

“Lights Are Gonna Find Me”: Creating a Reactive Lighting Control Scheme

---

A Thesis  
Presented to  
The Interdisciplinary Committee for Computer Science and Theatre  
Reed College

---

In Partial Fulfillment  
of the Requirements for the Degree  
Bachelor of Arts

---

Gabriel Howland

May 2025



Approved for the Committee  
(Computer Science and Theatre)

---

Jim Fix

---

Peter Ksander



# Acknowledgements

There are too many people who have positively affected my life. To name them all would double the length of this thesis.

To start off, I want to thank my advisors Jim and Peter. Peter, thank you so much for being in my corner these past four years from the academic advising Zoom call during the summer of my first year when I expressed my interest in this amalgamation that is my major. Jim, thank you so much for being so willing to help see this project out. Your support was so essential whenever I was feeling discouraged after yet another week of bugs and confusion. This experience was unforgettable for me, and I hope it was the same for y'all.

I want to thank my family. My parents, Leigh, Miles, Oma, Bapa, Noni, Popi, Aunt Jen, Uncle John, Aunt Liz, Uncle Scott, Derek, Adam, and everyone else. You have all been so supportive throughout my many, many, years of education. I love you all.

Thank you, Yik Yin. Your endless patience with my terrible schedule and planning abilities is so appreciated. Thank you for being by my side these past four years, and I hope to spend many more with you. I love you, and I know there is nothing but amazingness ahead of you.

Thank you to Rusty, L., and Rowan for being such amazing bosses. I have learned so much in my time here, and I have grown so much and in so many directions under your collective supervision. I am so proud of the work I have done in the scene shop and in SHARE and you all helped make that happen. Thank you to Rain, Max, Julia, and all my coworkers past and present. The work was fun, but getting to know you all was the best part of my work.

Thank you to the Theatre and Computer Science departments. You have both welcomed me with open arms and allowed me to find my place here. Thank you to the faculty who put up with me. Thank you to the staff who run these departments, it could not be done without your help. Thank you to my fellow seniors, we did it!

Thank you to my friends, both in Portland and back at home. I hold you all so close to my own heart. You all have made my life for the better, and are what I am going to miss most about this chapter of my life. I could not have finished Reed without every one of you. I love you all, and I hope to see you all again, and again.

Thank you reader, for reading this thesis. I hope you enjoy it.



# List of Abbreviations

|             |                                  |
|-------------|----------------------------------|
| <b>DMX</b>  | Digital Multiplex                |
| <b>ACN</b>  | Architecture for Control Network |
| <b>sACN</b> | Streaming ACN                    |
| <b>DLT</b>  | Direct Linear Transform          |
| <b>SVD</b>  | Singular Value Decomposition     |
| <b>HSV</b>  | Hue Saturation Value             |



# Contents

|  |           |
|--|-----------|
| <b>Introduction</b> . . . . .                                | <b>1</b>  |
| <b>Chapter 1: The Problem</b> . . . . .                      | <b>5</b>  |
| 1.1 A History of Lighting Control . . . . .                  | 6         |
| 1.2 Taking the “Live” out of “Liveness” . . . . .            | 11        |
| 1.3 The Solution (A Solution?) . . . . .                     | 14        |
| <b>Chapter 2: Ops all Cameras</b> . . . . .                  | <b>17</b> |
| 2.1 Introduction . . . . .                                   | 17        |
| 2.2 Camera Basics . . . . .                                  | 18        |
| 2.3 The Camera Model . . . . .                               | 20        |
| 2.4 A (Brief) Linear Algebra Primer . . . . .                | 20        |
| 2.5 Camera Parameters . . . . .                              | 25        |
| 2.6 Stereo Normal Triangulation . . . . .                    | 27        |
| 2.7 Camera Calibration . . . . .                             | 29        |
| 2.8 Generalized Triangulation . . . . .                      | 31        |
| 2.9 From Pixels to Position . . . . .                        | 33        |
| <b>Chapter 3: Bringing the Project to Life (?)</b> . . . . . | <b>37</b> |
| 3.1 The Process Begins . . . . .                             | 37        |
| 3.2 DMX “Hell” . . . . .                                     | 38        |
| 3.3 So how did you get so close? . . . . .                   | 39        |
| 3.4 Where does that put us now . . . . .                     | 40        |
| <b>Conclusion</b> . . . . .                                  | <b>43</b> |
| <b>Appendix A: The Code of the Project</b> . . . . .         | <b>45</b> |
| <b>Bibliography</b> . . . . .                                | <b>47</b> |



# List of Figures

|     |   |    |
|-----|---|----|
| 2.1 | A Pinhole Camera . . . . .  | 18 |
| 2.2 | Left: A camera with a wide aperture, allowing for light from one point to project in multiple areas. Right: A camera with a lens to refocus the light back to where it should go. . . . . | 20 |
| 2.3 | Dimensional Fingers showcasing 2-Space and 3-Space . . . . .  | 21 |
| 2.4 | Translation of a point $q$ to $q'$ . . . . .  | 24 |
| 2.5 | Scaling of a point $q$ to $q'$ . . . . .  | 24 |
| 2.6 | Counter-clockwise rotation of a point $q$ to $q'$ by $90^\circ$ . . . . .   | 24 |
| 2.7 | Flipping the Projection Plane . . . . .   | 27 |
| 2.8 | A Stereo Normal Example . . . . .   | 28 |
| 2.9 | Left: Image of LED, Right: Mask of LED by HSV . . . . .   | 35 |



# Abstract

I propose a stereoscopic vision system to track a performer in a theatrical space, and cover the process to creating such a software. Part of that includes an overview of photogrammetry, and the process for calibrating cameras, locating objects in three-dimensional space, and positioning a light in said space. This project is informed by hundreds of years of innovation in lighting control that has culminated in a system that forgoes the need for human operators, and is essentially "perfect." However, this "perfect" system is not a "living" or "live" system, which disconnects it from the performers.



