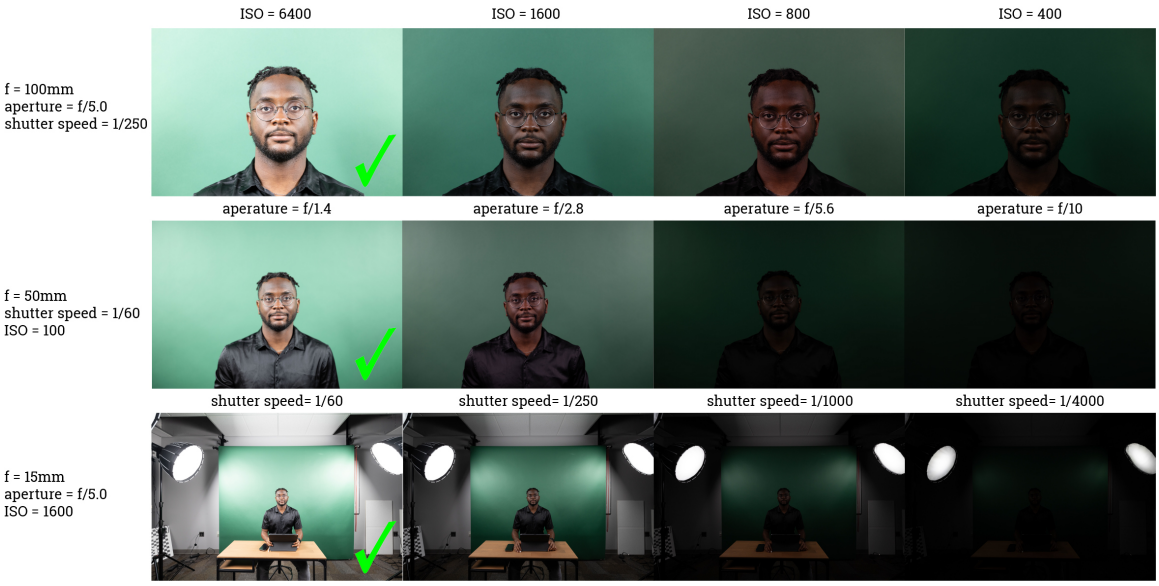


Figure 2.16: This figure demonstrates the effects of varying ISO, aperture, and shutter speed on image exposure. Each row represents images captured with a fixed focal length and a fixed pair of two exposure settings, while the third setting varies across the columns. The first row shows the impact of changing ISO, the second row, aperture, and the third row, shutter speed.

camera and creating a shallower depth of field. This is useful for isolating subjects from the background by creating a pleasing bokeh effect.

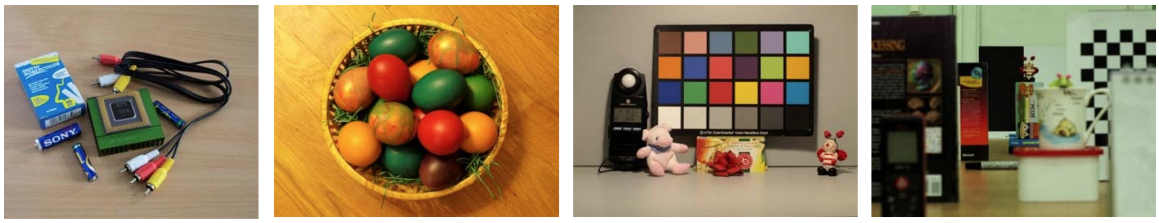
2.4.4 Balancing the Exposure Triangle

Balancing ISO, shutter speed, and aperture is key to achieving the correct exposure while meeting creative and technical requirements. This balance is especially important in our studio, where we work with diverse lighting setups and subject matter. For instance, a low-light scene may require a higher ISO and a wider aperture (lower f-stop) to maintain a fast enough shutter speed and avoid blur. Conversely, a brightly lit scene might necessitate a lower ISO, a smaller aperture (higher f-stop), and a faster shutter speed to prevent overexposure.

For standard recording in our studio, with the light sources set at 5600K color temperature, we manually set the ISO to 1600, aperture to f/5.0, and shutter speed to a 180-degree shutter angle. These settings are carefully chosen to ensure consistency in exposure throughout the recording process.

Our choice of aperture at f/5.0 allows for a sufficiently deep depth of field, ensuring that the entire scene remains in focus. This is important in our studio environment, where multiple elements and subjects may need to be captured sharply.

To adhere to the 180-degree shutter rule, we set our shutter speed to double the video's frame rate. Since our intended frame rate for recording is 50fps, the corresponding shutter



(a) Room scene (shade) (b) Energy saving lamp (c) Macbeth chart (d) Fluorescent light

Figure 2.17: Test images illuminated by different light sources without white balance [67].

speed is 1/100s. This rule helps achieve the natural motion blur characteristic of cinematic videos, enhancing the visual quality and realism of the footage.

In Canon cameras, shutter speed is represented in degrees rather than fractions of a second. To convert the traditional shutter speed value to a shutter angle, we use the formula:

$$\text{Shutter Angle} = \frac{\text{Shutter Speed} \times 360^\circ}{\text{Frame Rate}} \tag{2.4}$$

Thus, for a shutter speed of 1/100s at a frame rate of 50fps, the shutter angle is:

$$\text{Shutter Angle} = \frac{1/100 \times 360^\circ}{50} = 180^\circ \tag{2.5}$$

This precise control over exposure settings ensures that our recordings maintain a consistent look and feel, regardless of the subject matter being filmed.

2.5 White Balance

White balance is an aspect of the imaging pipeline that ensures the colors in an image appear natural and consistent under various lighting conditions. It refers to the process of adjusting the colors so that the whites in an image appear truly white, thereby balancing the other colors accordingly.

There are two primary methods for achieving white balance: manual white balance and automatic white balance (AWB). Manual white balance involves setting a specific color temperature, measured in Kelvin (K), that corresponds to the lighting conditions. This ensures that the colors in the images are consistent and accurate, even when lighting conditions vary. This method is useful in controlled environments, such as our studio, where the lighting conditions are known and constant.

Automatic white balance, on the other hand, is a feature that automatically adjusts the white balance based on the perceived color temperature of the scene. This involves analyzing the scene using computer vision algorithms and making real-time adjustments to the color settings to achieve a neutral color balance. While AWB can be convenient in dynamic and

unpredictable lighting situations, it may not always provide the desired accuracy, especially in mixed lighting conditions or when creative control over the color palette is needed.

In our studio, we manually choose a white balance setting of 5600K in each camera setting, which corresponds to the color temperature of daylight. This decision aligns with the color temperature of our studio light sources, which are specifically set to simulate natural daylight. By matching the white balance to the color temperature of the lighting, we ensure that the colors in our recordings are accurate.

To maintain consistency, we disable automatic white balance (AWB) on our cameras. In this environment, where the lighting conditions remain constant, there is no need for the camera to adjust the white balance dynamically. By fixing the white balance at 5600K, we ensure that the color representation remains consistent throughout the recording sessions.

2.6 Microphone

Audio quality is an important component of the video production process, significantly impacting the viewer's experience. Good audio can enhance a video's clarity and impact, while poor audio can diminish the overall quality, making even the best visuals less effective. Capturing high-quality audio is essential for ensuring that the final production resonates with the audience, whether the content is an interview, a lecture, or a creative film project.

Microphones are essential tools in audio production that allow us to capture sound with precision and clarity. The choice of microphone and its setup can significantly influence the quality of the recorded audio, making it crucial to select the appropriate type for the specific recording environment and purpose.

2.6.1 Types of Microphones

Microphones come in various types, each designed for specific recording scenarios.

Dynamic Microphones are ideal for live performances and situations where the microphone might be exposed to rough handling. They do not require external power and are less sensitive to high sound pressure levels, making them a good choice for capturing loud sounds, such as musical instruments or live events. However, they may not be as sensitive or accurate as other types when it comes to picking up the subtle details of softer sounds.

Condenser Microphones are more sensitive and provide a broader frequency response, making them well-suited for studio recording. They are often used to capture vocals and instruments with great detail and accuracy. This type of microphone is ideal for capturing the subtle details of performance but may pick up more ambient noise, so it is best used in a controlled environment.

Shotgun Microphones are highly directional, meaning they are designed to pick up sound from a specific direction while rejecting noise from other directions. This makes them ideal for film and video production, where it is important to capture focused sound from

the subject while minimizing background noise. Shotgun microphones are often used in situations where the microphone needs to be placed out of the camera's frame, such as in interviews or on-set dialogues.

Lavalier Microphones, also known as lapel mics, are small and typically clipped to a person's clothing. They are commonly used in presentations, interviews, and other situations where a discreet microphone is needed. Wireless Lavalier microphones provide freedom of movement, making them an excellent choice for dynamic presentations or scenarios where the subject needs to move around without being tethered by cables.

2.6.2 Relevant Microphone Concepts

When setting up microphones for studio use, it is important to understand several key concepts that can affect audio quality:

Decibels (dB): Decibels are a unit of measurement for sound intensity. In audio recording, dB levels are used to express the loudness of a sound. This is important for setting microphone gain and ensuring that the audio is neither too quiet nor too loud, which could cause distortion.

Gain: Gain refers to the amplification of the microphone's signal. If the gain is set too high, the audio may clip, leading to a distorted recording. If it is too low, the audio may be too quiet and require excessive amplification in post-production, which can introduce noise.

Frequency Response: This is the range of frequencies that a microphone can pick up. A flat frequency response means that the microphone captures all frequencies equally, while a shaped frequency response might boost certain frequencies. Understanding the frequency response of your microphone is important for ensuring it is well-suited to the sound source you are recording.

2.6.3 Microphone Setup in Our Studio

Given the diverse needs of our production environment, we have selected a variety of microphones to cover different scenarios.

For capturing dialogue and interviews, we primarily use **shotgun microphones** such as the Rode VideoMic NTG and Sennheiser MKE 400. These microphones allow us to isolate the speaker's voice while minimizing ambient noise, making them ideal for clear and focused audio recording in both controlled studio settings and more unpredictable environments.

In situations where presenters or actors need to move freely, we employ **wireless lavalier microphones** like the Rode Wireless GO II. The flexibility provided by wireless lavaliers is crucial in dynamic shoots, allowing for unobtrusive audio capture without restricting the movement of the subject.

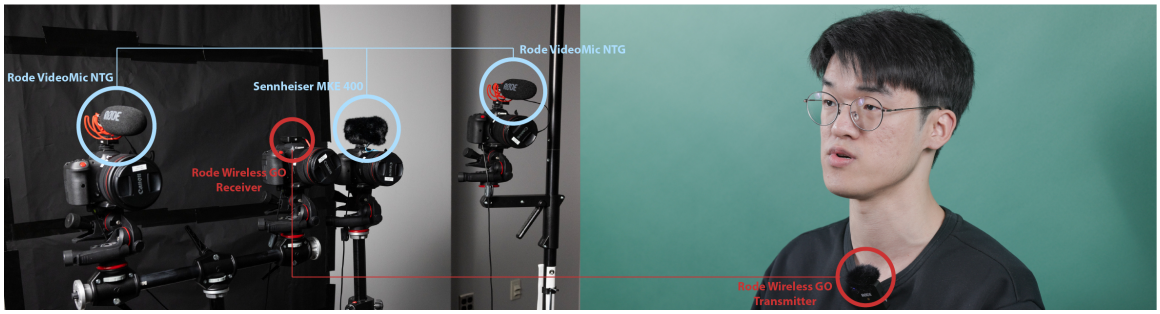


Figure 2.18: Audio setup: Rode VideoMic NTG and Sennheiser MKE 400 mounted on main cameras facing the subject. Rode Wireless GO II receiver mounted on another camera, transmitter clipped to the speaker's collar.

2.6.4 Microphone Placement

The shotgun microphones are mounted on top of the main cameras, with the microphone head facing the subject. The receiver of the Lavalier microphone is mounted on top of one of the other cameras, and the transmitter is hung on the collar of the speaker so the mouthpiece is as close as possible to the speaker. These configurations ensure high-quality audio capture tailored to our specific requirements.

2.6.5 Microphone Audio Configuration

To optimize audio quality, we conducted a series of tests to determine the best settings for our studio environment. We experimented with various dB levels and found that setting the microphone input to -6dB provided the best balance between capturing clear sound and avoiding distortion.

Additionally, to prevent the internal microphones of the cameras from picking up unwanted ambient noise or interfering with the high-quality audio from external sources, we set the internal microphone levels to 0 for cameras equipped with external audio inputs. This eliminates any echo or phase issues that could arise from using multiple audio sources, ensuring that the recorded sound remains clean.

2.7 Timecode

Synchronization is a fundamental aspect of both professional video production and multi-view computer vision research. In video production, accurate synchronization ensures that footage captured from multiple cameras can be seamlessly integrated, maintaining continuity and consistency in scenes. This is particularly crucial during live events or shoots involving multiple angles, where timing discrepancies could lead to misaligned audio, visual disruptions, or significant editing challenges.

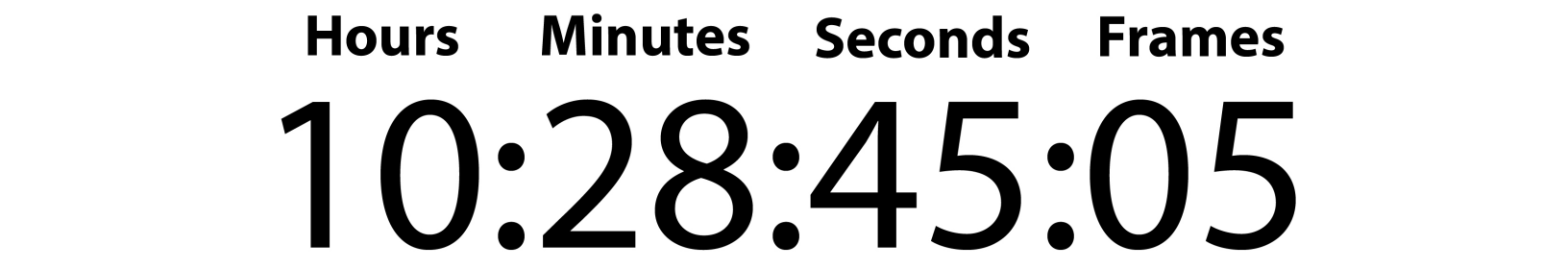


Figure 2.19: Example of timecode labeling, showing how each frame is assigned a unique identifier in the format HH:MM:SS:FF (Hours:Minutes:Seconds:Frames).

In multi-view computer vision, precise synchronization is equally critical, especially for applications like 3D reconstruction. Accurate timing across different camera views is essential for reconstructing a precise three-dimensional model of a scene. This requires capturing the exact same moment from various perspectives, making time synchronization vital for ensuring that all cameras are perfectly aligned in time. Even slight discrepancies, such as those in the millisecond range, can introduce errors in depth information, leading to inaccuracies in the reconstructed model.

Timecode is a system used to precisely label all the frames in a recording, allowing for the exact identification of a specific point in time within the footage. Each frame in a video or audio recording is assigned a unique timecode value, typically represented in hours, minutes, seconds, and frames (HH:MM:SS:FF). Figure 2.19 illustrates what timecode looks like, showing how each frame is sequentially labeled with a unique timecode value.

2.7.1 Types of Timecode

Several types of timecode are used in the industry, each with its specific applications and characteristics. The most commonly used is the SMPTE (Society of Motion Picture and Television Engineers) timecode, which is widely adopted in video and audio production. SMPTE timecode is an 80-bit digital code that provides a time reference to synchronize video and audio signals. It includes information about the time of day, the frame count, and the type of timecode being used (drop frame or non-drop frame).

2.7.2 Free Run Timecode vs. Record Run Timecode

Timecode can be configured in different modes depending on the specific requirements of the production. The two most common modes are Free Run and Record Run.

Free Run Timecode is a mode where the timecode continues to run continuously, regardless of whether the camera is actively recording or not. This is useful in situations where multiple cameras are used over extended periods, as it ensures that all cameras maintain the same time reference throughout the shoot. Free Run Timecode is particularly

beneficial for multi-camera setups, as it simplifies the synchronization process during post-production.

Record Run Timecode, on the other hand, only advances the timecode when the camera is recording. This mode is typically used in single-camera setups or situations where the camera only records intermittently. Record Run Timecode ensures that there is no gap in the timecode during recording but requires more attention during post-production to ensure proper synchronization with other footage or audio sources.

2.7.3 Timecode Frame Rate

The frame rate of timecode refers to the number of frames per second (fps) that are labeled with timecode values. Common frame rates include 24 fps, 25 fps, and 30 fps. The choice of frame rate is critical, as it must match the frame rate of the recorded video to ensure proper synchronization.

In our configuration, the timecode frame rate is set to 25 fps, which aligns with the Phase Alternating Line (PAL) video standard commonly used in many parts of the world. This frame rate ensures compatibility with our production cameras and simplifies the post-production process.

2.7.4 Drop Frame and Non-Drop Frame Timecode

Drop frame and non-drop frame timecode are two methods for addressing the discrepancy between a video's nominal frame rate and its actual playback speed.

Nominal frame rate refers to the theoretical or intended frames per second (fps) at which a video is supposed to play back. However, due to technical reasons, the actual playback speed might slightly differ from the nominal rate. For example, the nominal rate might be 30 fps, but the actual playback speed could be 29.97 fps.

Non-Drop Frame Timecode (NDF) counts every frame sequentially without skipping any. This method is straightforward but can lead to slight timing inaccuracies over long periods because the actual frame rate of video (e.g., 29.97 fps) is slightly less than the nominal rate (30 fps). Over an hour of footage, this discrepancy can result in a timecode error of a few seconds.

Drop Frame Timecode (DF) compensates for this discrepancy by occasionally skipping (or "dropping") frame numbers. DF timecode drops frame numbers at specific intervals to ensure that the timecode accurately reflects the real-time duration of the video. It is important to note that DF timecode does not actually drop any frames from the video; it only skips certain timecode numbers to correct the time drift. This makes DF timecode more accurate for long recordings, particularly in broadcast television.

(a) Tentacle Sync E Timecode Generator

(b) 256GB Camera SD Cards

Figure 2.20: (a) A compact timecode generator for setting the timecode of each camera, and (b) a set of SD cards used for storing the recordings from each camera in our studio.

2.7.5 Timecode Configuration

In our setup, we utilize the Tentacle Sync E, a compact and user-friendly timecode generation device. The Tentacle Sync E generates a consistent timecode reference, which is used to set the timecode of each camera in the setup. By doing so, all cameras are synchronized to the same timecode, aligning their recordings as closely as possible in time. However, it is important to note that this approach does not entirely eliminate the possibility of time shifts between cameras, which could introduce slight discrepancies in the synchronization.

In our configuration, we set the camera's Time Code Run to *Free Run*, and our timecode framerate is set to 25 fps. This means the timecode runs continuously, regardless of whether the cameras are actively recording. This setting is important for maintaining synchronization throughout extended recording sessions, as it ensures that all cameras maintain a consistent time reference.

2.8 Storage

On-camera storage is primarily handled using SD cards, which provide a portable medium for recording raw footage directly from the cameras. Each camera in our setup is equipped with an SD card slot, enabling the storage of recorded video files. Several factors contribute to the size of video files, affecting both the quality of the footage and the storage capacity needed.

Framerate The framerate determines the number of frames captured per second. Higher framerates, such as 50 fps or 60 fps, result in smoother motion representation but also increase the data rate, as more frames need to be recorded and stored per second.