# Introduction

Tonight, the Super Trouper lights are gonna find me  
Shining like the sun (Sup-p-per Troup-p-per)  
Smiling, having fun (Sup-p-per Troup-p-per)  
Feeling like a number one

Tonight, the Super Trouper beams are gonna blind me  
But I won't feel blue (Sup-p-per Troup-p-per)  
Like I always do (Sup-p-per Troup-p-per)  
'Cause somewhere in the crowd there's you<sup>1</sup>

In this thesis I propose a reactive lighting control scheme that tracks a performer on a stage. This thesis explores lighting control, specifically for live performance. It takes a look at how lighting was originally controlled manually, how the technology has advanced to today through the use of time coding, and proposes a system for control that is reactive to a performer. Lighting control has advanced at a breakneck speed over the past half-century as the world entered a digital age. Where lighting control rooms were packed with levers and room-scale dimming racks now sit lighting desks, or even just a laptop. As the technology for controlling lights has advanced, the lighting design for live performances has gotten more complicated, while being largely prerecorded, resulting in perfectly repeatable sequences. While a designer's vision is accomplished in collaboration with the director and performers, it stays static barring catastrophe, which introduces problems. Consider the actor who moves in tandem to a moving light. This is usually accomplished through hours of rehearsal, and fine-tuning movements in order to keep pace with the light. However, if the actor

---

1. ABBA – *Super Trouper*, accessed May 2, 2025, <https://genius.com/Abba-super-trouper-lyrics>.

in a particular showing wanted to modify a movement, either in the route or in the speed, they would be constrained by the lighting.

This thesis attempts to alleviate those constraints on route and speed by creating a lighting control scheme that tracks a performer through a theatrical space. It is a reactive scheme that hopes to meet a few criteria:

1. The scheme must be able to run using off-the-shelf components and computing resources.
2. The scheme must be open source.
3. The scheme should run on top of existing theater infrastructure.

It is worth mentioning that this concept is not revolutionary. There are numerous choices for high-end performer tracking offered by the likes of Cast Group's BLACK-TRAX<sup>2</sup>, Follow-Me<sup>3</sup>, and zactrack<sup>4</sup>. However, these systems are always closed-source, prohibitively expensive, hardware intensive, or otherwise difficult for small productions to access. The software developed during this thesis hopes to try to make it easier to access.

This project is spread across three parts. The first engages with the concept of lighting control, and how its advancement to repeatability comes at a disconnect and a cost. It will provide an abridged history of lighting control, and how this modern age of control is without an essential "human" element.

The second chapter establishes a mathematical camera model, calibrates it, and showcases how we use said camera model to locate a point in three-dimensional space.

The third chapter explores the culmination of the work done in the previous two chapters, and the stumbling block I hit when preparing to present the software in a

---

2. "BlackTrax – CAST Group of Companies," April 13, 2025, accessed May 2, 2025, <https://web.archive.org/web/20250413233816/https://cast-soft.com/blacktrax/>.

3. "Home - Follow-Me," February 13, 2025, accessed May 2, 2025, <https://web.archive.org/web/20250213084710/https://follow-me.com/>.

4. "Automated Follow System — Zactrack Lighting Technologies GmbH — Wien," May 1, 2025, accessed May 2, 2025, <https://web.archive.org/web/20250501182028/https://www.zactrack.com/>.

lighting design for the dance thesis of Beier (Belle) Li. It covers the issues I faced while attempting to create a prototype, and where this process leaves off at.



# Chapter 1

## The Problem

In this chapter, I present the problem and context behind said problem explored in this thesis. Throughout history, lighting was intertwined with the performance of the actor; technicians controlling the lights in methods akin to their own performance. The systems that controlled lights only encouraged this, as they all had their own quirks and eccentricities that introduced variability into every show. In this modern age, the number of technicians involved has only continued to shrink as modern software focuses on complexity, perfection, and repeatability. This isn't necessarily a bad direction, lighting has only become more spectacular and intricate, but there's a growing disconnect between the actors and the design, especially with automated lighting.

When I refer to this disconnect, it is the relationship between the mover and the performer that suffers the most. The versatility of a moving light to position anywhere on a set should be freeing for a performer, but oftentimes the inverse is true. Since these automated lights are given a series instructions, they will act repeatably, which means that the performer is usually expected to do the same. This repetition has adverse effects on the liveness that is so essential for the theatre. Here lies the beginning of the problem.

For many theatres the fix is simple: Hire a follow spot operator, or use one of the commercial products available like Blacktrax. However, for the theatres with budgets too low for the commercial option and the inability to either hire an operator or hold a follow spot, there should still exist a solution. That is the rest of the problem, and that is what I hope to solve.

This software is not intended to be a replacement for a human operator, it is designed to be a tool for theatres and productions unable to use conventional solutions. It is software that is open-source, and designed to work with off the shelf hardware like web cameras.

First, we are going to go back. It is important to see how we got to the control systems we use today, and also to see how lighting control changed as a concept, and how the disconnect formed. Then we look deeper into where we are today, specifically analyzing the commercial software available today, and what their shortcomings are. Finally, equipped with the previous knowledge of what exists today and what did exist, I will go deeper into the specification of this software.

## 1.1 A History of Lighting Control

We begin in the 16th century, in the “cradle of stage lighting”<sup>1</sup> that was the Italian Renaissance. Here lies the earliest records and examples of a “standard practice” for lighting a theatre. When looking at the compiled histories of lighting and its role in theatre, authors agree that the distinction of lighting from scenic elements took its hold here. Richard Pilbrow’s history of lighting technology in Stage Lighting Design indicates that there is evidence of paint and reflective costumes compensating for the less visually interesting sunlight and torchlight used previously: “A stage direction of 1501 in the Mona Passion reads, ‘See to it that the painter goes to Paradise to paint

---

1. Richard. Pilbrow, *Stage Lighting Design : The Art, the Craft, the Life* (New York: By Design Press, 1997), p. 163, ISBN: 0-89676-139-8.

Raphael's face red.' In the same play, Christ is described with gold hands and feet, and 'let there be a big sun behind him.'"<sup>2</sup> Max Keller acknowledges that there was use of artificial lighting in Greek and Roman theatre, but "nothing has been recorded or passed down to us about theatrical lighting as we understand it."<sup>3</sup>

The first recorded lighting control came at the turn of the 17th century. Nicola Sabbattini, an Italian architect, outlined a rudimentary system of dimming light through tin cans suspended over candles or lamps. He describes it as follows: "When it is desired to darken the whole stage in a moment, this method is used: as many cans of soldered tin are made as there are lamps to be darkened. [This] done, you adjust each cylinder over its lamp [in] such a manner that by one motion on the side of the stage, the cylinders descend over the lamps and darken them."<sup>4</sup> Wax and oil was expensive, and in order to be able to see performers, many were needed. Theatres opted to use hundreds<sup>5</sup> of candles or lamps positioned strategically to cast the best light on the performers. The process of lighting and maintaining the candles was a gargantuan task, requiring many technicians to participate in the upkeep. Sabbattini mentions this in his text: "Every care must be taken to [light the candles] as quickly as possible to avoid restlessness in the spectators who think this business is endless." Wicks needed to be trimmed, lamps refilled, and molten wax would sometimes fall upon the spectators.<sup>6</sup>

The next jumps in technology appeared in the 19th century when gas-powered light entered the theatrical scene in 1803 with a demo in the Lyceum Theatre. Full adoption appeared in the Chestnut Theatre in 1816 and the Lyceum in 1817.<sup>7</sup> Soon

---

2. Pilbrow, p. 166-167.

3. Max. Keller et al., *Light Fantastic : The Art and Design of Stage Lighting*, 2nd rev. and updated ed., Light (Munich ; Prestel, 2006), p. 33, ISBN: 978-3-7913-3685-5.

4. Barnard Hewitt, ed., *The Renaissance Stage : Documents of Serlio, Sabbattini and Furttenbach*, trans. Allardyce Nicoll, John H. McDowell, and George R. Kernodle (Coral Gables, Fla. : University of Miami Press, 1958), p. 111-112, ISBN: 978-0-87024-004-1, accessed December 9, 2024, <http://archive.org/details/renaissancestage0000unse>.

5. Pilbrow, *Stage Lighting Design : The Art, the Craft, the Life*, p. 173.

6. Hewitt, *The Renaissance Stage*, p. 97.

7. Pilbrow, *Stage Lighting Design : The Art, the Craft, the Life*, p. 174.

after, the gas table—a contraption with levers, valves, and piping designed to control the flow of gas to any location in the theatre—became produced by companies like Clemanccon. These gas tables are considered the first “switchboard” for theatrical lighting, or elaborate control schemes that could modify the intensity of a light from a single location. These tables had pipes that would control sections of the theatre, like footlights, auditorium lights, proscenium lights, etc.<sup>8</sup>

A few years after gas lighting was adopted, inventor Thomas Drummond was inspired by lectures he attended at the Royal Institution. He witnessed chemist Michael Faraday mix hydrogen and oxygen in the presence of calcium oxide (or lime).<sup>9</sup> The resulting lime would create a bright white light source. Drummond quickly created a small enclosure with piping for the gases and a block of quicklime. This was named the Drummond light; the produced light, when reflected, created a pure beam, allowing a performer to be directly illuminated. This type of light was named after the quicklime used inside its body, and thus “limelight” was created in 1826.<sup>10</sup> This style of lighting allowed for brighter, whiter light while still being controlled by the gas table. The directness of the light also allowed it to be turned into the first follow-spot.