Resolution Resolution refers to the number of pixels used to display an image or video, typically described in terms of width by height (e.g., 1920x1080 for Full HD). Higher

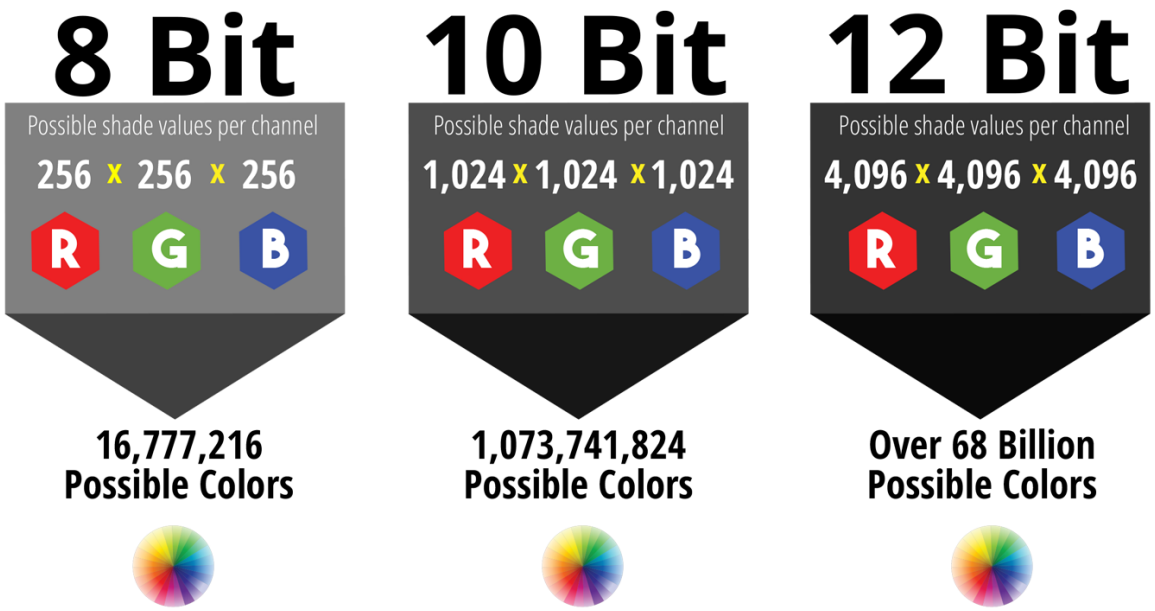


Figure 2.21: Chart showing the possible colors in 8-bit, 10-bit and 12-bit video [60].

resolutions, such as 4K (3840x2160), contain more pixels, resulting in sharper and more detailed images. This increase in pixel count enhances image quality but also means that each frame requires more data to store, thereby increasing the overall file size. For instance, a 4K image has four times the number of pixels compared to a Full HD image, leading to a corresponding increase in storage requirements.

Recording Format and Bit Depth The recording format determines how the video data is compressed and stored. Uncompressed formats produce the highest quality but result in very large file sizes, while compressed formats reduce file size at the cost of some quality loss. The bit depth, such as 8-bit, 10-bit or 12-bit, refers to the number of bits used to represent the color information for each pixel. A higher bit depth allows for a wider range of colors and finer gradation, enhancing the visual quality while increasing the file size.

Bit Rate Also known as data rate, it measures how much data is processed (recorded or played back) per second of video. Higher bit rates generally improve video quality, as more data is available to represent the image, reducing compression artifacts. However, higher bit rates also result in larger file sizes. This means that the bit rate must be carefully balanced with storage capacity and quality requirements, as excessive bit rates can consume storage quickly without significant quality gains.

These settings collectively determine the size of the recorded video files and, consequently, the storage capacity required.

2.8.1 Camera Video Settings

We utilize full-frame settings with a frequency of 50.00 Hz and a frame rate of 50 fps. The 50.00 Hz frequency setting is chosen to match the local power supply frequency, reducing the risk of flicker in the video caused by artificial lighting. This allows us to capture detailed videos with smoother motion and reduced motion blur. We record in a 10-bit color depth format, which provides a broader range of color information. Specifically, we use the XF-AVC 10-bit file format, which is developed by Canon for 4K UHD footage and is suitable for professional workflows such as ours. We set the recording resolution to 4K, with a bit rate of 260 Mbps, which balances quality and file size effectively. Given these video settings, we are able to store 129 minutes of footage on a 256GB SD card.

Chapter 3

Data Capture and Processing

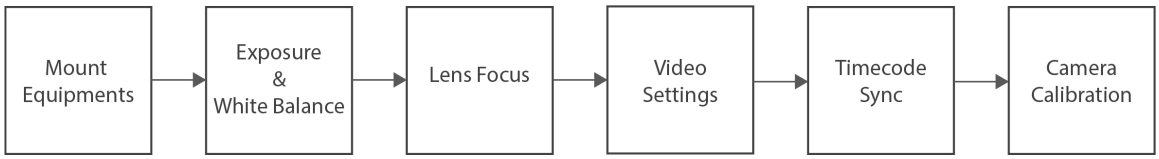


Figure 3.1: Step-by-Step Capture Procedure Pipeline for Each Recording Session. We begin by carefully mounting the recording equipment, including cameras, lenses, and lighting. Next, we ensure accurate exposure and confirm that the white balance is correctly aligned with the light source's color temperature. The lens focus is manually adjusted to guarantee the subject is sharply in focus. We then verify that video settings, such as framerate and recording format, are configured correctly. Prior to initiating the recording, we synchronize the timecode across all cameras. Once recording begins, we proceed with the calibration process to ensure alignment and consistency.

Now that all the studio components are set up, this section focuses on capturing data with this setup, including transferring raw data to storage servers, encoding raw video files for easy distribution, and processing the encoded video files.

3.1 Data Capture

The first step in the data capture process is to ensure that all equipment, including cameras, light sources, and external audio microphones, are turned on and functioning as expected with the preset configurations. For the cameras, it is essential to confirm that the white balance matches the color temperature of the light sources. Additionally, we check to ensure sufficient storage space on the SD cards to prevent interruptions during the recording process due to limited storage capacity.

Once all components are verified to be operating correctly, we proceed to the next stage of the capture process.

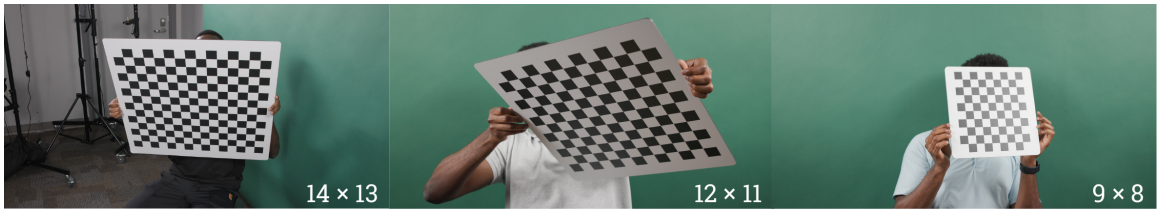


Figure 3.2: Checkerboard calibration patterns used in the studio: (left) largest (14 × 13), (middle) medium (12 × 11), and (right) smallest (9 × 8).

3.1.1 Synchronization Process

As previously discussed, each camera operates with its own running timecode. To synchronize the timecode across all cameras before recording, we use the timecode generator to update the timecode of each camera. According to the official documentation, once synced, the Tentacle Sync E can maintain synchronization with a drift of less than one frame in 24 hours. Frame drift refers to the gradual loss of synchronization between frames of a multi-camera setup over time. In our setup, we re-synchronize all cameras using the timecode generator before each recording session, which typically lasts no more than two hours. This practice ensures we avoid any issues related to frame drift.

During the synchronization process, we update the timecode of each camera and simultaneously start the recording process for each camera.

3.1.2 Calibration Process

With all cameras recording, we check and ensure that they remain stable and are not subject to any movement before starting the calibration process. It is important that the position and orientation of the cameras remain unchanged throughout the recording, as this ensures accurate 3D reconstruction of the scene using the correct camera parameters. The calibration process involves displaying a pattern (a checkerboard pattern in our case) from different orientations to each camera. This pattern will later be used to obtain the camera parameters, including the pose of each camera in the scene.

In our studio, we utilize three checkerboard calibration patterns of different dimensions. The largest board has a dimension of 14 × 13 with a square side of 0.042 meters, the medium has 12 × 11 with a square side of 0.035 meters, and the smallest is a 9 × 8 pattern with a square side of 0.0269 meters. The use of different pattern dimensions provides several benefits. Larger patterns offer more points for the calibration process, enhancing the accuracy of the camera parameters. Smaller patterns, on the other hand, are easier to manage in tighter spaces or when a closer range capture is needed. This variety ensures flexibility and precision in different recording scenarios.

To ensure we know the exact frames where a pattern orientation is shown, we use a clapboard to mark the frame. This allows us to easily identify each calibration frame using

Figure 3.3: Clap Board, for marking frames using the audio waveform during recording.

the waveforms of the audio recordings during the processing stage. Additionally, we use the clapboard to mark the frame where we capture a clean plate of the scene, which is useful in the green screen keying stage of the pipeline.

3.2 Data Transfer

Once the recording session is complete, we begin the process of transferring the raw data from the cameras to our storage servers.

Transferring and storing large volumes of high-resolution video data presents significant challenges. One of the significant challenges we encountered was transferring these very large recording files quickly enough to our storage servers in situations where we had to record multiple sessions. This challenge becomes critical when we record more than 4 hours per day. For instance, consider a 5-hour recording session. With a 1 Gbps connection, it takes at most 125 MB/s to transfer data to our storage servers. This means it will take:

$$\text{Transfer Time to Storage Server} = \frac{8 \times 256 \text{ GB} \times 1024 \text{ MB/GB}}{125 \text{ MB/s}} \approx 5 \text{ hours} \tag{3.1}$$

to copy all recordings from the 8 SD cards to the storage server.

Since the SD cards can only record videos that are approximately two hours long, we must switch SD cards every two hours. Emptying the previous SD cards takes approximately five hours, creating a bottleneck. We can record for approximately four hours straight, but then we have to wait for the first set of SD cards to be emptied, resulting in a three-hour waiting time.

One option is to utilize a high-speed network by increasing the Ethernet transfer connection from 1 Gbps to 10 Gbps, but this would require changing the wiring, which is expensive. Another option is to utilize Solid-State Drives (SSDs), which offer faster read and write speeds and improved reliability compared to traditional HDDs while being less expensive than high-speed networks.

To address this issue, we utilize a Samsung EVO 970 2 TB SSD with a maximum write speed of 3300 MB/s. This allows us to transfer all recordings from the 8 cameras at once

to the SSD using an SD card reader with a transfer speed of approximately 250 MB/s per card. The combined transfer speed of all SD cards is:

$$\text{Combined Transfer Speed} = 8 \times 250 \text{ MB/s} = 2000 \text{ MB/s} \tag{3.2}$$

which is less than the maximum write speed of 3300 MB/s of the SSD. With the SSD, the transfer time for a single set of 8 SD cards is:

$$\text{Transfer Time to SSD} = \frac{8 \times 256 \text{ GB} \times 1024 \text{ MB/GB}}{2000 \text{ MB/s}} \approx 0.5 \text{ hours} \tag{3.3}$$

Thus, after recording for 4 hours, we only need approximately 30 minutes to transfer the recordings to the SSD, rather than 5 hours to transfer directly to the storage servers with zero waiting time. Given that the SSD has a capacity of 2 TB, it can hold up to:

$$\frac{2000 \text{ GB}}{256 \text{ GB}} \approx 7.8125 \approx 7 \text{ sessions} \tag{3.4}$$

or 14 hours of recording. The recordings on the SSD can then be moved to the storage servers in the background.

In summary, by utilizing the Samsung EVO 970 2 TB SSD, we have effectively reduced the data transfer bottleneck, enabling more continuous and efficient recording sessions in our Computational Photography Research Studio.

3.3 Time Synchronization

After the recorded video data has been successfully stored on the studio server, the first stage of our preprocessing steps is to synchronize all the records from each of the cameras to have the same start time and duration.

The reference start time and duration are automatically determined using the formulas below:

$$S = \max(T_i) \tag{3.5}$$

$$D = \min(S + D_i) \tag{3.6}$$

Where:

- T_i is the timecode of the first frame for the i-th camera.
- D_i is the duration for the i-th camera.
- S is the timecode T of the camera that started recording last.

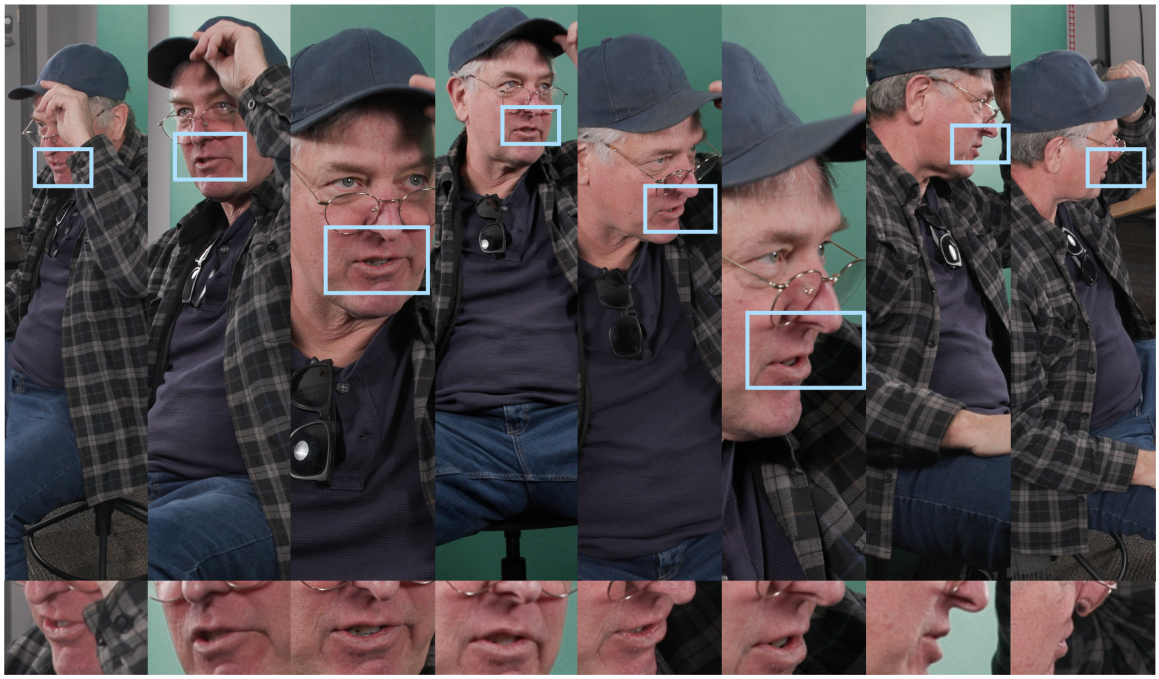


Figure 3.4: Example of a regular recording where the phase shift is not perceptible due to the absence of fast-paced motion.

- D is the duration of the shortest video starting from S

The timecode information is accessible in the metadata of each camera's video recording. Using this, we can determine the start time and duration in seconds for each camera to achieve a trimmed and synchronized set of videos. The equations for obtaining the start time and duration for each camera are given by:

$$ss_i = \frac{S - T_i}{F_i} \tag{3.7}$$

$$t_i = \frac{D}{F_i} \tag{3.8}$$

Where:

- $S - T_i$ is the number of frames to be skipped from the start of the i-th camera's video.
- F_i is the framerate of the i-th camera.
- ss_i is the final start time in seconds for the i-th camera.
- t_i is the final duration in seconds for the i-th camera.

These parameters, s_s and t, can be easily used with `FFmpeg` to trim and synchronize the videos.

Figure 3.5: Illustration of a fast-paced event using a clapboard, highlighting the visible phase shift difference across camera frames.

3.3.1 Phase Shift in Cameras

A bottleneck we discovered was that even after synchronizing with the timecode generator, analyzing the audio waveforms of different cameras (as illustrated in Figure 3.6) revealed at most about 30 millisecond difference between any two camera frames. This discrepancy, known as phase shift in cameras, arises from slight variations in each camera's mechanical shutter/exposure timing. Although synchronized to the same timecode, the actual start time of frame capture can vary slightly across cameras due to variations in the internal clocks, triggering mechanisms, and electronic processing speeds.

This phase shift is not visually obvious in most of our recordings, which do not involve fast-paced activities. For instance, in Figure 3.4, we show a regular recording where this difference is not perceptible. However, in Figure 3.5, we illustrate a fast-paced event using a clapboard to highlight this difference.

Phase shift in cameras is due to the variations in each camera's mechanical shutter/exposure timing, resulting in slight differences in the timing of frame capture. Each camera has an internal clock that governs its timing and synchronization, and small inaccuracies in these clocks can lead to phase shifts over time. Additionally, variations in how cameras are triggered to start recording can cause differences in frame capture times. Differences in the electronic processing speed of cameras can also result in phase shifts.

3.3.2 Mechanisms of Shutter Operation in Video Recording

In video recording, the operation of the shutter differs significantly from still photography. The process involves both mechanical and electronic components to manage exposure and

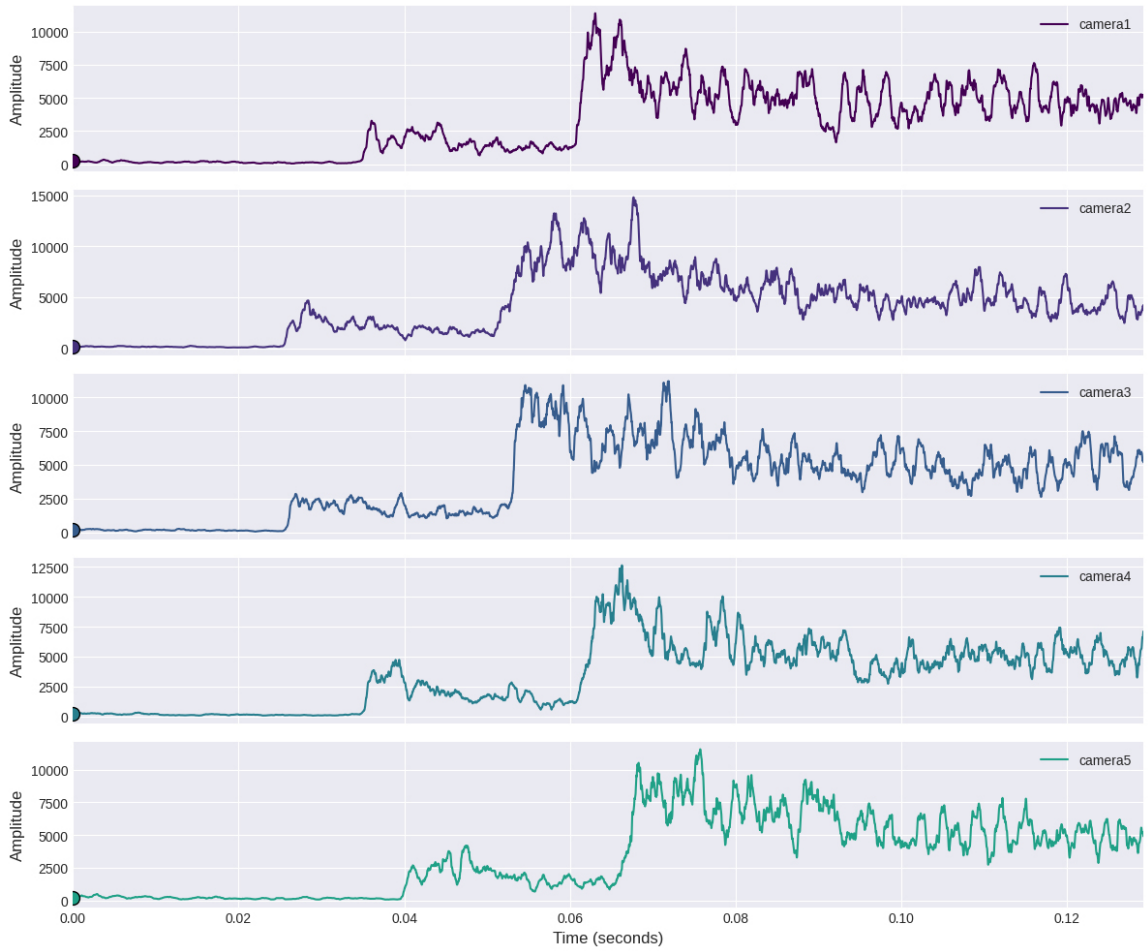


Figure 3.6: Audio waveforms of the clap signal from five synchronized cameras, illustrating a phase shift of at most 30 millisecond between any two camera frames. This discrepancy is due to variations in each camera's mechanical shutter and exposure timing, despite synchronization to the same timecode.

frame capture efficiently. At the start of video recording, the mechanical shutter opens to allow continuous light exposure to the sensor. The mechanical shutter generally remains open for the duration of the recording, with the sensor using an electronic shutter mechanism to control exposure. The sensor controls exposure electronically for each frame, with the electronic shutter exposing the sensor for each frame. This process is managed by the camera's internal clock to ensure consistent intervals, such as 1/50th of a second for 50 fps.

3.3.3 Camera Internal Clock

The internal clock in a camera ensures consistent frame capture rates, manages sensor readout timing, and coordinates various internal processes. This is crucial for timing and synchronization and is essential for video recording, as it ensures frames are captured at consistent intervals. However, potential issues can arise with the internal clock. Clock drift, resulting from small inaccuracies, can lead to frame rate drift over time. Additionally, temperature sensitivity can affect clock accuracy, causing timing discrepancies.

3.3.4 Managing Phase Shifts

To manage phase shifts and ensure synchronization across multiple cameras, several techniques can be employed. Genlock systems use a master clock to synchronize all cameras to the same timing source, ensuring uniform frame capture timing across the setup. Time code generators provide a consistent time reference that all cameras can follow, facilitating coordinated operation. Manual synchronization involves aligning frames based on audio or visual cues, such as claps, captured by all cameras, allowing for post-processing alignment.

While continuous synchronization using Genlock would mitigate the frameshift issue, our cameras do not support Genlock. Instead, we analyze the audio waveforms to precisely measure the shifts and determine how much the video frames of each camera are apart.

To achieve this, we first identify the camera with the minimum delay as the reference for alignment by comparing the clap signals in the audio waveforms of each camera. We take audio segments around the clap index and compute the cross-correlation between them to find the time differences between signals.

Given the noisiness of the signals, we apply a max pooling 1D to smoothen the signals as a preprocessing step before finding the cross-correlation. The final delay is the mean of the results of cross-correlation between all the audio segments that represent the clap indices. The equations used are as follows:

$$R_{xy}(n) = \sum_{m=-\infty}^{\infty} x(m)y(m+n) \tag{3.9}$$

Where $R_{xy}(n)$ is the cross-correlation of camera signals x and y, and n is the lag.

$$\Delta t = \frac{1}{N} \sum_{i=1}^{N} R_{xy}(n_i) \tag{3.10}$$

Where Δt is the mean delay, N is the number of audio segments, and $R_{xy}(n_i)$ is the cross-correlation result for each segment.

After calculating the delay for each camera using the minimum delay as a reference, this delay information is taken as input to FFmpeg to extract the precise millisecond-accurate synchronization of the recorded videos. The delay for each camera is adjusted as follows:

$$\Delta t_{\text{adjusted}} = \Delta t - \min(\Delta t) \tag{3.11}$$

Where $\Delta t_{\text{adjusted}}$ is the adjusted delay for each camera.

Although we are able to determine the time delay for each camera, achieving millisecond-level synchronization is challenging due to missing frames. One potential solution is to interpolate frames to account for these discrepancies, but this approach requires further research to ensure high-quality frame interpolation. Consequently, we store the delay information for future synchronization improvements.

3.4 Data Encoding

After obtaining the start time and duration for each camera recording, the next step is to encode the raw videos using these parameters. The RAW videos from the cameras are stored as Material Exchange Format (MXF) files, which are ambiguous. To manage these files, we use `FFmpeg` to encode the raw videos into both `mp4` video files for easy distribution and to extract raw frames as `png` files for research purposes.

In digital video production, the choice of data formats and compression techniques significantly impacts the quality, file size, and processing requirements of the final video output. Digital video production utilizes several data formats, each with its own advantages and limitations. Two prevalent formats are RAW and H.264.

RAW format preserves the unprocessed sensor data, providing the highest possible quality. This format captures all the details without any compression artifacts, making it ideal for post-production, where extensive color grading and effects are applied. The file size of RAW is significantly larger compared to compressed formats, as RAW files contain all the data captured by the camera's sensor. Additionally, RAW files require substantial processing power and time during post-production, and they necessitate specialized software for decoding and editing.

H.264, also known as AVC (Advanced Video Coding), is a widely used format that balances quality and file size through efficient compression techniques. While it may not match the quality of RAW, it provides high-quality video suitable for most production needs. The compression significantly reduces file size, making H.264 more manageable for

storage and transfer. Additionally, H.264 files are easier to handle and edit compared to RAW, requiring less computational power and storage space.

For our video encoding, we use the `libx264` encoder to create H.264 video streams due to its efficiency and high-quality video compression at relatively low bitrates. `libx264` is a software library and the most widely used encoder for creating H.264/AVC (Advanced Video Coding) video streams. It is known for its high efficiency and ability to produce high-quality video at relatively low bitrates. `libx264` produces high-quality video at lower bitrates compared to older codecs, which helps save storage space and bandwidth. H.264 is supported by almost all modern video players, devices, and streaming platforms, ensuring broad compatibility. The library offers a wide range of settings and parameters to fine-tune the encoding process for specific needs, such as balancing quality and file size or optimizing for specific hardware. Additionally, `libx264` supports different H.264 profiles (Baseline, Main, High), allowing for compatibility with various devices, from low-power mobile devices to high-definition broadcasting systems.

There are two primary methods for encoding the videos: setting the constant rate factor (CRF) or the bitrate. CRF is used to control the quality of the video output in a variable bitrate (VBR) encoding scheme. It adjusts the bitrate dynamically based on the complexity of the video to maintain a consistent quality level. In FFmpeg, CRF values typically range from 0 to 51. Lower values indicate higher quality and larger file size, with CRF 0 meaning lossless compression. Higher values indicate lower quality and smaller file size, with CRF 51 being the lowest quality. For good quality, a CRF between 18 and 28 is usually used, with CRF 23 often considered a balanced setting.

Using the bitrate focuses on maintaining a consistent bitrate, which may result in variable quality, especially when dealing with complex scenes or motion at low bitrates. Conversely, a high bitrate can lead to a wastage of bits on simple scenes. Bitrate specifies the amount of data used to encode the video per second. It can be set as a constant bitrate (CBR) or as an average bitrate (ABR) in a variable bitrate scheme. CBR maintains the bitrate constant throughout the video, ensuring a predictable file size but potentially wasting bits on simple scenes or lacking enough bits for complex scenes. ABR allows for variability, aiming to achieve a target average bitrate over the entire video, providing a balance between quality and file size. Bitrate is usually measured in kilobits per second (kbps) or megabits per second (Mbps).

We chose to use CRF over bitrate because CRF controls video quality with an indirect influence on file size, ensuring a consistent quality level, which is crucial for our purposes and maintaining the visual fidelity of our recordings. Bitrate settings provide a predictable file size but may result in variable quality, which is not ideal for our needs.

We tested our studio recordings on CRF values ranging from 10 to 28 and concluded that a value of 20 provided the best balance between quality and size (see Figure 3.7).

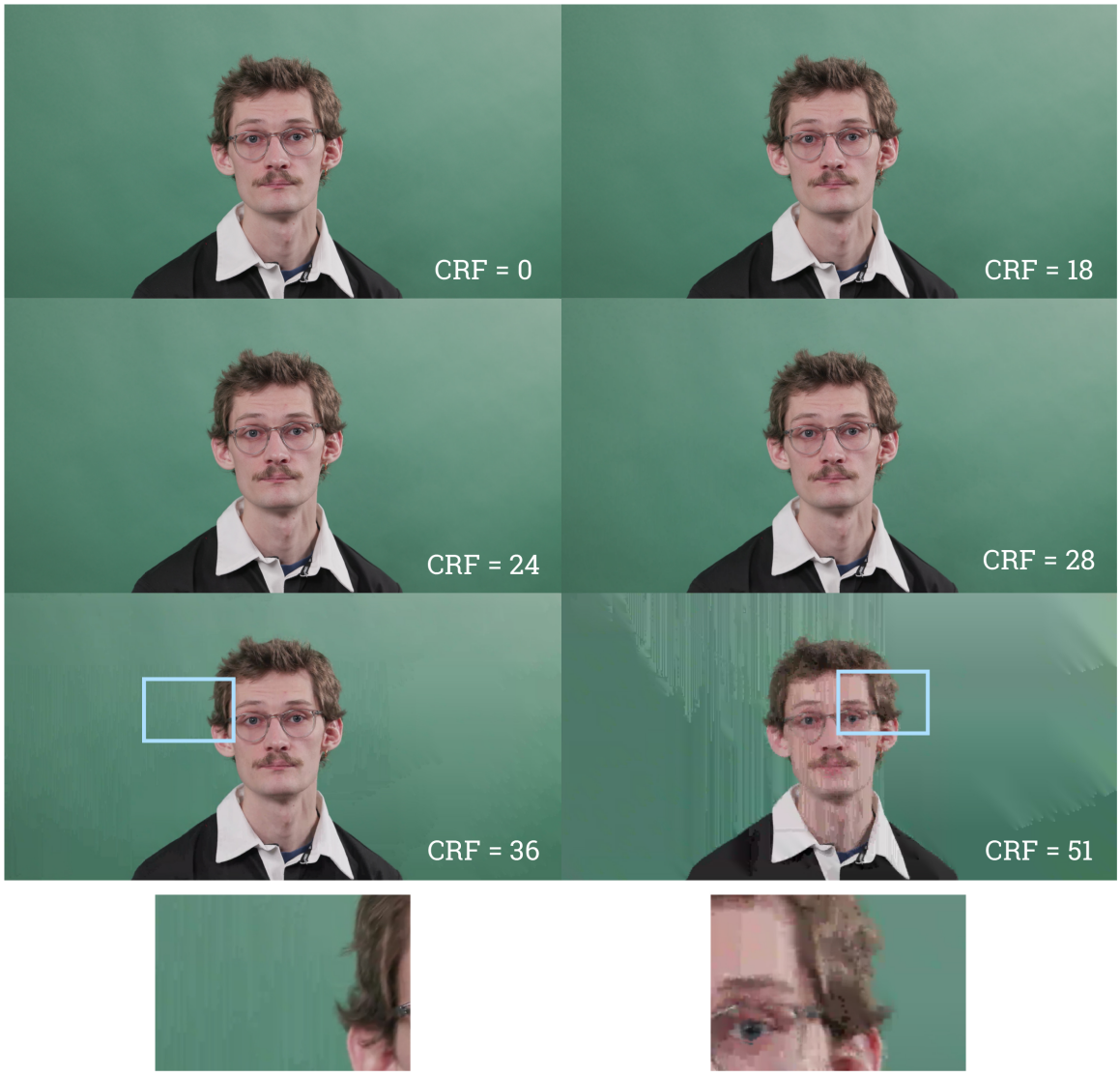


Figure 3.7: Comparison of video quality and file size across different CRF values. As the CRF value approaches 51, compression artifacts become more noticeable.

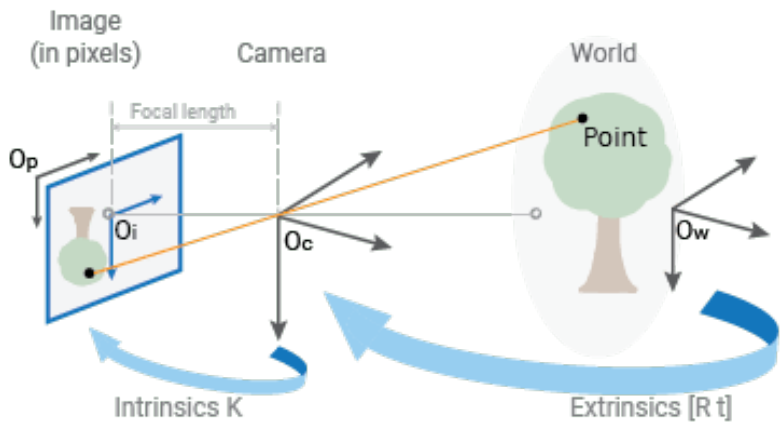


Figure 3.8: Mapping from world coordinates to image coordinates. The world points are first transformed into the camera's coordinate system using the extrinsic parameters. Subsequently, the camera coordinates are projected onto the image plane using the intrinsic parameters [41].

We also added an option to save frames as `png` images for image analysis. For frame extraction, we extract all frames for each camera recording using the `png` codec with `rgb48` pixel format. This ensures that the extracted frames are 16-bit raw `png` files (i.e., $48/3 = 16$ for 3 channels).

3.5 Camera Calibration

Camera calibration is a fundamental process in computer vision that is essential for accurately interpreting how a camera captures the world. The calibration process involves determining a set of parameters that describe the camera's imaging geometry. These parameters are typically divided into two main categories: intrinsic and extrinsic.

The intrinsic parameters are related to the internal characteristics of the camera, such as the focal length, optical center or principal point, and lens distortion coefficients. These parameters define how the camera lens projects 3D world points onto the 2D image plane. In contrast, the extrinsic parameters describe the camera's position and orientation in space relative to the world coordinate system. They define the transformation from the world coordinate system to the camera coordinate system, allowing the mapping of real-world objects into the camera's field of view.

The calibration process involves estimating these parameters by analyzing images of a known calibration pattern, such as a checkerboard. Accurate calibration is crucial for a wide range of applications, including 3D reconstruction, motion capture, and scene understanding.

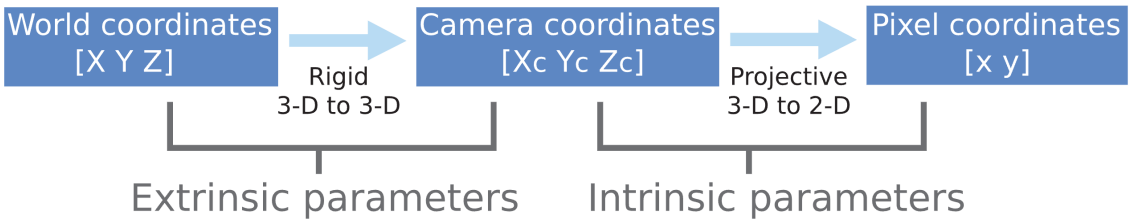


Figure 3.9: Mapping from 3D world coordinates to camera coordinates to pixel coordinates [41].

3.5.1 Camera Matrix

In computer vision, the camera matrix or projection matrix describes the mapping of a pinhole camera from 3D points in the world to 2D points in an image. This process is typically expressed by the equation:

$$\mathbf{x} = \mathbf{P}\mathbf{X} \tag{3.12}$$

where $\mathbf{X} = [X, Y, Z, 1]^T$ represents the homogeneous coordinates of a 3D point in the world, $\mathbf{x} = [x, y, 1]^T$ denotes the corresponding 2D point in the image, and $\mathbf{P}$ is the 3×4 camera projection matrix.

The camera matrix $\mathbf{P}$ is a product of the intrinsic matrix $\mathbf{K}$ and the extrinsic parameters $[\mathbf{R}|\mathbf{t}]$, which encapsulate the rotation matrix $\mathbf{R}$ and the translation vector $\mathbf{t}$. This relationship is mathematically represented as:

$$\mathbf{P} = \mathbf{K}[\mathbf{R} \mid \mathbf{t}] \tag{3.13}$$

The combination of the intrinsic and extrinsic parameters in the camera matrix $\mathbf{P}$ allows us to project a 3D point in the world to a 2D point in the image. This process involves transforming the 3D world coordinates $\mathbf{X}$ into camera coordinates using the extrinsic parameters, followed by a mapping from camera coordinates to the image plane using the intrinsic matrix.

This projection from 3D to 2D is an example of forward projection, where the goal is to determine the image position of a point given its position in 3D space. Conversely, backward projection involves estimating the 3D coordinates of a point given its 2D image coordinates and knowledge of the camera matrix $\mathbf{P}$.

While most vision problems focus on deriving backward projection equations to recover 3D scene structure from images, in other to understand the camera matrix, it is important to first understand forward projection, which describes the process by which 3D world points are projected onto a 2D image plane.

3.5.2 Forward Imaging Model: From 3D to 2D

If we consider a basic perspective projection of the pinhole camera, where a 3D point $\mathbf{X} = (X, Y, Z)$ in the world coordinate is projected onto a 2D image plane at a point $\mathbf{x} = (x, y)$. The center of projection, often denoted as the camera's optical center, is the point from which all light rays emanate before intersecting the image plane. In the figure below, the projection process is illustrated, showing the relationship between the 3D world point, the image plane, and the center of projection.

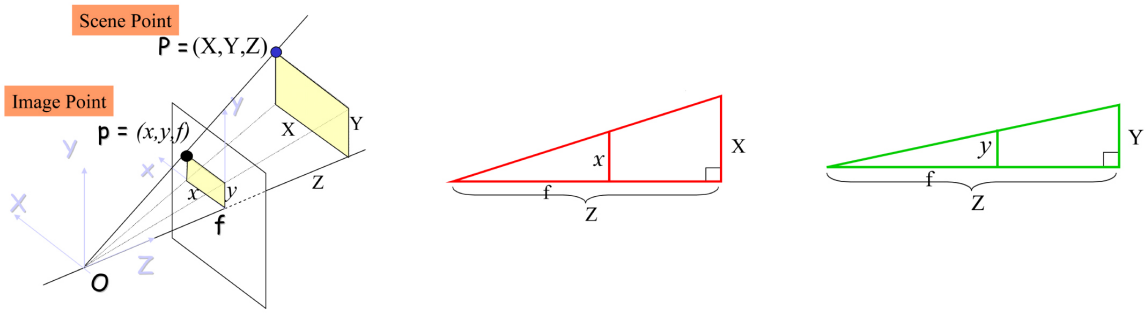


Figure 3.10: Basic perspective projection of a 3D world point onto a 2D image plane, illustrating the center of projection **O** [8].

Using the rule of similar triangles, we can derive the perspective projection equations. Consider the triangles formed by the 3D point, its projection onto the image plane, and the center of projection. By similar triangles, the following relationships hold:

$$x = \frac{fX}{Z}, \quad y = \frac{fY}{Z} \tag{3.14}$$

where f is the focal length of the camera, X and Y are the coordinates of the 3D point in the world, and Z is its depth relative to the camera. These equations describe how a 3D point in space is mapped onto a 2D image plane through perspective projection.

To further generalize this model, we introduce homogeneous coordinates. In homogeneous coordinates, a 2D point $\mathbf{x} = (x, y)$ in the image plane can be represented as a 3D point $\mathbf{x_h} = (x, y, w)$ by adding a fictitious third coordinate w. This third coordinate allows us to represent the scaling effect inherent in perspective projection and facilitates the conversion of the perspective equations into a matrix form. We can recover the original 2D point $\mathbf{x}$ by dividing by the third coordinate w, as follows:

$$\mathbf{x} = \left(\frac{x_h}{w}, \frac{y_h}{w} \right)$$

Given the perspective projection equations $x = \frac{fX}{Z}$ and $y = \frac{fY}{Z}$, we can express these equations in homogeneous coordinates as:

$$\mathbf{x_h} = \frac{1}{Z} \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \mathbf{X_h} \tag{3.15}$$

Alternatively, this can be written using the intrinsic matrix $\mathbf{K}$ as:

$$\mathbf{x_h} = \frac{1}{Z} \mathbf{K} \begin{bmatrix} \mathbf{I} & \mathbf{0} \end{bmatrix} \mathbf{X_h} \tag{3.16}$$

where

$$\mathbf{x_h} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}, \quad \mathbf{X_h} = \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Here, the matrix represents the perspective projection, mapping the 3D world point $\mathbf{X} = (X, Y, Z)$ to its 2D image coordinates $\mathbf{x} = (x, y)$, with $\mathbf{K}$ known as the intrinsic matrix. The identity matrix $\mathbf{I}$ separates the intrinsic parameters from the extrinsic parameters (which are not included in this particular formulation), indicating that the extrinsic parameters are arbitrary in this case and can be applied separately depending on the specific camera setup. The factor $\frac{1}{Z}$ accounts for the scaling effect due to the perspective projection, where points further away from the camera (larger Z) appear closer together in the image, thus compressing the coordinates.

To account for the camera's internal characteristics, such as the optical center, we can refine the intrinsic matrix to include these parameters. The final form of the intrinsic matrix $\mathbf{K}$ incorporates the principal point (c_x, c_y), leading to the following:

$$\mathbf{K} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \tag{3.17}$$

where f_x and f_y are the effective focal lengths in the x and y directions, respectively.