Following the advent of the incandescent lamp in 1879, the Savoy Theatre in London fully converted from gas-light to electric light. Large theatre venues were quick to adopt the technology. After the Savoy’s adoption of electric lighting, most large venues fully converted by 1887. This rapid conversion to electric lighting was due to the sheer improvements an incandescent bulb gave over gas. Gas light consumes oxygen, produces fumes, and lots of heat. Compared to gas, incandescent lamps produce less heat, and consume no oxygen creating a much nicer theatre going experience: “A

---

8. The above text contains a collection of excellent renderings of gas tables and other pre-electrical devices

9. Craig M. Carver, *A History of English in Its Own Words*, in collab. with Internet Archive (HarperCollins, 1991), p. 157, ISBN: 978-0-06-270013-1, accessed May 1, 2025, <http://archive.org/details/historyofenglish00carv>.

10. Pilbrow, *Stage Lighting Design : The Art, the Craft, the Life*, p. 174.

Captain Shaw noted that the temperature in the grid of the Savoy was 68°Fahrenheit, while at the Alhambra, lit with gaslight, it was 105°.”<sup>11</sup> There was also a dramatic safety improvement from removing open flames from the theatrical spaces. During the Savoy’s opening, proprietor Richard D’Oyly Carte “appeared before the curtain and demonstrated the new safety of electricity by smashing a lit lamp wrapped in muslin. This was greeted with tumultuous cheers.”<sup>12</sup>

Electricity requires a different form of control. Instead of a flame providing the illumination, light is produced by running current through a thin filament. When provided current, the filament heated up, reaching temperatures high enough to make the filament glow. If a light wanted to be dimmed, one could decrease the current going to a bulb by increasing resistance. Multiple mediums were used: be it sand, water, or different amounts of copper. In its early introduction, dimmers took up lots of room; the gas tables—which could be operated by a single person—turned into room scale operations, humming with electricity.<sup>13</sup> Lighting control was dominated in the 1930’s by systems such as the Bordoni and Salani control systems, which were variable rate transformers, which allowed finer control over the dimming of a light.<sup>14</sup> These were mechanical apparatus, with interlocking gears that often required multiple technicians to manage it.<sup>15</sup> A byproduct of these room scale dimmers was the *grand master* and *master* control arms. These were levers or wheels that could be locked to specific dimmers to allow all of them to dim at the same time, even if set at different levels.

Technological advancements progressed swiftly. The next step was to shrink the footprint of the control surface. This was originally done through hundreds of small lines of wire called “tracker wire” which allowed the levers or switches to be placed

---

11. Pilbrow, p. 176.

12. Pilbrow, p. 175.

13. Pilbrow, p. 179.

14. Keller et al., *Light Fantastic : The Art and Design of Stage Lighting*, p. 207.

15. Frederick Bentham, *Stage Lighting*, in collab. with Internet Archive (London : G. Allen & Unwin, 1980), p. 84-89, accessed May 1, 2025, <http://archive.org/details/stagelighting0000fuch>.

closer together—and more importantly, further from the dimmers—into one large bank. Not twenty years later vacuum tubes and relays were all the rage, and the “light organ” became the first lighting console that was mechanically independent from the dimmers (opting to control the dimmers electrically rather than mechanically).<sup>16</sup> These were consoles that controlled lighting at a single desk, not by running about a room. Like the gas table previously, the job of lighting a theatrical space returned to just one operator, however the number of controllable lights at the operator’s disposal increased, allowing for more complicated designs. This cut down on technicians and costs. Newer consoles allowed for presets and, shortly after, memory. This was initially set up on punch cards, but as magnetic tape became prominent in computing, it too transitioned to theatre.<sup>17</sup> Once memory was introduced, technicians no longer had to scramble to set each scene, and designers could just load scenes from memory; the console could recreate it with minimal effort.

In 1986, the US Institute for Theatre Technology (USITT) made the jump to digital control with a signaling protocol called Digital Multiplex (DMX<sup>18</sup>). DMX was originally designed to be a standardized protocol for controlling dimmers. With it, (and its revision in the 1990), lighting control devices continued to shrink. The three decades following the genesis of DMX cemented itself as *the* essential protocol for communicating with fixtures, which was vital as lighting consoles advanced in the ability to save and load cues. The Musical Instrument Digital Interface (MIDI<sup>19</sup>) which was created in 1983, was adapted as a control protocol to allow remote activation of lighting cues, often in time with sound or video cues. The adaptation of MIDI

---

16. Keller et al., *Light Fantastic : The Art and Design of Stage Lighting*, 480.

17. Keller et al., 481.

18. *Entertainment Technology—USITT DMX512-A Asynchronous Serial Digital Data Transmission Standard for Controlling Lighting Equipment and Accessories* (ESTA, 2024), accessed May 1, 2025, <https://tsp.estra.org/tsp/documents/docs/ANSI%20E1.11%20-%20202024.pdf>.

19. “MIDI Show Control Specification,” Google Docs, accessed April 29, 2025, [https://drive.google.com/file/d/1Hbx4KbuQRysnEF6L3sXhlwJtc4RE33L/view?usp=drive\\_link&usp=embed\\_facebook](https://drive.google.com/file/d/1Hbx4KbuQRysnEF6L3sXhlwJtc4RE33L/view?usp=drive_link&usp=embed_facebook).

was followed with Open Sound Control (OSC<sup>20</sup>), a control protocol created in 2002, which allowed specificity in control messages.<sup>21</sup> DMX continues to be the dominant communication protocol.