3.5.3 Forward Imaging Model: From 3D to 3D

In the forward imaging model, the transformation from a 3D point in the world coordinate system to the 3D camera coordinate system is a crucial step before projecting onto the 2D image plane. This transformation is governed by the camera's extrinsic parameters, which include the rotation and translation matrices. These matrices define how the world coordinates are mapped to the camera's coordinate system.

Consider a 3D point $\mathbf{P_W} = (X_W, Y_W, Z_W)$ in the world coordinate system. The same point in the camera coordinate system is denoted as $\mathbf{P_C} = (X_C, Y_C, Z_C)$. These two

representations describe the same physical point but in different coordinate systems. The relationship between the world coordinates and the camera coordinates is defined by a change of basis, represented by the extrinsic parameters of the camera.

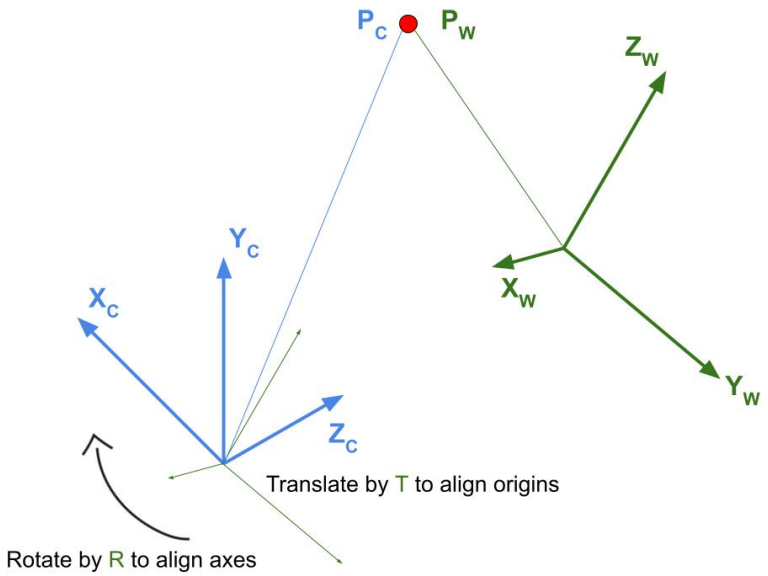


Figure 3.11: Transformation of a 3D point from world coordinates $\mathbf{P_W} = (X_W, Y_W, Z_W)$ to camera coordinates $\mathbf{P_C} = (X_C, Y_C, Z_C)$ using the rotation matrix $\mathbf{R}$ and translation vector $\mathbf{t}$.

The transformation from the world coordinate system to the camera coordinate system can be expressed mathematically as:

$$\mathbf{P_C} = \mathbf{R}\mathbf{P_W} + \mathbf{t} \tag{3.18}$$

where $\mathbf{R}$ is a 3×3 rotation matrix that describes the orientation of the camera relative to the world coordinate system, and $\mathbf{t}$ is a 3×1 translation vector that describes the position of the camera in the world coordinate system.

The rotation matrix $\mathbf{R}$ is responsible for rotating the world coordinates into the camera's coordinate frame. This matrix encodes the angular orientation of the camera, which can be described by the angles of rotation around the three principal axes (pitch, yaw, and roll). The translation vector $\mathbf{t}$, on the other hand, shifts the origin of the world coordinate system to the origin of the camera coordinate system, effectively translating the point in space.

To derive this transformation, consider the figure above, which illustrates the relationship between the world coordinates and the camera coordinates. The point $\mathbf{P_W} = (X_W, Y_W, Z_W)$ in the world coordinate system is first rotated by the matrix $\mathbf{R}$ to align with the camera's orientation. This rotated point is then translated by the vector $\mathbf{t}$ to account for the camera's

position, resulting in the corresponding point $\mathbf{P_C} = (X_C, Y_C, Z_C)$ in the camera coordinate system.

The combined transformation can be succinctly represented in homogeneous coordinates as:

$$\begin{bmatrix} X_C \\ Y_C \\ Z_C \\ W_C \end{bmatrix} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} X_W \\ Y_W \\ Z_W \\ 1 \end{bmatrix} \tag{3.19}$$

This matrix equation allows for a straightforward conversion from world coordinates to camera coordinates, making it possible to perform operations in the camera's reference frame, which is essential for subsequent image projection processes.

3.5.4 Forward Imaging Model: Combining Intrinsics and Extrinsics

Having established the separate roles of intrinsic and extrinsic parameters in the camera model, we now combine these components to define the complete forward imaging model.

This model describes how a 3D point in the world coordinate system is transformed and projected onto the 2D image plane through a series of linear transformations. The extrinsic parameters transform the coordinates of a 3D point $\mathbf{P_W} = (X_W, Y_W, Z_W)$ from the world coordinate system to the camera coordinate system $\mathbf{P_C} = (X_C, Y_C, Z_C)$. Once the point is in the camera coordinate system, the intrinsic parameters come into play, projecting the 3D camera coordinates onto the 2D image plane.

The complete forward imaging model is derived by combining the extrinsic and intrinsic parameters into a single operation. Now that we know the extrinsic parameters, continuing from equation 3.15, we can replace the identity matrix $\mathbf{I}$ with the extrinsic parameters.

$$\mathbf{x_h} = \frac{1}{Z} \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \mathbf{X_h} \tag{3.20}$$

Substituting the intrinsic matrix with $\mathbf{K}$ and the extrinsic matrix with $[\mathbf{R} \mid \mathbf{t}]$:

$$\mathbf{x_h} = \frac{1}{Z}\mathbf{K}[\mathbf{R} \mid \mathbf{t}]\mathbf{X_h} \tag{3.21}$$

Substituting $\mathbf{P}$ into the equation for projecting a world point onto the image plane, we obtain:

$$\mathbf{x} = \mathbf{P}\mathbf{X_w} = \mathbf{K}[\mathbf{R} \mid \mathbf{t}]\mathbf{X_w} \tag{3.22}$$

Now that we understand camera calibration and parameters, we will discuss how we obtain the precise parameters of the cameras in our studio's multi-camera setup.

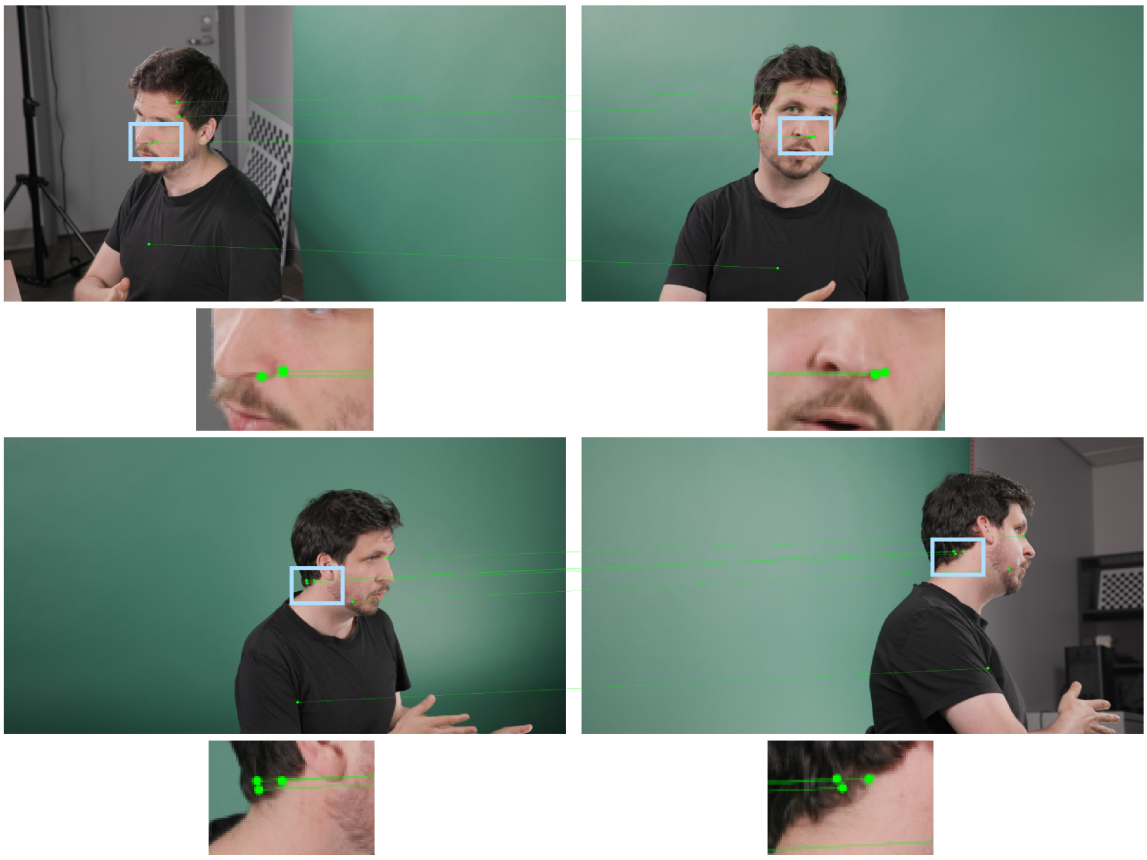


Figure 3.12: Example of sparse and imprecise feature matching in our studio. The lack of distinct features results in sparse matches.

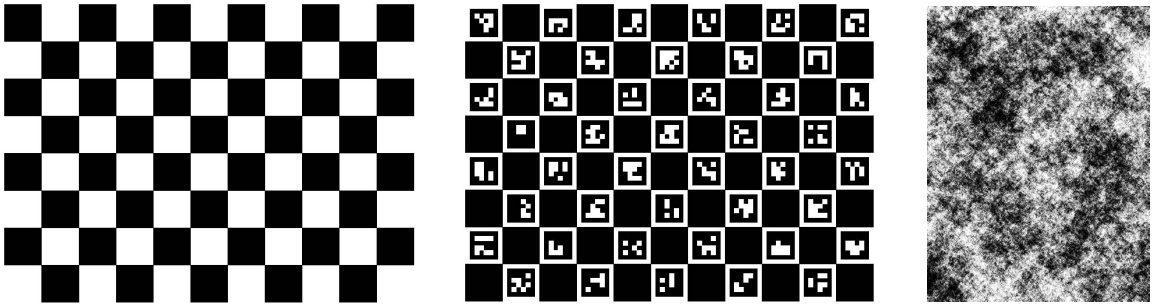
3.6 Studio Multi-camera Calibration

Calibration methods can be broadly classified into two categories: those that use a pattern of known geometry and those that rely on feature-matching points across the images captured by different cameras. Each of these methods has its strengths and weaknesses, and the choice of method depends on the specific requirements of the project.

Multi-camera calibration using feature matching points involves identifying corresponding points across different images based on their appearance. This method can be quite effective in scenes with rich textures, where distinct features can be easily matched. However, this approach often struggles in scenes with large texture-less areas. A typical example is our studio with a large green screen in the background, where there are little to no distinct features to match. Even when features are matched, the accuracy is often not sufficient for high-resolution images, where pixel-level precision is necessary, such as the one generated from our studio. Figure 3.12 illustrates this issue, where the matched feature points are sparse and are also not pixel-perfectly aligned. The matched features are obtained using the state-of-the-art feature matching algorithm [37].

Given the limitations of feature matching, we opted to use a calibration method based on a pattern of known geometry. This approach allows for much more precise alignment, as the geometric properties of the pattern are known and can be used to accurately solve for the camera parameters.

3.6.1 Calibration Using Pattern of Known Geometry



(a) Checkerboard Pattern (b) ChArUco Pattern (c) Random Pattern

Figure 3.13: Some types of well-defined calibration patterns with known geometry.

Calibration using patterns of known geometry involves capturing images of a well-defined pattern, such as a checkerboard, from multiple camera angles. The known dimensions and layout of the pattern allow for the accurate determination of both intrinsic and extrinsic parameters. Various patterns can be used for this purpose, including checkerboards, charuco boards, and random patterns. Figure 3.13 shows these three types of patterns.

Each type of pattern has its advantages. Checkerboard patterns, for instance, are widely used due to their simplicity and the ease with which corners can be detected. Charuco boards combine the benefits of checkerboards and AprilTags, providing high accuracy even in cases where some parts of the board are occluded or poorly illuminated. Random patterns, introduced by Li et al. [25], offer robustness against repetitive texture problems but can be more challenging to detect and process. In our setup, we chose to use a checkerboard pattern as it offers a good balance between ease of detection and accuracy.

Our calibration toolbox relies on images taken from checkerboard patterns to solve for the intrinsic and extrinsic parameters of each camera. The calibration process consists of three main stages:

Checkerboard Points Detection In the first stage, we perform accurate corner detection on the checkerboard images to detect the 2D points. Since the entire optimization relies on these detected corners, we employ a robust multi-scale approach to detect corners with sub-pixel accuracy.

Single Camera Calibration In the second stage, we use the patterns seen by each camera to optimize its intrinsic parameters. This involves adjusting parameters such as focal length, principal point, and lens distortion to best fit the observed data.

Multi-Camera Calibration Finally, we compute the poses of each camera in a reference global coordinate system. This is done by using the patterns that are seen by multiple cameras (overlapping patterns) in a bundle adjustment setup. The bundle adjustment refines both the intrinsic and extrinsic parameters by minimizing the reprojection error across all cameras.

3.6.2 Checkerboard Points Detection

To detect the corners in the checkerboard pattern, our pipeline expects the total number of patterns, their grid shape, and the metric size of each block as input. Next, we run our corner detection on every image in the calibration sequence to detect a pattern matching one of the pattern configurations provided in the input.

To avoid long computation time, we propose to estimate a rough corner location in a multi-scale manner first and then compute the accurate sub-pixel values in the second step.

Given a calibration image, we downscale it to $\frac{1}{16}$ resolution and employ the built-in checkerboard corner detection algorithm provided by OpenCV [4]. Note that we take the expected grid shape of the patterns in the input. In case no matching pattern was found, we increased the resolution by a factor of 2 and rerun the process. We continue this process till $\frac{1}{2}$ of the original image resolution. In our experiments, our calibration sequence is captured at 4K. We avoid processing at their original high resolution as we only need a rough corner

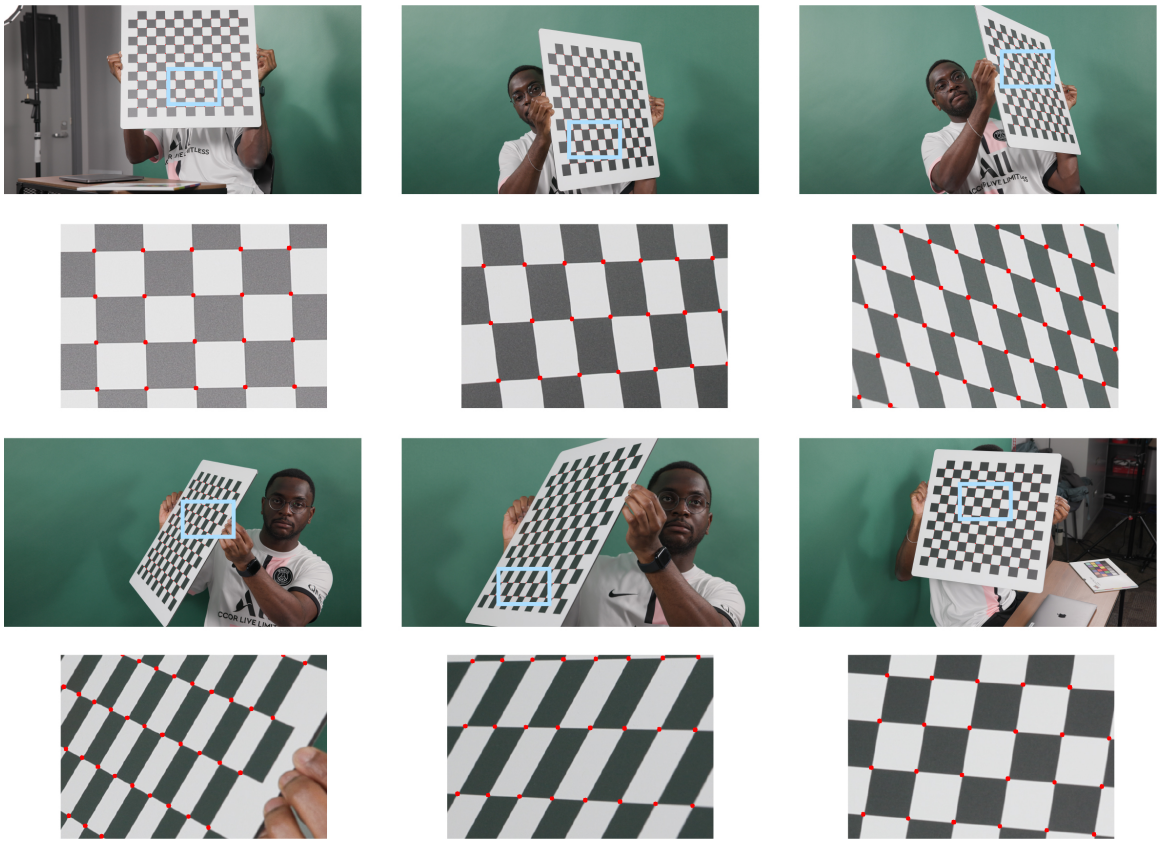


Figure 3.14: Detected checkerboard points at the high-resolution images.

location at this stage. If no corner is found at this resolution, we discard the image from the calibration sequence.

Given the rough corner location, we find the accurate corner locations with sub-pixel accuracy following Förstner *et al.* [10]. We set the search window size in their algorithm according to the scale on which the rough corner location is found.

With the known grid shape and metric size of each block, we can compute the 3D position of each corner point as well. The pairs of 2D and 3D coordinates will be used in the calibration process as described in Section 3.6.3 and 3.6.4.

3.6.3 Single Camera Calibration

The single-camera calibration process involves estimating the intrinsic parameters of each camera and the pose of the checkerboard patterns as seen by the camera.

We perform a non-linear optimization using the Levenberg-Marquardt [47] algorithm to optimize the camera parameters by minimizing the re-projection error, which is the distance between the observed 2D points and the projected 3D points using the current estimate of the camera parameters. This iterative process continues until the re-projection error is minimized to an acceptable level.

For practical implementation, we use the available API in OpenCV that can perform this calibration process, such as the `cv2.calibrateCamera` function.

The result of this calibration process is a set of camera parameters that include the intrinsic matrix $\mathbf{K}$ (recall $\mathbf{K}$ from Section 3.5.1), which encompasses the focal lengths and optical center. In addition to the intrinsic parameters, real-world cameras also require the estimation of lens distortion, which needs to be corrected. The distortion coefficients $\mathbf{D}$ account for these distortions and typically include radial and tangential distortions.

The distortion coefficients are defined as:

$$\mathbf{D} = [k_1, k_2, p_1, p_2, k_3], \tag{3.23}$$

where k_1, k_2, k_3 are the radial distortion coefficients, and p_1, p_2 are the tangential distortion coefficients.

Radial distortion causes straight lines to appear curved, and it is modeled as follows:

$$x_{\text{distorted}} = x(1 + k_1 r^2 + k_2 r^4 + k_3 r^6), \tag{3.24}$$

$$y_{\text{distorted}} = y(1 + k_1 r^2 + k_2 r^4 + k_3 r^6), \tag{3.25}$$

where r is the radial distance from the center of the image, defined as $r^2 = x^2 + y^2$.

Tangential distortion occurs when the lens and the image plane are not perfectly parallel. It is modeled as follows:

$$x_{\text{distorted}} = x + [2p_1xy + p_2(r^2 + 2x^2)], \tag{3.26}$$

$$y_{\text{distorted}} = y + [p_1(r^2 + 2y^2) + 2p_2xy]. \tag{3.27}$$

Additionally, the calibration process estimates the pose of the checkerboard patterns as seen by the camera.

The reprojection error $\mathcal{E}$ is minimized using the following objective function:

$$\mathcal{E} = \sum_{i=1}^{N} \|\mathbf{m}_i - \pi(\mathbf{M}_i, \mathbf{K}, \mathbf{R}, \mathbf{t}, \mathbf{D})\|^2, \tag{3.28}$$

where N is the number of points, $\mathbf{m}_i$ are the observed 2D points, $\mathbf{M}_i$ are the corresponding 3D points, $\pi(\cdot)$ is the projection function, and $\mathbf{R}$ and $\mathbf{t}$ are the rotation and translation vectors of the patterns, respectively.

The projection function $\pi(\cdot)$ maps a 3D point $\mathbf{M}$ to a 2D point $\mathbf{m}$ using the intrinsic parameters and distortion coefficients:

$$\mathbf{m} = \pi(\mathbf{M}, \mathbf{K}, \mathbf{R}, \mathbf{t}, \mathbf{D}). \tag{3.29}$$

This process iterates until the re-projection error is minimized to an acceptable level.

3.6.4 Multi-Camera Calibration

In the previous step we computed the intrinsic camera parameters and also the position of each pattern in each camera's coordinate system. The aim of the multi-camera calibration is to obtain each camera's metric pose with respect to a reference global coordinate.

We assume that cameras are synchronized so that they capture images of the patterns at the same time. This means that the relative poses of the pattern with respect to each camera are known via overlapping views.

The overlapping views (seeing the same pattern) allow us to compute the relationship between the poses of the two cameras that see the same pattern. Following Li *et al.* [25] we create a pose graph to represent the relationship between cameras and patterns. The graph's vertices are the camera and pattern objects, while the edges represent the relationship between them.

Li *et al.* [25] select a random camera as the root node of the graph, which represents the reference global coordinate. We define the root node of the graph as the camera that saw the most patterns. This camera's pose is set to identity. To obtain the initial poses for other cameras and patterns, we start traversing from the root node and transform the pose of each pattern connected to that camera to the global coordinate using the equation below.

$$\mathbf{P}_{\mathbf{g}_{pat_j}} = \mathbf{P}_{\mathbf{g}_{cam_i}}^{-1} \mathbf{P}_{\mathbf{pat_j}}^{\mathbf{cam_i}} \tag{3.30}$$

where $\mathbf{P}_{\mathbf{g}_{pat_i}}$ represents the pose of the pattern i in the global coordinate system, $\mathbf{P}_{\mathbf{g}_{cam_i}}$ represents the pose of the camera i in the global coordinate system, and $\mathbf{P}_{\mathbf{pat_i}}^{\mathbf{cam_i}}$ represents the pose of the pattern with respect to the camera as computed in the previous step.

For each transformed pattern, we find all other cameras connected to this pattern that have not yet been visited. We then compute the pose of each of these cameras using the equation below:

$$\mathbf{P}_{\mathbf{g}_{cam_i}} = \mathbf{P}_{\mathbf{pat_j}}^{\mathbf{cam_i}} \mathbf{P}_{\mathbf{g}_{pat_j}}^{-1} \tag{3.31}$$

By traversing all cameras, we can determine the initial estimates of each camera's pose with respect to the global reference coordinate system.

Finally, we perform a non-linear optimization using the Trust Region Filter (TRF) [14] algorithm to optimize the camera parameters by minimizing the sum of all reprojection errors. We discovered that optimizing both intrinsic and extrinsic camera parameters directly from the initial poses caused very slow convergence and incorrect final camera parameters. As a result, following [51], we first optimized only the extrinsic parameters while keeping the intrinsic parameters fixed. In a final refinement step, we then optimized both the intrinsic and extrinsic parameters of the system altogether.

The optimization problem with only the extrinsic parameters is defined as:

$$\underset{\mathbf{P}_{\mathbf{g}_{cam_i}}, i \neq \text{ref}}{\arg\min} \sum_{i,j} \sum_{k} \| \mathbf{p}_k^{img_j} - \mathbf{p}^{\text{reproj}} \left(\mathbf{P}_{\mathbf{g}_{cam_i}} \mathbf{P}_{\mathbf{g}_{pat_j}} \mathbf{p}_k^{pat_i}, \mathbf{K}_i \right) \|^2 \tag{3.32}$$

while the optimization problem with both the intrinsic and extrinsic parameters is:

$$\underset{\mathbf{P}_{\mathbf{g}_{cam_i}}, \mathbf{K}_{ref}, \mathbf{K}_i, i \neq \text{ref}}{\arg\min} \sum_{i,j} \sum_{k} \| \mathbf{p}_k^{img_j} - \mathbf{p}^{\text{reproj}} \left(\mathbf{P}_{\mathbf{g}_{cam_i}} \mathbf{P}_{\mathbf{g}_{pat_j}} \mathbf{p}_k^{pat_i}, \mathbf{K}_i \right) \|^2 \tag{3.33}$$

Here, $\mathbf{K}_i$ represents the intrinsic parameters of camera i, $\mathbf{p}_k^{img_j}$ denotes the observed image points in image j, $\mathbf{p}_k^{pat_i}$ denotes the pattern points in pattern i, and $\mathbf{p}^{\text{reproj}}$ is the reprojected point calculated based on the estimated parameters. The optimization is over $\mathbf{P}_{\mathbf{g}_{cam_i}}$ with $i \neq$ ref, where the reference frame is set to the camera that observed the most patterns.

3.6.5 Analysis

In our initial calibration process, we utilized the medium calibration pattern (12×11) due to its optimal balance between the number of calibration points and its visibility within the camera frames. This pattern provided more calibration points than the smaller pattern (9×8) while ensuring that the entire checkerboard remained visible, unlike the larger pattern

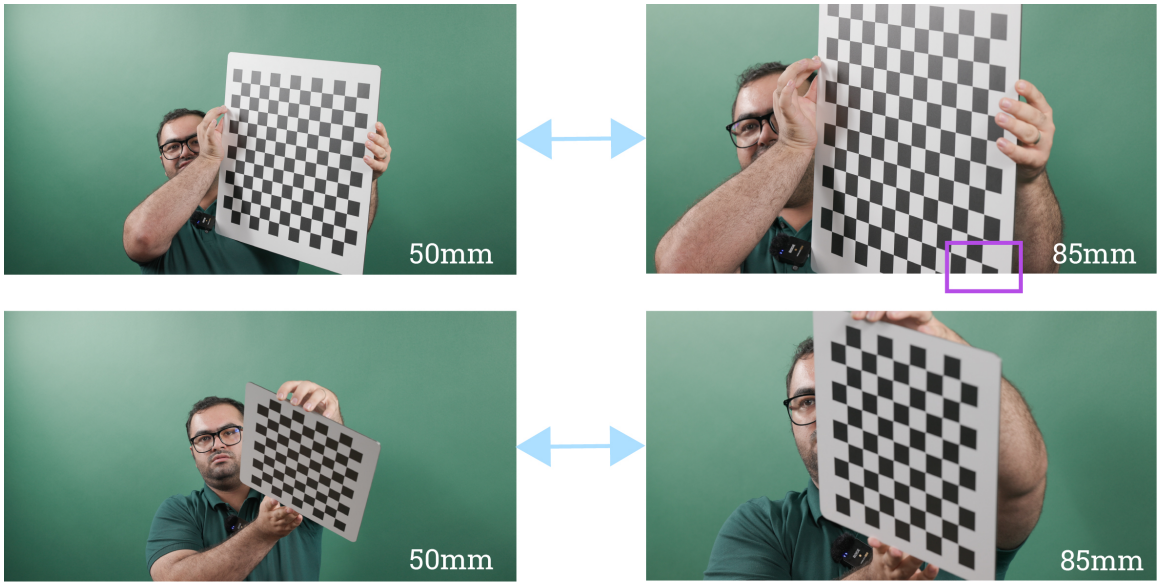


Figure 3.15: Illustration of the difficulties in fitting the medium (or large) pattern in camera frame with a narrow field of view. This also makes it hard to have overlapping patterns within neighboring cameras as shown (represented by rows).

(14×13), which often extended beyond the field of view due to the varying focal lengths in our setup.

During calibration, we captured nine checkerboard orientations per camera, anticipating that the overlapping fields of view among cameras would increase the number of images with detected checkerboard points. This was particularly expected for cameras with wide fields of view, which could capture the pattern even when it was being shown to neighboring cameras, and for cameras positioned close to each other. In our experiments, the number of images with detected checkerboard points for these cameras ranged between 25 and 30. However, cameras at the extremities of the setup or those with narrow fields of view detected fewer frames, often between 6 and 10, leading to a limited number of detected checkerboard points and impacting the reprojection error. This also increased the risk of a camera becoming disconnected during bundle adjustment, potentially resulting in its exclusion from the system.

To address this, we integrated a smaller pattern into our calibration process despite knowing that it is inherently less accurate due to its lower resolution—fewer pixels per square and detectable pattern points—compared to the medium pattern. Our primary objective was to avoid losing any cameras from the system, recognizing that the trade-off would be a slight increase in the overall reprojection error.

Figure 3.15 demonstrates the limitations of the medium (or large) patterns within zoomed-in lenses, underscoring the necessity of incorporating the smaller pattern. Table 3.1 presents a comparison of the number of detected patterns and the associated reprojection

errors under different configurations: medium pattern only, small pattern only, half medium and half small patterns, and full medium combined with full small patterns.

Table 3.1: Comparison of Detected Patterns and Reprojection Errors under Various Configurations

Camera	**Medium Pattern**	**Small Pattern**	**Medium + Small**
Camera 1	19	35	54
Camera 2	24	30	54
Camera 3	12	24	36
Camera 4	39	49	88
Camera 5	41	51	92
Camera 6	46	49	95
Camera 7	25	28	53
Camera 8	17	26	43
Total Detected Patterns	94	98	192
Reprojection Error	**1.96**	**2.89**	**2.01**

The results indicate that while the inclusion of the small pattern did increase the number of detected checkerboard points, especially for cameras at the periphery, it also introduced a small trade-off in terms of reprojection error. For instance, with the medium pattern alone, the reprojection error was 1.96 for 94 patterns, whereas with the small pattern alone, the error increased to 2.89 for 98 patterns. The combined use of medium and small patterns resulted in a reprojection error of 2.01 for a total of 192 patterns.

The decision to incorporate the smaller pattern was driven by the need to ensure comprehensive camera coverage and connectivity during calibration, even at the expense of a minor increase in reprojection error. This approach results in a more robust calibration process, minimizing the risk of camera exclusion and ensuring a reliable overall system calibration.

Chapter 4

Automatic Green Screen Keying

Figure 4.1: We achieve high resolution green screen keying with our method compared to other background matting methods.

In our studio pipeline, we record our subject in front of a green screen so that we can isolate (key) the subject for production purposes. Existing commercial green screen keying solutions like The Foundry's Keylight and IBK, Blackmagic Design Ultimatte, or Boris Continuum Primatte offer sophisticated tools with a high level of flexibility. However, they require experience and the large amount of manual work that the artist is left with - to fix the keying errors with inpainting and rotoscoping - create a demand for a reliable automated keying solution. In this chapter, we provide a detailed exploration of our green screen keying methodology, tailored specifically for our studio's setup. We examine the technical challenges we faced in our pursuit of high-quality keying results and outline the strategies and pipelines we developed to address these issues.

4.1 Related Work

Image Matting is a highly underconstrained problem. Removing the background layer without artifacts requires the pixel-accurate estimation of foreground colors (Fg), background colors (Bg), and alpha transparency (α) as seven unknowns [54]. The relationship between each of these is represented by the composite equation 4.1 that forms an image.

$$I = \alpha \cdot \mathrm{Fg} + (1 - \alpha) \cdot \mathrm{Bg}, \tag{4.1}$$

where I is the observed image, α is the alpha matte indicating the transparency level, Fg is the foreground color, and Bg is the background color.

In general, the vast number of possible foreground objects in varying combinations and complex backgrounds poses a significant challenge to the algorithm. A common response to this challenge is the integration of additional inputs that help reduce the solution space [1, 11, 23, 35, 44, 52, 64]. In interactive scenarios, approaches can benefit from user guidance to indicate background and foreground areas in the form of scribbles [7, 11, 22–24] or dense trimaps with an additional uncertainty area [2,18,44,59,64]. Provided the capturing process can be directed accordingly, a clean image of the background can be used as additional input for background matting, guiding the estimation process based on simple pixel differences or higher-level features using neural networks [35, 52].

The success of deep learning for complex prediction tasks has helped improve the accuracy of the input-constrained matting methods, leveraging more diverse user prompts [28] and opened up the possibility for natural image matting approaches without additional inputs [21, 27, 32, 34, 40, 55, 66]. Most of these networks are trained to learn the distribution of possible backgrounds and foregrounds [40, 65, 66], leveraging semantics [21, 55] to reason about their accurate separation, or utilizing pretrained segmentation methods as initial guide [19,28,48]. This, however, requires large, diverse datasets with provided ground truth, which are expensive to generate in high quality. In practice, these methods are hence limited to specific object distributions [21, 34, 62] or fall behind the constrained methods in terms of alpha accuracy.

Extending the matting task to image sequences for effective video matting poses additional challenges for neural networks. While a perfect alpha estimation on a per-image level would be readily adaptable to image sequences, slight inaccuracies during inference can lead to temporal inconsistencies with visible flickering between frames. Video matting frameworks thus rely on temporal losses [21, 36, 56, 57] and specialized, recurrent or transformer-based architectures [18,27,65] that increase temporal coherency. At the same time, the computational overhead results in a reduced capacity for detail reconstruction and lower prediction accuracy.

Our framework, in contrast, focuses on the constrained green-screen keying task in a postproduction environment with controlled studio data. We adjust our pipeline to leverage

the underlying assumptions effectively and reserve network capacity for high-accuracy detail reproduction.

Keying drastically reduces the complexity of the general matting task by assuming a monochromatic, high-contrast image background and controlled lighting. Under these assumptions, the distribution of background colors is bounded effectively. Interactive keying solutions exist for a long time [2, 3, 54] and are well integrated into professional postproduction pipelines.

Despite the high amount of manual labor still necessary, the corpus of automatic keying literature is relatively small [9, 15–17, 20, 26, 53]. The existing approaches mostly focus on adjusting the capturing process in the controlled studio environment to maximize the benefit to the keying task. Grundhöfer and Bimber [15] investigate the use of video projectors projecting rapidly changing colors on the background and extend their concept to active LED screens [16], later followed by [17]. Similarly, Smirnov *et al.* [53] minimize green colors in the foreground by using magenta lighting with a large distance to an active background display. Other approaches use polarized backgrounds [9]. While these methods are fully automatic and greatly reduce workload, they impose large constraints on the capturing setups, making them impractical for real-world scenes or inaccessible for low-budget productions.

In line with works by Jin *et al.* [20] and Li *et al.* [26], we focus our method on the algorithmic optimization of the alpha estimation. We avoid specialized lighting and equipment and reduce manual labor by keeping the required data to a single additional clean plate per sequence.

4.2 Method

We present an approach for automated high-resolution green-screen keying for professional postproduction environments. Our system is centered around two important observations: First, alpha estimation is a highly localized task with existing closed-form solutions for smaller image windows given a trimap [23]. Second, professional movie productions are not unconstrained in-the-wild settings. We can assume controlled lighting and usage of monochromatic screens, significantly limiting the distribution of possible image backgrounds. We can further assume the existence of a captured clean background plate for more complex backgrounds since it is a well-established part of the production pipeline tailored toward existing keying solutions. As an additional constraint, we assume the camera to be static.

Consequentially, we build our framework from three separate stages. We first generate a trimap for a single frame with a clean plate as additional input. This enables an image matting network as our second stage to predict accurate alpha and foreground colors, leveraging a convolutional architecture for localized prediction. Third, we feed the predicted

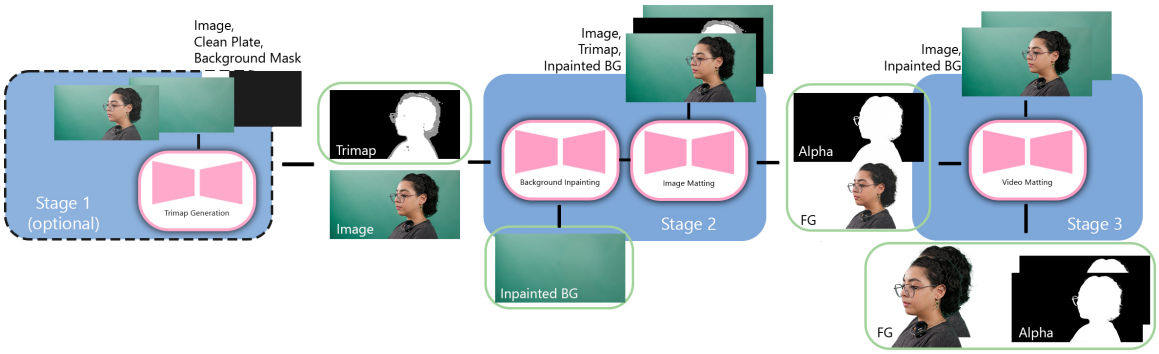


Figure 4.2: We show the three stages of our pipeline. We start from an optional Stage 1, predicting a trimap from an image, a clean plate, and a rough mask. The trimap output and the image serve as inputs to the background inpainting and the image matting in Stage 2. Finally, we predict a consistent alpha and foreground for a frame, given the background, the previous foreground, and the previous alpha. The outputs of our networks are circled in green.

single-frame results to a video matting network, propagating the estimation over the sequence and enforcing temporal consistency. Each stage is carefully designed to maximally simplify the prediction tasks, leaving more network capacity for detail reconstruction. We can thus automate the alpha matting task for image sequences with high accuracy, requiring only a single clean plate as additional input. We show an overview of our pipeline in Fig. 4.2 and detail the individual building blocks in the remainder of this section.

4.2.1 Trimap Generation

Our trimap generator is responsible for separating an image into three regions: foreground, background, and an uncertain area where pixels might contain a mixture of foreground and background colors. Although this task resembles image segmentation, it is less complex because pixel-accurate differentiation is not required. Instead, the model focuses on assigning challenging pixels to the uncertain area without needing to establish a clear boundary for intricate elements like hair or out-of-focus objects.

Leveraging the assumption of an existing clean plate, the task is further simplified as the network can use this additional input to learn basic, low-level similarity concepts for background prediction. This approach is sufficient for simple images where the foreground is distinct from the background, as illustrated in the first row of Figure 4.3.

However, in more complex scenes, determining the foreground becomes challenging, particularly when there are moving objects in the scene. To address this, we introduce an additional input: a rough binary mask, as shown in the second row of Figure 4.3. This mask aids in distinguishing the foreground from the background in such scenarios.

The rough binary mask indicates clear background areas, giving the user the option to manually mask out these complex regions in the background if necessary. This provides

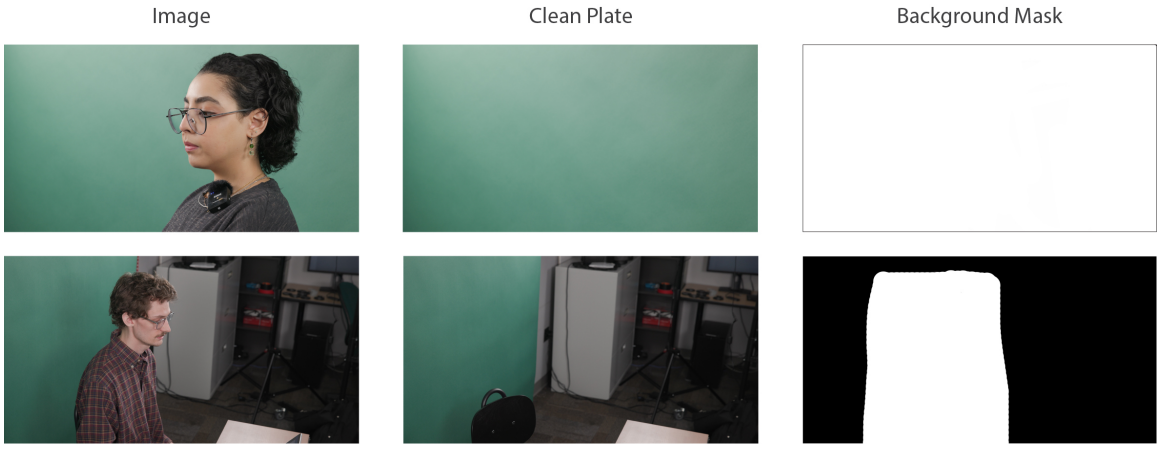


Figure 4.3: Examples of trimap generation inputs. The first row shows a simple scene where the foreground is distinct from the background, making trimap generation straightforward. The second row presents a more complex scene, where a rough binary mask is used to aid in accurate foreground and background separation.

greater flexibility in handling different scene complexities. It is important to note that the mask is optional; for less complex scenes where it is not needed, a simple white canvas image can be used instead.