## 1.2 Taking the “Live” out of “Liveness”

Throughout this history, the number of technicians required to run a show fluctuates. For example, back in the candle and oil lit theatres of the Italian Renaissance, Sabbatini recommends a large number of technicians to work together to simultaneously light the candles at the beginning of the show.<sup>22</sup> For the chandeliers that sat above the house, he specifies that three technicians tend each fixture.<sup>23</sup> Similarly, when operating gas lights Bram Stoker observed the Lyceum Theatre employed a “[large number of men] to look after the gas, to light and turn it off as required.”<sup>24</sup> It is with the gas table that records show that a smaller number of operators are required to run a show—as few as one was needed to operate the table because control was made available in one location.<sup>25</sup> The need for technicians grew again with the advent of electrical dimming, as the size of the control surface expanded to room-scale, then soon decreased as the control surfaces consolidated into a single operator at a lighting desk. As modern technology continues to improve, there is not a growth in need for technicians. In fact, as the advancements in control moved from hardware to software, there is less of a need for operators as a whole.

---

20. “OSC Specification 1.0,” April 7, 2021, accessed April 29, 2025, [https://opensoundcontrol.stanford.edu/spec-1\\_0.html#introduction](https://opensoundcontrol.stanford.edu/spec-1_0.html#introduction).

21. MIDI was originally used to send a “note” and a “velocity” which was mapped to shortcuts on the lighting console.

22. Hewitt, *The Renaissance Stage*, p. 97.

23. Hewitt, p. 98.

24. Bram Stoker, “Irving and Stage Lighting,” *The Nineteenth Century and After: A Monthly Review*, May 1911, p. 904, accessed May 1, 2025, <https://www.bramstoker.org/pdf/nonfic/lighting.pdf>.

25. Frederick Penzel, *Theatre Lighting before Electricity*, in collab. with Internet Archive (Middletown, Conn. : Wesleyan University Press, 1978), p. 55, ISBN: 978-0-8195-5021-7, accessed May 1, 2025, <http://archive.org/details/theatrelightingb0000penz>.

Modern lighting—and overall show—control that evolved out of the ability to digitally record cues is timecoding. The concept of timecoding is that there is a global clock running through a performance and certain lights activate after a set amount of time. Timecoding is a technique that plays nice with sound and video cues, allowing lights to match a prerecorded video clip or sound file down to the millisecond. In this modern age, timecoding can affect all aspects of a performance, including the human performers, who rely on certain lighting changes or other design elements to cue their own movements. This begins to present a problem if a performer has any deviation in their performance: “A time based system is less forgiving to human performers. If, for example, the [cue was] triggered at 4 minutes and 35 seconds into every performance, the performer would be out of luck if their performance varied much.”<sup>26</sup> Timecoding is effective as a synchronization technique, but can be a disservice to performers because the computer will not stop without human intervention.

This presents a problem, and one that is the foundation of the problem that exists in my thesis. As lighting control technology has advanced, the need for a human operator—or multiple operators—decreases. Throughout all the previously referenced texts, a through line is present: the operator is intrinsically tied to the performance.

In the introduction of David Haye’s *Light On The Subject*, Peter Brook mentions the process of cuing a production of Hamlet in the Moscow Art Theatre. The lighting designer, Joe Davis, had just finished his 100th cue. He instructed the operators to go back to the first cue to which they registered confusion.

In the Moscow Art Theatre, a lighting plot was unknown. The electricians would be present at every rehearsal, which unlike our miserable four weeks would often last two years. They ended up knowing the play as well as the actors, and slowly built up the lighting stroke by stroke, day by day.

---

26. John Huntington, *Introduction to Show Control: Connecting Entertainment Control Systems for Live Shows* (NYC, USA: Zircon Designs Press, 2023), p. 7, ISBN: 978-1-7357638-4-2.

When the performers came, they did not work by cues, they lived the lighting changes as the actor lived his entrances, his exits and his changing moods.<sup>27</sup>

While two years is a long time to produce and create a show, the idea that the operators. The concept of “lighting rehearsals” (akin to dry tech now) was conceived as a rehearsal entirely for the operators to familiarize themselves with the performance without actors present.<sup>28</sup>

Nowadays, the timecoding and networking of lighting with other media elements continues to reduce the requirement of a technician to even operate the lighting console. This troubles the concept of liveness, which is one of the main draws of theatre. Part of that concept of liveness expresses itself as a fundamental difference between a repeatable and perfect show, and the continual effort to reach said show. The rehearsal process, for actors and operators, progresses towards a perfect but there always runs the risk of variability. Director Jordan Tannahill says as such in his essay, “Why Live: A Question for 21st Century Theatre”: “We continue to rehearse and perform shows with the aim of reproducing the same event—the same text, the same ebbs and flows of laughter and pathos, the same moments of revelation—night after night. In other words, we often aim to create an inert, knowable, and replicable entity.” Tannahill stresses that this process is with the *aim* of reaching a replicable entity, and that aim is what makes a show “live”. Timecoding a show skips this “aim” step, once it is set up, it is already inert. Theodore Fuchs expresses displeasure at the use of digital recording of cues, comparing it to “a pianola used in place of a pianist in an orchestra of live musicians.”