In line with recent works [62, 68], we formulate trimap estimation as a classification problem. Our model receives the image I, the clean plate, and the background mask as a concatenated $h \times w \times 7$ input and outputs two concatenated feature maps $\hat{M}_{Fg}$ and $\hat{M}_{Bg}$ indicating foreground and background assignments, respectively. These two feature maps can be combined to form a trimap, where the non-foreground and non-background regions are the uncertain regions.

Trimap Network Loss

To supervise the trimap network, we employ a combination of binary cross-entropy (BCE) and mean squared error (MSE) losses, each serving distinct but complementary purposes in the learning process.

Binary Cross-Entropy Loss: Since the trimap network outputs two concatenated feature maps, $\hat{M}_{Fg}$ and $\hat{M}_{Bg}$, to ensure accurate classification of the foreground and background regions, we apply the standard BCE loss to each of these maps individually. The BCE loss effectively penalizes incorrect classifications within each map, thereby enforcing the network to distinguish between the foreground and background regions with high confidence. The BCE loss is defined as:

$$\mathcal{L}_{BCE}^{Fg} = BCE(s(\hat{M}_{Fg}), M_{Fg}), \quad \mathcal{L}_{BCE}^{Bg} = BCE(s(\hat{M}_{Bg}), M_{Bg}), \tag{4.2}$$

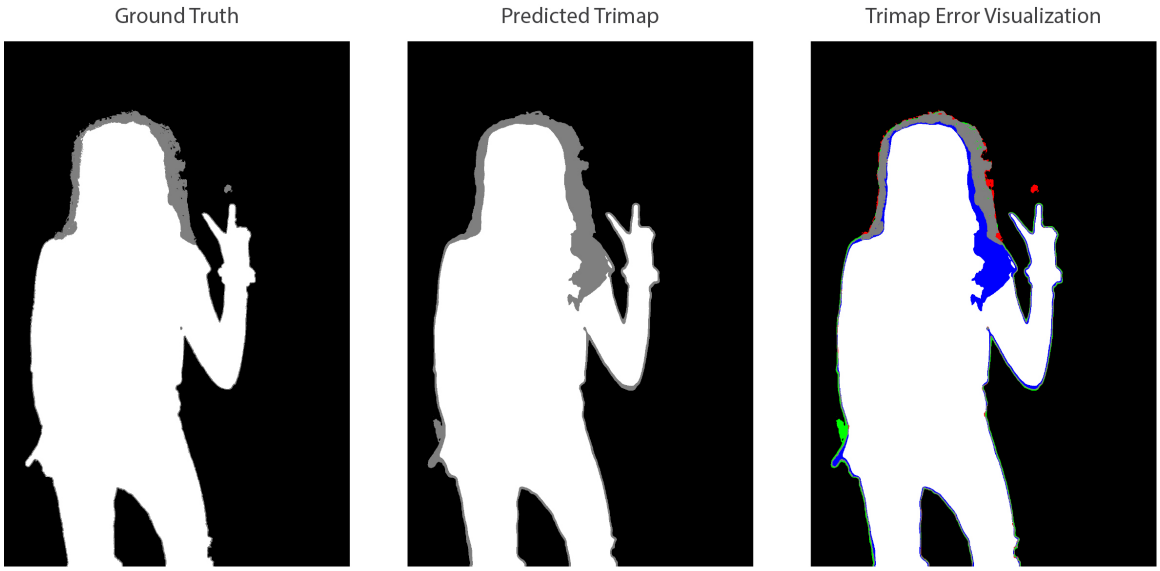


Figure 4.4: Trimap error visualization showing acceptable and unacceptable prediction cases. Blue regions indicate the foreground predicted as unknown; green regions indicate the background predicted as unknown (both acceptable). Red regions highlight critical errors, where foreground is predicted as background, background as foreground, or uncertain regions as either foreground or background (unacceptable).

where s is the sigmoid function that converts the output logits into probabilities, and M_{Fg} and M_{Bg} represent the ground truth foreground and background masks, respectively.

Mean Squared Error Loss: While the BCE loss addresses the general classification task for each of the foreground and background maps, it is also important to guide the network in a manner that aligns with the specific requirements of trimap generation. We designed an additional MSE loss to penalize severe errors, which include predicting uncertain regions as foreground or background, predicting foreground as background and vice versa.

The MSE loss is introduced to explicitly penalize critical errors that would misguide the matting network, which relies on the trimap to accurately distinguish between these regions. The MSE loss is defined as:

$$\mathcal{L}_{MSE}^{Fg} = MSE(\hat{M}_{Fg}, M_{Fg}), \quad \mathcal{L}_{MSE}^{Bg} = MSE(\hat{M}_{Bg}, M_{Bg}). \tag{4.3}$$

To justify the selection of MSE, we include an error visualization (Figure 4.4) that demonstrates how the network's predictions align with the ground truth. The figure consists of three components: the ground truth trimap, the predicted trimap, and an error visualization trimap. In the error visualization, regions where the network predicts foreground or background as uncertain are coloured blue (foreground as unknown) and green (background as unknown). These cases are acceptable, as they reflect the network's uncertainty, which is less problematic to the matting process. However, regions where the network incorrectly predicts foreground

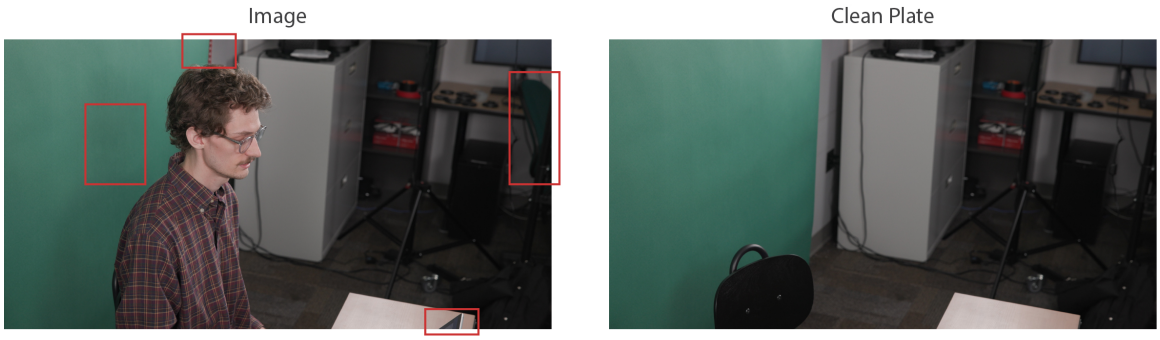


Figure 4.5: Illustration of common discrepancies in the clean plate background compared to the true background. Notable differences include shadows cast by the subject, and objects such as the chair, laptop, and chain. These discrepancies affects the accuracy of the matting process.

as background, background as foreground, or uncertain as either foreground or background are highlighted in red. These are unacceptable cases, as they could significantly mislead the matting network.

Combined Trimap Estimation Loss: To effectively balance these objectives, we combine the BCE and MSE losses for the trimap network's training. The overall loss function is formulated as:

$$\mathcal{L}_{tri} = \mathcal{L}_{BCE}^{Fg} + \lambda \cdot \mathcal{L}_{MSE}^{Fg} + \mathcal{L}_{BCE}^{Bg} + \lambda \cdot \mathcal{L}_{MSE}^{Bg}, \tag{4.4}$$

where λ is a weighting factor set to 10 to amplify the impact of the MSE loss.

4.2.2 Localized Background Matting

To accurately estimate the alpha matte and foreground, we try to minimize the number of unknowns in the compositing equation (Equation 4.1). One way to achieve this is by providing a clean plate that represents the background in the scene. However, this clean plate often differs slightly from the true background due to various factors.

As shown in Figure 4.5, some of these discrepancies include shadows cast by the subject on the green screen, as discussed earlier in section 2.1, as well as objects like the chair, laptop, and chain that were present during the capture. These differences make the clean plate unusable for our matting task.

Background Inpainting

To address the limitations of using the clean plate, we turn to an approach that estimates the true background by inpainting the unknown regions in the background from the input composite image. Conceptually, this follows the work of Tang *et al.* [59], who suggested that background inpainting can be framed as a simplified image inpainting problem. In

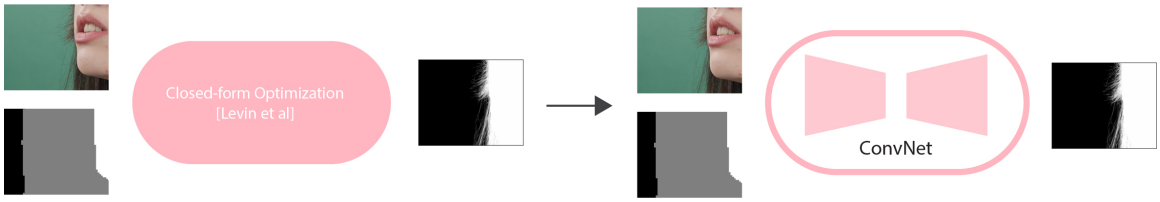


Figure 4.6: We replace the closed-form optimization with a ConvNet architecture, which estimates both the alpha and foreground colors while incorporating the local smoothness prior through its learned filters.

their approach, the mask input traditionally used in image inpainting is replaced with a trimap, as we are only concerned with inpainting the unknown regions within the trimap.

In our method, we integrate a background inpainting block into the second stage of our matting network. This block takes the composite image and the estimated trimap as input and generates a consistent background that more accurately reflects the true scene. To achieve this, we retrain a state-of-the-art inpainting model [58] to ensure that the generated backgrounds are temporally consistent across frames, accounting for the slight differences between the captured frames and the clean plate.

Background Inpainting Loss

For the background inpainting task, we supervise the network using a combination of the mean absolute error (MAE) loss on image I and a multi-scale smoothness loss (MSG) [33, 61] applied to the spatial image gradients over $m = 4$ scales. The MSG loss ensures smooth gradient transitions in the background, while the MAE loss enforces sharp detail reproduction. These losses are defined as:

$$\mathcal{L}^{Bg}_{\text{MSG}} = MSG(\hat{Bg}, Bg) = \frac{1}{N}\sum_{m} SAD(\nabla \hat{Bg}^{m}, \nabla Bg^{m}), \tag{4.5}$$

where Bg is the ground truth background acquired by masking the input image I with the background region from the trimap. By limiting the reconstruction supervision to only the background area for this sub-stage, the network additionally learns to implicitly inpaint the foreground.

Our base loss for background inpainting is then given by:

$$\mathcal{L}_{\text{Base}} = \mathcal{L}^{Bg}_{\text{MAE}} + \mathcal{L}^{Bg}_{\text{MSG}}. \tag{4.6}$$

Background Matting

Our image matting stage conceptually follows Levin *et al.* [23]. They demonstrate that for a spatially constrained image window, the matting problem can be solved via closed-form

optimization subject to some smoothness constraints. The smoothness constraint is based on the assumption that within a small local window, the foreground and background colors should vary smoothly, leading to a smooth transition in the alpha values. This constraint enforces that the alpha values of neighboring pixels should not change abruptly unless there is a significant color difference indicating a boundary between the foreground and background. While this method applies the smoothness prior uniformly across the image, it can be too restrictive or too loose depending on the content of the image.

Given these limitations, we lift this approach by replacing the closed-form optimization with a Convolutional Network (ConvNet) architecture and estimating both the alpha and foreground colors. ConvNet architectures are fitting candidates for the localized matting formulation since their receptive field naturally supports accurate detail prediction in image windows. The ConvNet incorporates the local smoothness prior through its learned filters, which can adapt based on the specific content of the image, applying stronger or weaker smoothing where needed. This approach allows us to incorporate the local smoothness prior from [23] via the network's receptive field without suffering from the limitation of applying the prior uniformly over the entire image. Additionally, the nature of the matting task prevents global inconsistency issues during inference on the full image. This is an advantage against transformer-based architectures, which would require larger models with additional computing to achieve a similar level of accuracy and are thus mostly trained on lower image resolutions [27, 44].

The image matting network itself is trained to predict the alpha matte and foreground color of an input frame with the estimated trimap and inpainted background as auxiliary inputs. Crucially, the network only sees small image crops as input during training, encouraging it to focus on the intricate details within the uncertain area of the trimap. Only in inference do we feed the full-resolution input to the network, which is then able to propagate the learned information through its convolutional structure. We explain our data processing in detail in Section 4.3.1

Image and Video Matting Losses

A combination of five distinct losses supervises both the image and video matting networks. These losses are designed to ensure accurate, high-resolution alpha estimation and foreground color reproduction.

Alpha and Foreground Losses We use the same MAE + MSG base combination as in the background inpainting task but apply it separately to the estimated alpha matte α and the foreground colors Fg. The Mean Absolute Error (MAE) ensures that the average difference between the estimated and ground-truth values is minimized, providing a straightforward and effective measure of accuracy. On the other hand, the Multi-scale Gradient (MSG) loss is crucial for capturing fine details and edges at multiple scales. By applying the gradient loss

across four different scales, we ensure that the network can accurately predict the intricate boundaries of the alpha matte and foreground colors, especially in high-resolution images.

Hard Mask Loss To further refine the alpha matte, we employ the Hard Mask loss (HM) as introduced by Zhou *et al.* [69]. This loss focuses supervision on the pixels that are hardest to determine, as identified by the highest absolute differences:

$$\mathcal{L}_{\text{HM}}^{\alpha} = \frac{1}{S} \sum_{S} |\alpha_{pred} - \alpha| \cdot mask, \tag{4.7}$$

where S represents the top p-% of the pixels within the masked region, with $p = 50$ during training.

Color Loss We introduce a color loss $\mathcal{L}_{\text{Color}}$ in the YUV color space for better foreground color reproduction. Unlike individual RGB losses, which may not fully capture perceptual differences in color, the YUV color space separates an image into brightness (Y) and color components (U and V). This representation aligns more closely with human vision, as brightness and color are processed separately by the human visual system. By computing the Mean Squared Error (MSE) between the estimated and ground-truth foreground colors in this space, the color loss $\mathcal{L}_{\text{Color}}$ ensures that the network produces foreground colors that are perceptually closer to the ground truth. This approach is particularly advantageous for maintaining the integrity of colors in complex scenes where subtle variations in color can significantly impact the overall quality.

$$\mathcal{L}_{\text{Color}} = MSE(\text{YUV}(\hat{Fg}), \text{YUV}(Fg)), \tag{4.8}$$

where $\hat{Fg}$ and Fg are the estimated and ground-truth foreground colors, masked with the non-zero areas in α.

Compositing Loss Finally, we enforce the accuracy of the reconstructed input frame by ensuring that the predicted foreground, alpha, and background adhere to the compositing equation (Eq. 4.1):

$$\mathcal{L}_{\text{Comp}} = MAE(\hat{I}, I), \quad \hat{I} = H(\hat{\alpha} * \hat{Fg} + (1 - \hat{\alpha}) * Bg), \tag{4.9}$$

where H is the harmonization network [6], as detailed in Sec. 4.3.1.

Final Matting Loss The final training loss for both the image and video matting networks is a weighted sum of the individual losses:

$$\mathcal{L}_{\text{Image}} = \mathcal{L}^{Fg}_{\text{Base}} + \lambda_{\text{Color}} * \mathcal{L}^{Fg}_{\text{Color}} + \mathcal{L}^{\alpha}_{\text{Base}} + \mathcal{L}^{\alpha}_{HM} + \mathcal{L}_{\text{Comp}}, \tag{4.10}$$

where λ_{Color} and the gradient weights in $\mathcal{L}_{\text{Base}}$ are set to 5.

4.2.3 Temporal Consistency

The third and final stage of our pipeline forms a video matting network. It is tasked with predicting Fg_t and α_t for a given frame I_t from the previous Fg_{t-1} and α_{t-1} in addition to the inpainted background Bg_t. This formulation serves two purposes: First, the additional inputs from the previous frame shift the focus onto the small temporal differences instead of the more complicated full matting task, leaving again more capacity for detail reconstruction and allowing an enforced temporal consistency. Second, it enables us to reduce the amount of required auxiliary input modalities to a single clean plate per image sequence.

The video matting network's pipeline shares large parts with the image matting network from Stage 2 and is trained likewise on image crops.

Video Temporal Consistency Loss

To ensure smooth transitions between consecutive frames in the video matting network, we introduce a temporal consistency loss $\mathcal{L}_T$, adapted from Sun *et al.* [57]. This loss is applied in a cycle-consistent manner to enforce temporal coherence across frames. For each pair of consecutive frames, we first perform a forward iteration where the previous frame's ground truth modalities are used as inputs, and the image matting loss $\mathcal{L}_{\text{Image}}$ is applied to the current frame's outputs. Subsequently, a second iteration is performed in the reverse temporal direction: here, the network estimates the previous frame using the current frame's estimates as auxiliary inputs. This reverse iteration is supervised by the temporal consistency loss $\mathcal{L}_T$, defined as:

$$\mathcal{L}_T = MSE(\Delta\hat{\alpha}, \Delta\alpha), \quad \Delta\alpha = \alpha_t - \alpha_{t-1}, \tag{4.11}$$

where $\Delta\alpha$ represents the difference between the alpha mattes of consecutive frames. This cycle-consistent approach ensures that the network learns to maintain temporal coherence, even when predicting frames in reverse order. The final loss for the video matting network is thus a combination of the image matting loss and the temporal consistency loss:

$$\mathcal{L}_{\text{Video}} = \mathcal{L}_{\text{Image}} + \lambda_T * \mathcal{L}_T, \quad \lambda_T = 5. \tag{4.12}$$

4.3 Implementation

In this section, we discuss the detailed implementation of each stage of our green screen keying pipeline, which is specifically tailored to our studio setup.

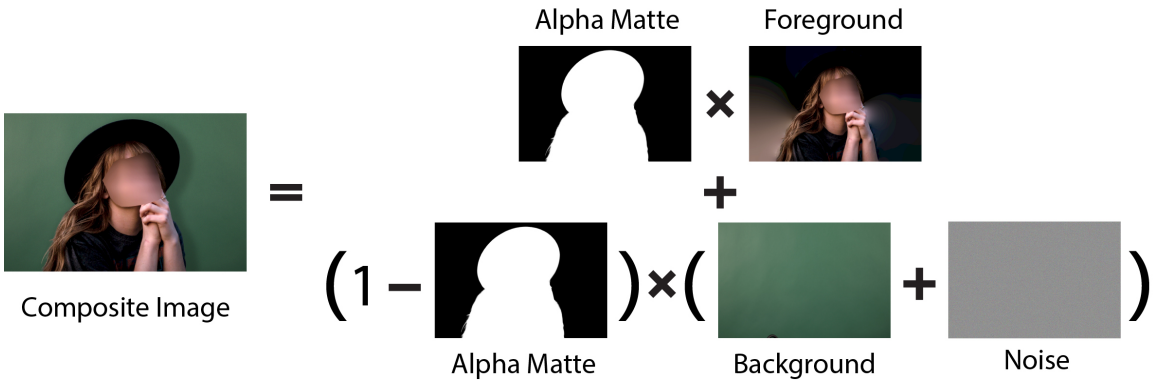


Figure 4.7: Illustration of our synthetically generated composite image for training following Eq. 4.1. This operation is done after linearization of the foreground and background image with a simple gamma-correction, gamma=2.2.

4.3.1 Training Data

We create a large collection of available matting datasets, consisting of Composition-1k [64], AIM-500 [30], Alpha Matting [50], Distinctions-646 [45], PPM-100 [21], P3M-10k [29], RefMatte [31], SIMD [55], Tears of Steel [12], and extend the provided backgrounds with set of randomly selected images from Unsplash[1] as well as captured studio screens. All datasets were manually screened, and faulty foregrounds or alpha channels were removed.