The issue is often money. Timecoding software is so predominant that the cost for entry is low. Hiring operators is usually more expensive than paying for software.

---

27. David Hays, *Light on the Subject : Stage Lighting for Directors and Actors –and the Rest of Us*, 1st Limelight ed. (New York: Limelight Editions, 1989), p. 1, ISBN: 0-87910-126-1.

28. Bentham, *Stage Lighting*, p. 252.

This monetary incentive is present in the upgrades to lighting hardware as well. Electricity was cheaper than gas which was cheaper than candles. Liveness is literally being “priced out,” especially in the spaces with limited resources or budgets.

### **1.3 The Solution (A Solution?)**

Lighting control software is heading in the direction allowing performers to be less restricted by timecoding. Specifically, the more recent advances in software appear in the manipulation of automated lighting.

When I refer to automated lighting, what I am referring to is a lighting fixture (or luminaire) with the ability to change the direction of the beam through the use of motors. Also known as moving lights, movers, or “intelligent” lights, these are controlled through a series of computer instructions, most often through the DMX protocol. Movers have a variety of sizes and capabilities from the Rosco I-Cue™—an attachment placed on the end of an instrument with a moving mirror—to the monstrous Vari-Lite VL3000, a lamp mounted on a two-axis system that allows for the whole unit to pan and tilt. These moving lights are expensive, with price ranges starting in the thousands.

Movers are excellent at a couple of different tasks. They allow a designer to place a pool of light anywhere on a stage (set permitting), can often change color or the shape of the beam simultaneously, and can provide an accent light that moves with a performer. Before movers, that accent light came from a follow spot, which was human operated. Nowadays, these movers are well integrated into the timecode control scheme that exists in the majority of performance venues, which opens up the same issues with repeatability and perfection with an actor.

The present solutions to this issue are two-fold: hire a human operator to make moves live, or use software. These two solutions are currently visible in large venues,

but each is costly. While timecoding software has been present in some form for decades, the live tracking software hasn't. In this case, while humans are expensive, the software is more so. There are only a couple of major software distributors that provide a live-tracking solution: BLACKTRAX, Follow-Me, and zactrak are all live solutions with prices beginning at \$10,000 and increasing with venue size. Neither solution is cost-effective for the minor production outfits who don't have the space for a follow spot and its operator, or the money to shell out for the nice software. That is where my project aims to fit. I believe that the hardware present across all tiers of production is more than capable of tracking a performer; by creating open-source software, I aim to provide a solution for those smaller resourced, smaller budgeted production outfits to take advantage of the technology that is available, and to work to try and restore some liveness back into this timecode world.



# Chapter 2

## Oops all Cameras

### 2.1 Introduction

Now that we are equipped with the context behind this problem, it's time to dive into the theory and calculations that power my proposed solution. My solution is a pipeline, that funnels data through a few processing steps. A performer stands on the stage, holding or wearing a beacon. Ideally, this is emitting light in a non-visible spectrum to not distract the audience. The performer is captured in two webcams, and the images undergo the first round of processing to highlight the beacon in each image. Then, we triangulate the location of the beacon with respect to the cameras. Finally, we take the beacon's location and turn it into instructions for a moving light in the space which will direct its beam onto the beacon.

The vast majority of the following content falls under the field of "photogrammetry" or "the art, science, and technology of obtaining reliable information about physical objects and the environment, through processes of recording, measuring, and interpreting images and patterns of electromagnetic radiant energy and other phenomena" i.e. the measurement of life through images. In particular, we look at the problem of stereo vision, whereby we determine three-dimensional information of

a scene as recorded by two cameras viewing the scene. There is a massive open source library called OpenCV for computer vision that will be used to tackle some problems in the following chapter.<sup>1</sup>

## 2.2 Camera Basics

To start, we present a model for what a camera does, and then how we can use cameras to assess three-dimensional info about a scene. The most basic camera is a pinhole camera, made of a small light-sealed box with a small hole on one end. Light enters the box through that pinhole and creates an image projected on the opposite side of the box.