During training, we randomly select foreground-alpha pairs and a background image and generate synthetic composites following Eq. 4.1 after linearization with a simple γ-correction, $\gamma = 2.2$. To increase the degree of realism, we augment the backgrounds with additive Gaussian noise $\mathcal{N}(0.9, 0.1)$ and artificial shadows by using the randomly blurred, shifted, and scaled alpha as a shadow map. Lastly, we apply a recent image harmonization method [6] to further increase the realism. Figure 4.7 shows the compositing operation of a sample from our training dataset.

These synthetically composited images are the foundation for all three stages. Before feeding the data into the inpainting and trimap networks, the gamma correction is removed so that these networks receive the data in its original gamma-corrected form. The matting networks, however, receive the linear data as input. Only the background inpainting network from Stage 2 is trained with an unmodified background, i.e. without artificial noise and shadows. Additionally, we apply supplemental augmentations and modifications for each stage, tailored to the individual training objective.

Trimap Generation We generate two types of trimaps during the training process. The first trimap, the ground-truth trimap, is used to supervise our trimap network in Stage 1. This trimap is derived directly from the selected alpha mattes, where alpha values within

[1] www.unsplash.com

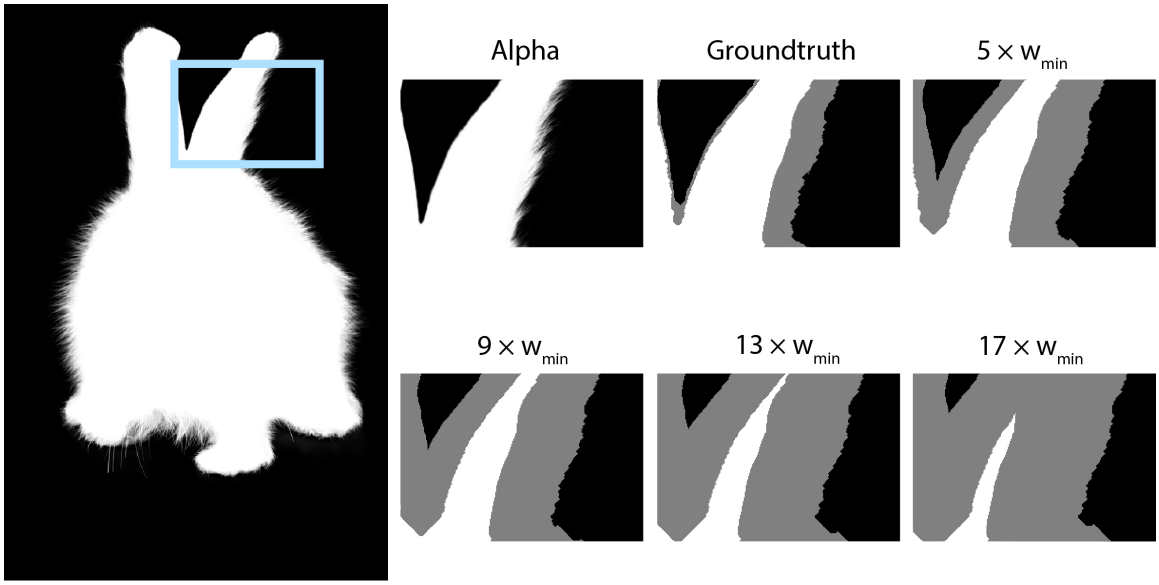


Figure 4.8: Illustration of the different generated trimap during training. The groundtruth trimap is used for the supervision of the trimap network. The dilated trimap is used to mimick user-drawn trimaps during the training of the inpainting and image matting network. We show the effect of dilating using kernel sizes of the minimum width od the trimap scaled by different factors.

$(0, 1)$ are set to 0.5. For training the trimap network, the generated ground-truth trimap and all network inputs are randomly square-cropped to the minimum side length of the image and then scaled to the receptive field size of the network.

The second trimap is simulated to mimic a user-drawn trimap, which is used for robustness during the training of the inpainting and image matting networks. Starting from the ground-truth trimap, these maximally narrow trimaps are dilated to enhance robustness. The dilation is performed by setting the size of the dilation kernel to the minimum width of the trimap, scaled by a random factor $s \in [5, 20]$. Figure 4.8 illustrates the effect of the different dilation kernel sizes used during training.

Patch Selection Our aim is to train matting networks that can accurately separate foreground colors from the background as guided by a trimap. Characteristically, the uncertain area of the trimap as our primary point of interest only takes up a small amount of the full image. We, therefore, generate small patches from the inputs to train both of our matting networks. This approach grants us several advantages. For one, we can benefit from a closer fit to the local smoothness assumption from closed-form matting [23], resulting in an easier task for the network. Secondly, the networks do not waste computing on already established large background / foreground areas but focus on the smaller areas of interest instead.

We base the size for our training patches again on the minimum uncertainty width in the trimap, multiplied with a random scale factor in $[2, 6]$. We then sample several patches from

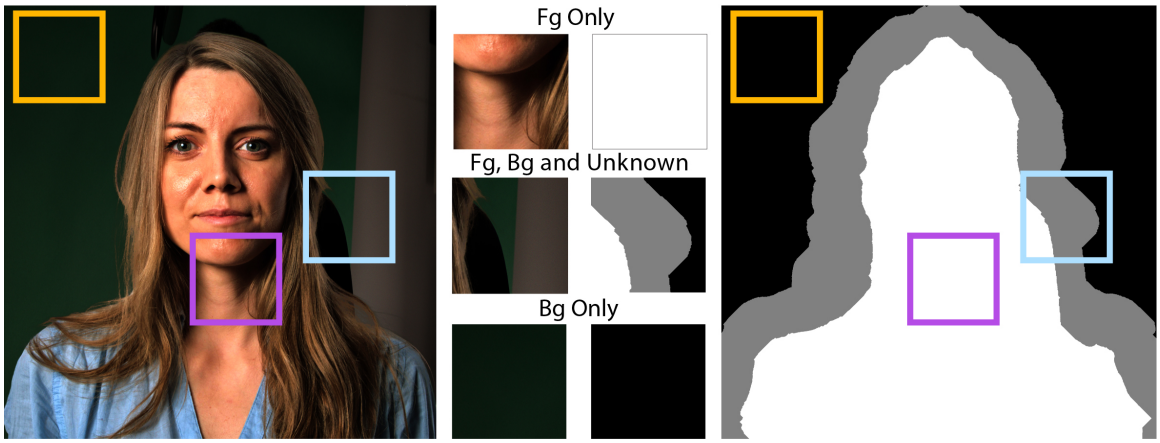


Figure 4.9: Illustration of patch selection strategy during training. The figure shows examples of patches selected from full foreground full background, and the uncertain area containing all three regions—foreground, background, and the uncertain area.

an image, constraining the location along the uncertainty area such that the selected patch shows all three areas with 95% probability. For the remaining cases, full-background and full-foreground patches are sampled equally, which increases the robustness during inference on the full image. Figure 4.9 illustrates this patch selection strategy, showing examples of patches that include these cases.

Synthetic Frame Pairs Training the video matting network poses a separate challenge, as it requires consecutive frame pairs as input. Unfortunately, the few existing video matting datasets, to our knowledge, do not match the quality requirements of professional postproduction environments. Hence, we resort to creating synthetic frame pairs from our composites and apply a set of random, frame-rate dependant homographies to simulate a temporal offset. For a given image, we first sample a frame rate fps $\in [24, 30, 50, 60, 120]$. We then copy the image and apply random translation, rotation and scaling to the copy, such that the sampled parameters for each transformation are scaled by $s_t = \frac{25}{\text{fps}}$. This yields a frame pair with slight offsets between the frames that vary in magnitude depending on the frame rate as an approximation of motion. We show an example of a resulting image pair in Fig. 4.10 and describe the procedure in more detail in the supplementary material.

4.3.2 Network Architectures and Training

We use the same ConvNet architecture for our trimap generator, the image matting network, and the video matting network. Based on its proven success in high-resolution imaging tasks [5,6,42], we adapt the decoder from [46] with ResNeXt101 as backbone [63] according to the respective task. For the trimap generator, we use the raw outputs of the last convolutional layer directly that suit our loss combination detailed in Sec. 4.2.1 in training and apply a sigmoid during inference. For both matting networks, we change the output

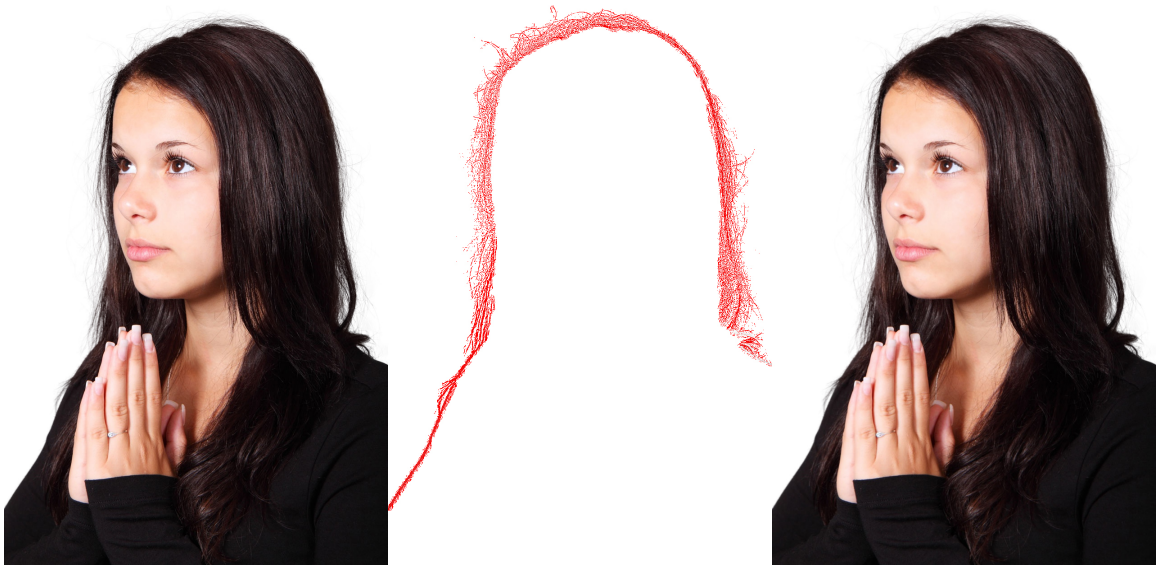


Figure 4.10: We show an example of an artificial frame pair with their pixel difference for the video matting training [30].

activation to a hard tanh, limiting the outputs to lie in $[0, 1]$ with a linear transition. We select RAdam [38] as our optimizer and CosineAnnealing [39] scheduling for all networks with learning rates of 1×10^{-4} for the trimap training and 2.5×10^{-4} for the matting networks, respectively. For the background inpainting, we keep the original setup from [58] and only change the in- and output channels as described.

4.4 Experiments

In this section, we present a comprehensive evaluation of our proposed video matting method, specifically tailored for movie post-production and optimized for our studio's unique requirements. Although our system is designed with these specific needs in mind, we still performed qualitative comparisons with state-of-the-art (SOTA) methods to demonstrate its effectiveness. We selected Background Matting V2 (BMv2) [35] as the current SOTA background matting method, Robust High-Resolution Video Matting (RVM) [36] as a fully automated video matting baseline, and FBA Matting (FBA) [11] as an interactive matting technique. For methods requiring trimap inputs, we generated trimaps using our trimap network.

4.4.1 Detail Reconstruction

Our method excels in reconstructing fine details, which is critical for high-quality matting, especially in post-production environments. Figure 4.11 presents a comparative analysis of our method against the aforementioned baselines, focusing on detail preservation in

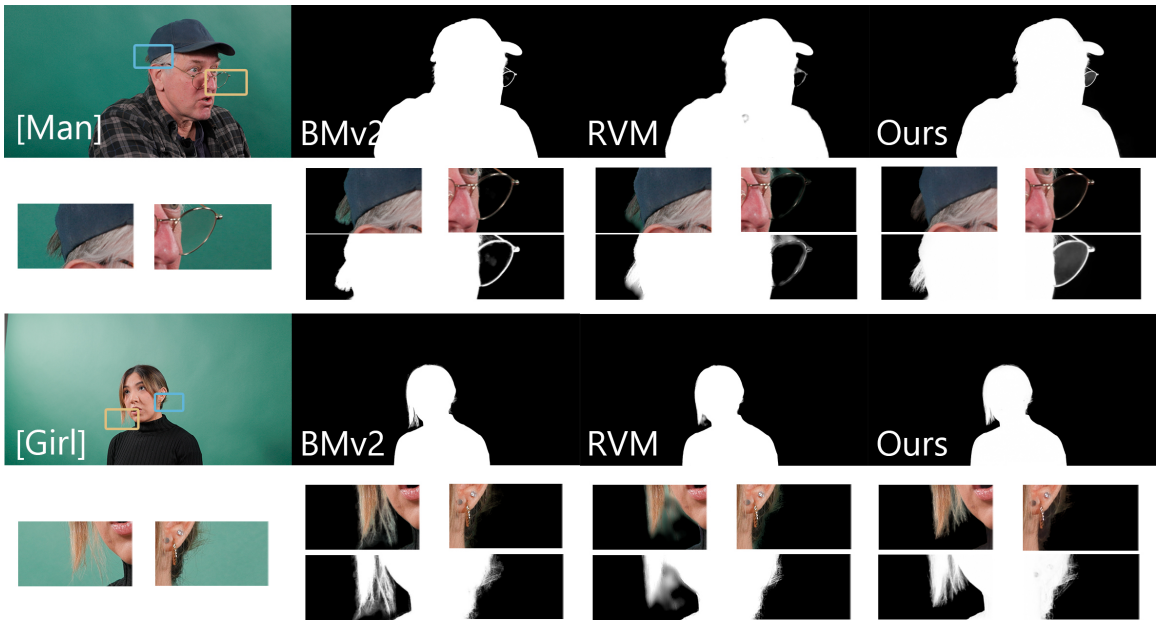


Figure 4.11: We show two examples of the detail reconstruction for our high-resolution studio data.

challenging regions such as hair, edges, and glasses. The samples used for comparison were captured in our studio setup, reflecting real-world production scenarios. As shown, our approach significantly outperforms the baselines in maintaining intricate details, which is essential for producing high-fidelity alpha mattes.

4.4.2 Temporal Consistency

Temporal consistency is crucial for video matting, as flickering or inconsistency between frames can distract the viewer and degrade the quality of the final production. To evaluate this aspect, we compared our method's temporal consistency against BMv2, RVM, and FBA over five consecutive frames. As illustrated in Figure 4.12, our method achieves better temporal stability, reducing artifacts and ensuring smoother transitions between frames.

4.4.3 Comparison Against Commercial Keying Solutions

To validate our method further, we conducted comparisons against commercial keying solutions, specifically Keylight. Figure 4.13 shows that our method delivers comparable performance in reconstructing details in challenging areas. Also, our method correctly handles scenarios where Keylight struggles, such as estimating inaccurate alpha matte for an earring that is green, an issue where Keylight often fails due to green screen spill.

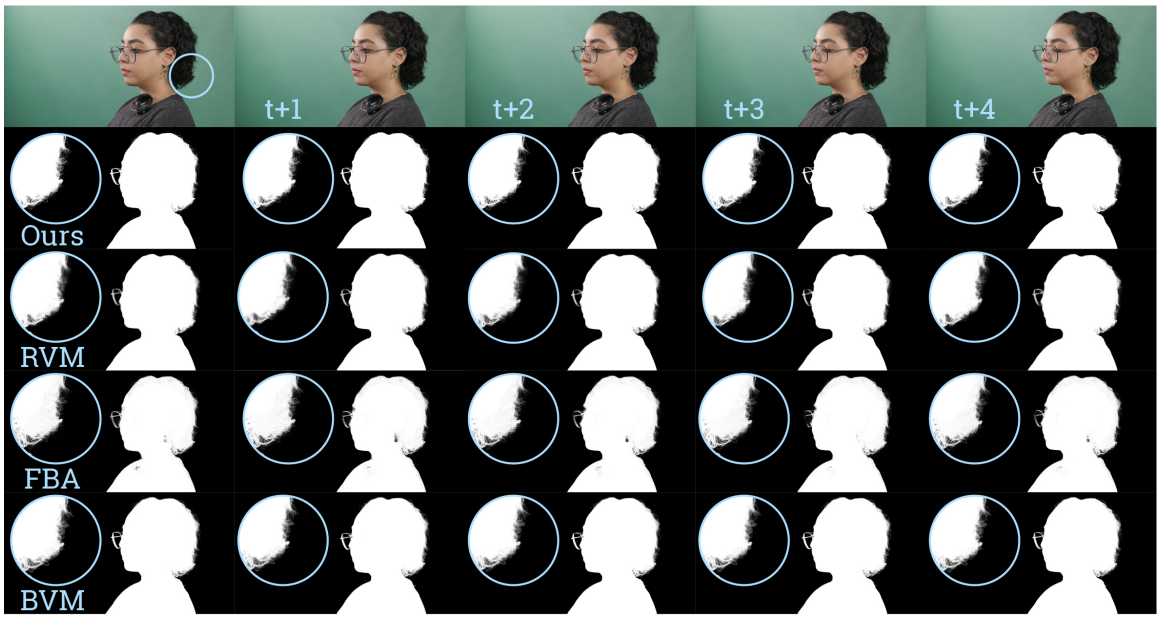


Figure 4.12: We show our results on our captured studio data for 5 consecutive frames.

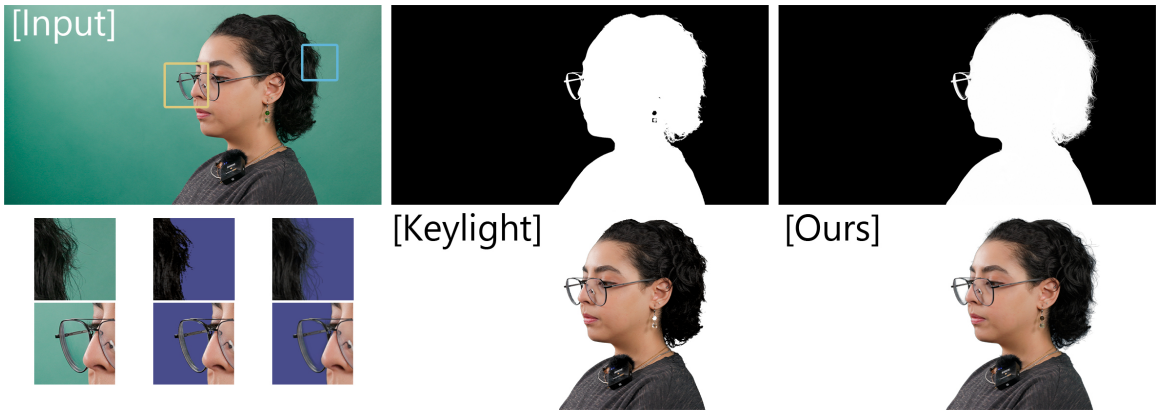


Figure 4.13: We show the result of our method in comparison with a result obtained from Keylight. Our method achieves similar performance in the reconstruction of details in the hair and the glasses and creates a complete alpha including the earring.

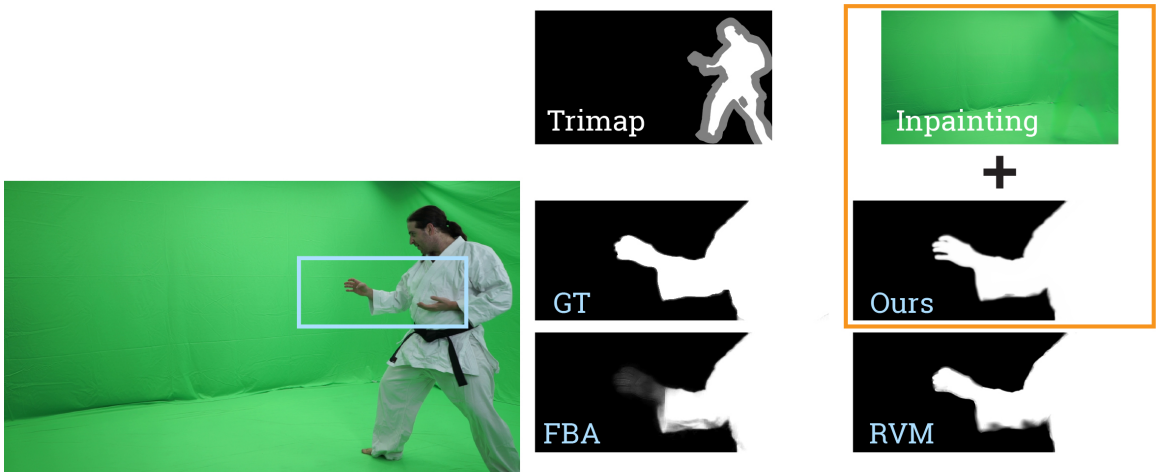


Figure 4.14: Comparison of alpha mattes produced by our method, FBA [11] , and RVM [36] under intense green spill conditions. Our method successfully maintains foreground integrity, avoiding the spill-induced errors seen in other approaches.