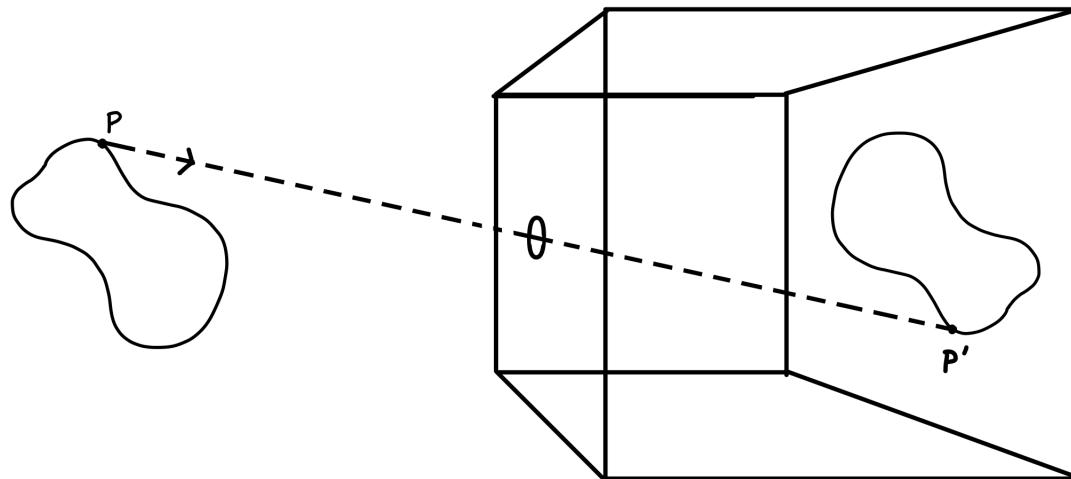


Figure 2.1: A Pinhole Camera

This setup is pictured in Figure 2.1. A couple of things are happening in this image that are worth taking note of. For example, the image seen by the pinhole camera ends up flipping the image in the real world. The reason for that is due to how light operates, and why the pinhole camera is effective in recreating the image.

---

1. Kari Pulli et al., “Real-Time Computer Vision with OpenCV,” *Communications of the ACM* 55, no. 6 (June 2012): 61–69, ISSN: 0001-0782, 1557-7317, accessed April 4, 2025, <https://doi.org/10.1145/2184319.2184337>, <https://dl.acm.org/doi/10.1145/2184319.2184337>.

Let's discuss how this image of the world appears as a projection inside the camera. This is depicted in Figure 2.1. Consider an illuminated object sitting in the world in front of the pinhole camera. Some light energy hitting an object is reflected in lots of directions, with some rays of light reflected in the direction of the camera, and others going elsewhere. Figure 2.1 showcases one such ray, reflecting off of point  $p$ , which travels through the pinhole, and onto the back face of the camera's body at point  $p'$ . Note that in this example, only a single ray is reflected off of point  $p$  and makes it through the pinhole, and so  $p'$  is where that part of the object is represented.

Tracing similar rays of light from different points on the object's surface leads to an (upside down) image of the object within the camera. This aggregate of this process can be visualized as two cones of the rays: The cone inside the box with a peak at the pinhole, and its base formed by the image on the camera wall. There is also the "opposite" cone outside the camera whose base is the surface of the object with its peak at the pinhole's entrance.

Realistically, the pinhole does need to have some width. That results in allowing more than a single ray of light to project from a single point. For example,  $p$  would have a cone of rays that would project onto the back wall of the camera, which would misrepresent the exact placement of  $p$ . However, the smaller the pinhole is, the less light information resulting from each point, thus projecting a dimmer object.

The fix for this is to increase the size of the pinhole, and placing a lens in the aperture. The lens focuses the cone of light rays from a point  $p$  on the object to a single spot  $p'$  inside the camera. An example of a camera with and without a lens are depicted in Figure 2.2.

The cameras that are used today are still based on this pinhole principle, but instead of a backplate, there lies a grid of sensors. Each sensor measures how much light hits that spot. That grid of measurements are then shown back to us as pixels on a screen.

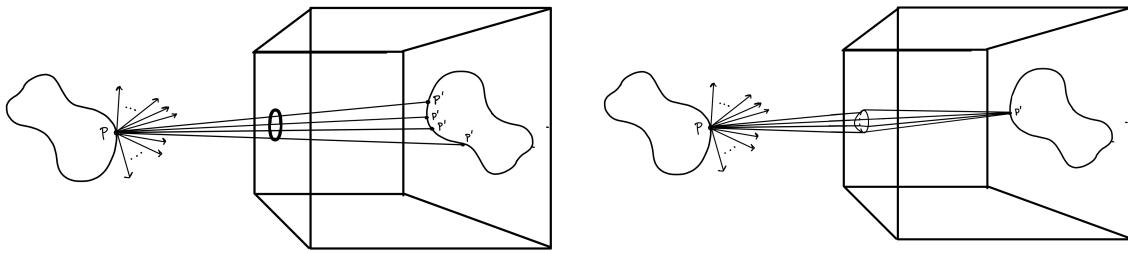


Figure 2.2: Left: A camera with a wide aperture, allowing for light from one point to project in multiple areas. Right: A camera with a lens to refocus the light back to where it should go.

## 2.3 The Camera Model

The idealized pinhole camera model we just described leads to the mathematics for our calculations. More specifically, we seek to calculate the image point  $p'$  on the camera sensor of an illuminating object point  $p$  somewhere in the three-dimensional world. We can represent this process as a matrix:<sup>2</sup>

$$p = \begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix} \quad p' = \begin{bmatrix} x \\ y \\ w \end{bmatrix}$$

$$p' = Mp$$

Where  $p$  is a  $4 \times 1$  vector representing our point in three-dimensional space (or three-space),  $p'$  is a  $3 \times 1$  vector representing the point in two-space, and  $M$  is some  $3 \times 4$  matrix “black box” which will move  $p$  to  $p'$ . More clarification will come on what that exactly means.

## 2.4 A (Brief) Linear Algebra Primer

Before we get too deep into the world of matrices and various transformations, it’s important to establish a baseline understanding of the calculations that are happening

---

2. You may expect the position in  $n$  dimensions to be represented with  $n$  coordinates. However, we will use homogeneous coordinates which adds an extra component  $w$ . Its advantages are forthcoming.