4.4.4 Effect of Intense Green Spill on Predicted Alpha

This is a case where the trimap is not sufficient for keying. Intense green spill from the green screen onto the subject can lead to the network confusing the spill on the foreground as a mix of both foreground and background. This confusion typically occurs in the uncertain region of the trimap, where the network must differentiate between foreground elements contaminated by the green spill and the actual background. Such scenarios often result in degraded matting quality, where foreground regions around the transition area are incorrectly classified as background, leading to incomplete or incorrect alpha mattes.

Our background inpainting networks plays a crucial role in resolving the ambiguity caused by green spills. Given an accurate trimap, the inpainting network infers the uncertain region by propagating information from the known background, effectively distinguishing between the foreground plus spill and the inferred background. This distinction allows our matting network to bypass the difficulty posed by the spill, as it no longer needs to contend with the uncertainty in the spill region. As a result, our method is more robust against intense green spill, avoiding the common pitfalls that other matting methods encounter in these scenarios.

For this evaluation, we compare our method against FBA Matting (FBA) [36] and Robust High-Resolution Video Matting (RVM) [36], as these methods do not require a clean plate, similar to our approach. Our approach maintains the integrity of the foreground, as demonstrated in Figure 4.14. Our method outperforms the baselines, providing more accurate alpha mattes even in the presence of significant green spills.

Figure 4.15: Comparison of alpha mattes generated at different inference resolutions: half the original resolution, the original resolution, and twice the original resolution, highlighting the improvement in detail preservation and edge precision at higher resolutions.

4.4.5 Inference Resolution

The resolution at inference plays a critical role in determining the quality of the resulting alpha matte. We observed that performing inference at a resolution twice the original input resolution leads to a substantial enhancement in matte quality. Figure 4.15 presents a comparative analysis of alpha mattes estimated at three different scales: half the original resolution, the original resolution, and twice the original resolution. The results demonstrate that inference at twice the original resolution preserves finer details and improves the details at the edges with intricate textures.

Chapter 5

Conclusion and Future Work

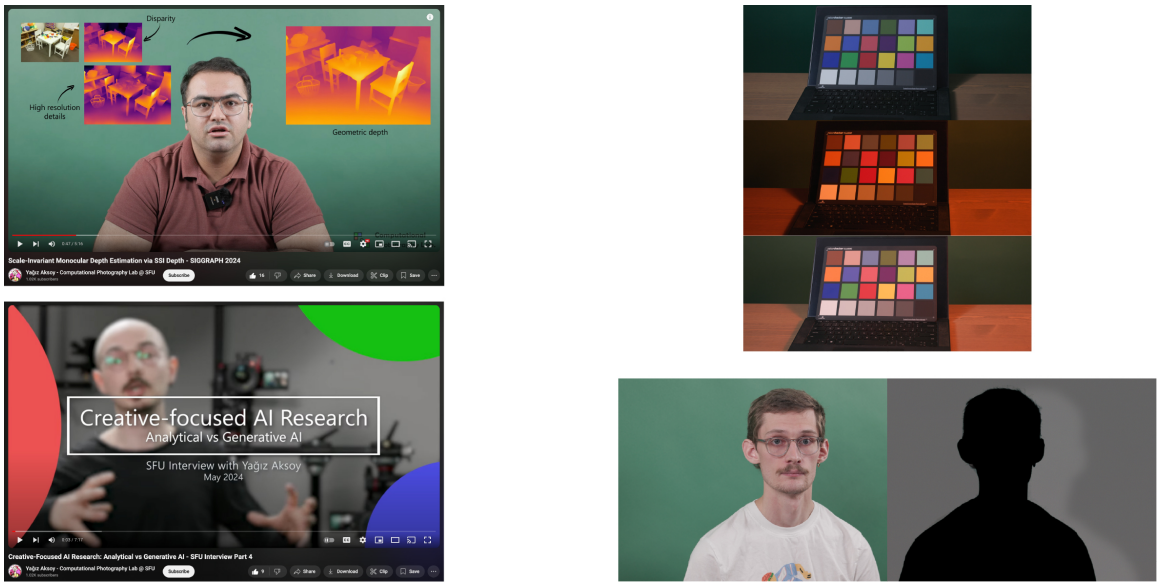


Figure 5.1: We are able to perform research and production simultaneously, from paper presentations and interviews to multi-illumination experiments and shadow analysis.

In this work, we presented the development and implementation of a computational photography research studio designed to be flexible, portable, user-friendly, and suitable for individuals with non-technical abilities. Our primary goal was to enable simultaneous AI research and production activities, addressing the limitations of traditional setups that rely heavily on computer vision cameras and controlled lighting, often resulting in high complexity.

In our studio, we are able to produce production-level videos, including paper presentations and interviews, and also perform computer vision research, such as multi-illumination experiments and shadow analysis, as shown in Figure 5.1.

Figure 5.2: Illustration of the drift in accuracy observed in the video matting network starting from the 50th frame.

5.0.1 Key Considerations in Studio Setup

A key consideration when setting up a studio is the importance of controlling external light sources. Since our objective was to maintain a controlled lighting environment, it was vital to block all external light that could interfere with the scene's illumination. We achieved this by using a matte aluminum sheet, which effectively prevented any external light from entering the studio. This consideration is essential for anyone designing a similar setup, especially when precise lighting conditions are required.

Another important consideration was the calibration process. While effective, the use of a checkerboard pattern posed certain limitations. The pattern needed to be fully visible in the frame before points could be detected, which required capturing up to 50 calibration orientations per camera. This was a time-consuming process and also resulted in the shortened lifespan of the clapboard due to frequent clapping. Although we mitigated this by mixing patterns and using smaller ones to increase visibility, the calibration process remains lengthy and resource-intensive.

5.0.2 Challenges and Limitations

While we successfully implemented a video matting network that maintained temporal coherence, we observed a drift in accuracy starting from the 50th frame. To address this, we re-applied the image matting network at the 50th frame and continued with the video matting network for the subsequent 50 frames. This approach, while effective, highlights a limitation in our current method and suggests the need for further research to improve long-term temporal coherence in video matting.

Another challenge we encountered was in achieving complete time synchronization between cameras. Although we were able to determine the millisecond time shift between cameras, the presence of missing frames meant that complete synchronization remained unsolved. One potential solution could involve interpolating frames to account for these discrepancies, but this approach requires further investigation, particularly for high-quality frame interpolation.

5.0.3 Future Work

Moving forward, some potential improvements include reducing the reprojection error, extending the versatility of the calibration toolbox to accommodate different types of calibration boards and camera models, and optimizing only one camera while keeping the extrinsic of other cameras fixed during the multi-camera calibration process.

Additionally, enhancing the accuracy and efficiency of the green screen keying method to fully integrate it into the studio's post-production pipeline is another promising direction for future work. Further research into these areas will help refine the studio setup and make it even more accessible and effective for AI-driven production.

Overall, this thesis has provided valuable insights into the challenges and considerations involved in setting up a computational photography research studio.

Bibliography

[1] Yağız Aksoy, Tunç Ozan Aydın, and Marc Pollefeys. Designing effective inter-pixel information flow for natural image matting. In *Proc. CVPR*, 2017.

[2] Yağız Aksoy, Tunç Ozan Aydin, Marc Pollefeys, and Aljoša Smolić. Interactive high-quality green-screen keying via color unmixing. *ACM Trans. Graph.*, 35(5):1–12, 2016.

[3] Walter Beyer. Traveling-matte photography and the blue-screen system: a tutorial paper. *Journal of the SMPTE*, 74(3):217–239, 1965.

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

[5] Chris Careaga and Yağız Aksoy. Intrinsic image decomposition via ordinal shading. *ACM Trans. Graph.*, 2023.

[6] Chris Careaga, S Mahdi H Miangoleh, and Yağız Aksoy. Intrinsic harmonization for illumination-aware image compositing. In *Proc. SIGGRAPH Asia*, 2023.

[7] Qifeng Chen, Dingzeyu Li, and Chi-Keung Tang. Knn matting. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(9):2175–2188, 2013.

[8] Robert Collins. Introduction to computer vision. Lecture notes, 2007. CSE/EE486: Computer Vision I, Pennsylvania State University.

[9] Kenji Enomoto, TJ Rhodes, Brian Price, and Gavin Miller. Polarmatte: Fully computational ground-truth-quality alpha matte extraction for images and video using polarized screen matting. In *Proc. CVPR*, 2024.

[10] Wolfgang Förstner and Eberhard Gülch. A fast operator for detection and precise location of distinct points, corners and centres of circular features. In *Proc. ISPRS intercommission conference on fast processing of photogrammetric data*, volume 6, pages 281–305, 1987.

[11] Marco Forte and François Pitié. F, b, alpha matting. *CoRR*, abs/2003.07711, 2020.

[12] Blender Foundation. Tears of steel, 2013.

[13] Ioannis Gkioulekas. Computer vision. Lecture notes, 2020. 16-385: Computer Vision, Carnegie Mellon University.

[14] Nick I. M. Gould, Caroline Sainvitu, and Philippe L. Toint. A filter-trust-region method for unconstrained optimization. *SIAM Journal on Optimization*, 16(2):341–357, 2005.

[15] Anselm Grundhöfer and Oliver Bimber. Virtualstudio2go: digital video composition for real environments. *ACM Trans. Graph.*, 27(5):1–8, 2008.

[16] Anselm Grundhöfer, Daniel Kurz, Sebastian Thiele, and Oliver Bimber. Color invariant chroma keying and color spill neutralization for dynamic scenes and cameras. *The Visual Computer*, 26:1167–1176, 2010.

[17] Hendrik Hachmann and Bodo Rosenhahn. Color-aware deep temporal backdrop duplex matting system. In *Proc. MMSys*, 2023.

[18] Wei-Lun Huang and Ming-Sui Lee. End-to-end video matting with trimap propagation. In *Proc. CVPR*, 2023.

[19] Chuong Huynh, Seoung Wug Oh, , Abhinav Shrivastava, and Joon-Young Lee. Maggie: Masked guided gradual human instance matting. In *Proc. CVPR*, 2024.

[20] Yue Jin, Zhaoxin Li, Dengming Zhu, Min Shi, and Zhaoqi Wang. Automatic and real-time green screen keying. *The Visual Computer*, 38(9):3135–3147, 2022.

[21] Zhanghan Ke, Jiayu Sun, Kaican Li, Qiong Yan, and Rynson WH Lau. Modnet: Real-time trimap-free portrait matting via objective decomposition. In *Proc. AAAI*, 2022.

[22] Philip Lee and Ying Wu. Nonlocal matting. In *Proc. CVPR*, 2011.

[23] Anat Levin, Dani Lischinski, and Yair Weiss. A closed-form solution to natural image matting. *IEEE Trans. Pattern Anal. Mach. Intell.*, 30(2):228–242, 2008.

[24] Anat Levin, Alex Rav-Acha, and Dani Lischinski. Spectral matting. *IEEE Trans. Pattern Anal. Mach. Intell.*, 30(10):1699–1712, 2008.

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

[26] Hanxi Li, Wenyu Zhu, Haiqiang Jin, and Yong Ma. Automatic, illumination-invariant and real-time green-screen keying using deeply guided linear models. *Symmetry*, 13(8):1454, 2021.

[27] Jiachen Li, Vidit Goel, Marianna Ohanyan, Shant Navasardyan, Yunchao Wei, and Humphrey Shi. Vmformer: End-to-end video matting with transformer. In *Proc. WACV*, 2024.

[28] Jiachen Li, Jitesh Jain, and Humphrey Shi. Matting anything. In *Proc. CVPR*, 2024.

[29] Jizhizi Li, Sihan Ma, Jing Zhang, and Dacheng Tao. Privacy-preserving portrait matting. In *Proc. MM*, 2021.

[30] Jizhizi Li, Jing Zhang, and Dacheng Tao. Deep automatic natural image matting. In *Proc. IJCAI*, 2021.

[31] Jizhizi Li, Jing Zhang, and Dacheng Tao. Referring image matting. In *Proc. CVPR*, 2023.

[32] Yaoyi Li and Hongtao Lu. Natural image matting via guided contextual attention. In *Proc. AAAI*, 2020.

[33] Zhengqi Li and Noah Snavely. Cgintrinsics: Better intrinsic image decomposition through physically-based rendering. In *Proc. ECCV*, 2018.

[34] Chung-Ching Lin, Jiang Wang, Kun Luo, Kevin Lin, Linjie Li, Lijuan Wang, and Zicheng Liu. Adaptive human matting for dynamic videos. *Proc. CVPR*, 2023.

[35] Shanchuan Lin, Andrey Ryabtsev, Soumyadip Sengupta, Brian Curless, Steven M. Seitz, and Ira Kemelmacher-Shlizerman. Real-time high-resolution background matting. *Proc. CVPR*, pages 8758–8767, 2020.

[36] Shanchuan Lin, Linjie Yang, Imran Saleemi, and Soumyadip Sengupta. Robust high-resolution video matting with temporal guidance. *Proc. WACV*, pages 3132–3141, 2021.

[37] Philipp Lindenberger, Paul-Edouard Sarlin, and Marc Pollefeys. LightGlue: Local Feature Matching at Light Speed. In *ICCV*, 2023.

[38] Liyuan Liu, Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Jiawei Han. On the variance of the adaptive learning rate and beyond. In *Proc. ICLR*, 2020.

[39] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. In *Proc. ICLR*, 2017.

[40] Sebastian Lutz, Konstantinos Amplianitis, and Aljoscha Smolic. Alphagan: Generative adversarial networks for natural image matting. In *Proc. BMVC*, 2018.

[41] MATLAB. Camera calibration.

[42] S. Mahdi H. Miangoleh, Mahesh Reddy, and Yağız Aksoy. Scale-invariant monocular depth estimation via ssi depth. In *Proc. SIGGRAPH*, 2024.

[43] Trung-Thanh Ngo, Asatilla Abdukhakimov, and Dong-Seong Kim. Long-range wireless tethering selfie camera system using wireless sensor networks. *IEEE Access*, 7:108742–108749, 2019.

[44] GyuTae Park, SungJoon Son, JaeYoung Yoo, SeHo Kim, and Nojun Kwak. Matteformer: Transformer-based image matting via prior-tokens. In *Proc. CVPR*, 2022.

[45] Yu Qiao, Yuhao Liu, Xin Yang, Dongsheng Zhou, Mingliang Xu, Qiang Zhang, and Xiaopeng Wei. Attention-guided hierarchical structure aggregation for image matting. In *Proc. CVPR*, 2020.

[46] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE Trans. Pattern Anal. Mach. Intell.*, 44(3):1623–1637, 2020.

[47] Ananth Ranganathan. The levenberg-marquardt algorithm. 2004.

[48] Anyi Rao, Linning Xu, Zhizhong Li, Qingqiu Huang, Zhanghui Kuang, Wayne Zhang, and Dahua Lin. A coarse-to-fine framework for automatic video unscreen. *IEEE Trans. Multimedia*, 25(11):2723–2733, 2023.

[49] Erik Reinhard, Erum Arif Khan, Ahmet Oguz Akyz, and Garrett M. Johnson. *Color Imaging: Fundamentals and Applications*. A. K. Peters, Ltd., 2008.

[50] Christoph Rhemann, Carsten Rother, Jue Wang, Margrit Gelautz, Pushmeet Kohli, and Pamela Rott. A perceptually motivated online benchmark for image matting. In *Proc. CVPR*, 2009.

[51] Laura Gonçalves Ribeiro, Ahmed Durmush, Olli Suominen, and Atanas Gotchev. Photogrammetric multi-camera calibration using an industrial programmable robotic arm. In *2019 11th International Symposium on Image and Signal Processing and Analysis (ISPA)*, 2019.

[52] Soumyadip Sengupta, Vivek Jayaram, Brian Curless, Steve Seitz, and Ira Kemelmacher-Shlizerman. Background matting: The world is your green screen. In *Proc. CVPR*, 2020.

[53] Dmitriy Smirnov, Chloe LeGendre, Xueming Yu, and Paul Debevec. Magenta green screen: Spectrally multiplexed alpha matting with deep colorization. In *Proc. DigiPro*, 2023.

[54] Alvy Ray Smith and James F Blinn. Blue screen matting. In *Proc. SIGGRAPH*, 1996.

[55] Yanan Sun, Chi-Keung Tang, and Yu-Wing Tai. Semantic image matting. In *Proc. CVPR*, 2021.

[56] Yanan Sun, Chi-Keung Tang, and Yu-Wing Tai. Ultrahigh resolution image/video matting with spatio-temporal sparsity. In *Proc. CVPR*, 2023.

[57] Yanan Sun, Guanzhi Wang, Qiao Gu, Chi-Keung Tang, and Yu-Wing Tai. Deep video matting via spatio-temporal alignment and aggregation. In *Proc. CVPR*, 2021.

[58] Roman Suvorov, Elizaveta Logacheva, Anton Mashikhin, Anastasia Remizova, Arsenii Ashukha, Aleksei Silvestrov, Naejin Kong, Harshith Goka, Kiwoong Park, and Victor Lempitsky. Resolution-robust large mask inpainting with fourier convolutions. In *Proc. WACV*, 2022.

[59] Jingwei Tang, Yağız Aksoy, Cengiz Öztireli, Markus Gross, and Tunç Ozan Aydın. Learning-based sampling for natural image matting. In *Proc. CVPR*, 2019.

[60] Pete Tomkies. Understanding bit depth and color rendition for video, August 27 2019.

[61] Benjamin Ummenhofer, Huizhong Zhou, Jonas Uhrig, Nikolaus Mayer, Eddy Ilg, Alexey Dosovitskiy, and Thomas Brox. Demon: Depth and motion network for learning monocular stereo. In *Proc. CVPR*, 2017.

[62] Xian Wu, Xiao-Nan Fang, Tao Chen, and Fang-Lue Zhang. Jmnet: A joint matting network for automatic human matting. *Computational Visual Media*, 6(2):215–224, 2020.

[63] Saining Xie, Ross Girshick, Piotr Dollar, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proc. CVPR*, 2017.

[64] Ning Xu, Brian Price, Scott Cohen, and Thomas Huang. Deep image matting. In *Proc. CVPR*, 2017.

[65] Jingfeng Yao, Xinggang Wang, Shusheng Yang, and Baoyuan Wang. Vitmatte: Boosting image matting with pre-trained plain vision transformers. *Information Fusion*, 103:102091, 2024.

[66] Zijian Yu, Xuhui Li, Huijuan Huang, Wen Zheng, and Li Chen. Cascade image matting with deformable graph refinement. In *Proc. ICCV*, 2021.

[67] Georgi Zapryanov, Daniela Ivanova, and Iva Nikolova. Automatic white balance algorithms for digital still cameras – a comparative study. 2012.

[68] Yuhongze Zhou, Liguang Zhou, Tin Lun Lam, and Yangsheng Xu. Sampling propagation attention with trimap generation network for natural image matting. *IEEE Trans. Circuits Syst. Video Technol.*, 33(10):5828–5843, 2023.

[69] Yuhongze Zhou, Liguang Zhou, Tin Lun Lam, and Yangsheng Xu. Sampling propagation attention with trimap generation network for natural image matting. *IEEE Trans. Circuits Syst. Video Technol.*, 33(10):5828–5843, 2023.

Appendix A

Code

Appendices should be used for supplemental information that does not form part of the main research. Remember that figures and tables in appendices should not be listed in the List of Figures or List of Tables.