Figure 2.3: Dimensional Fingers showcasing 2-Space and 3-Space

behind the scenes. We live in a three-dimensional world—at least that is our perception of the world. Thus, we can represent the locations of things in our world using vectors of three values:  $(x, y, z)$ . These values represent positions relative to an origin point in distances along three different perpendicular directions. This way of describing a position can be done at home. Take your hand and create the shape of a “finger gun” (where your index finger and thumb are both fully extended), now extend your middle finger halfway (see Figure 2.3 for an example). That creates a coordinate system in terms of your fingers, and you can represent the location of objects in terms of: “how many thumbs, index fingers, and middle fingers does it take to reach an object?”

When it comes to the camera, we have two of our directions (the  $X$  and  $Y$  directions) that are parallel to the  $X$  and  $Y$  axes of the projection plane and a third direction ( $Z$ ) which is perpendicular to the projection plane.

Remember our representation of  $p$  and  $p'$  from earlier:

$$p = \begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix} \quad \text{and} \quad p' = \begin{bmatrix} x' \\ y' \\ z' \\ w' \end{bmatrix}$$

Notice that the vectors representing the location in three-space and the location on the image plane contain an extra value. That is because these vectors are homogeneous coordinates. Homogeneous coordinates are representations of points in

computer graphics (and in our case, photogrammetry) where perspective plays a key role. Remember our representation of  $p$  and  $p'$  from earlier:

$$p = \begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix} \quad \text{and} \quad p' = \begin{bmatrix} x' \\ y' \\ w' \end{bmatrix}$$

The  $w$  component plays numerous roles. One is that it helps us conveniently express affine transformations—translation, scaling, and rotation—in matrix form.

For example, we can scale the points on a two-dimensional object whose corner is at the origin by the following  $3 \times 3$  matrix calculations:

$$q = \begin{bmatrix} 5 \\ 3 \\ 1 \end{bmatrix} \implies q' = \begin{bmatrix} \frac{4}{5} & 0 & 0 \\ 0 & \frac{5}{3} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 5 \\ 3 \\ 1 \end{bmatrix} = \begin{bmatrix} 5 \times \frac{4}{5} \\ 3 \times \frac{5}{3} \\ 1 \end{bmatrix} = \begin{bmatrix} 4 \\ 5 \\ 1 \end{bmatrix}$$

The above performs the calculation for a point  $q$  at the coordinate  $(5, 3)$  specified in homogeneous coordinates. This transformation is shown in Figure 2.5.

Similarly, we can rotate the points on a two-dimensional object whose corner is at the origin by the following  $3 \times 3$  matrix calculations:

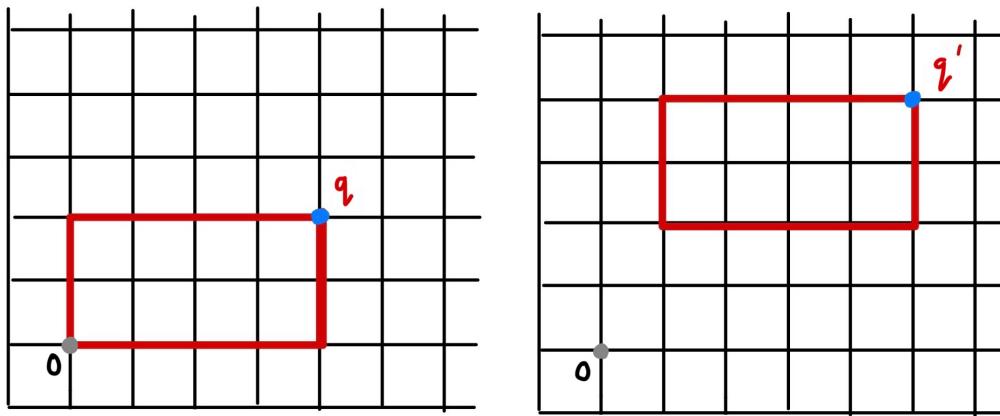
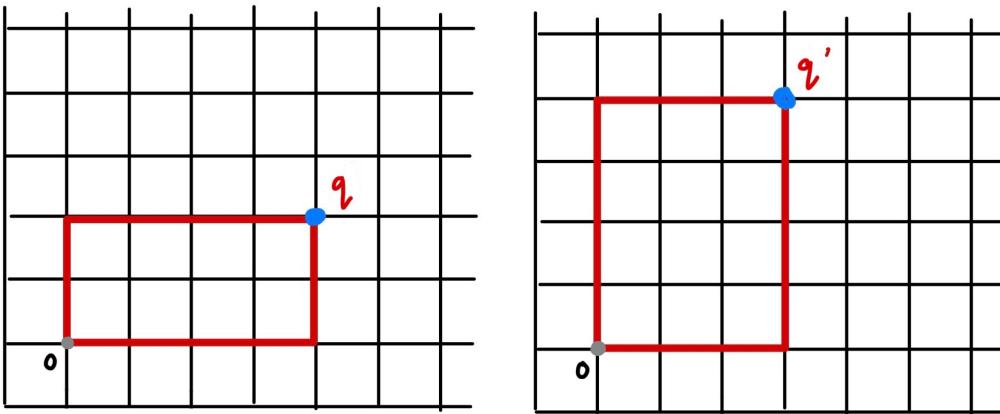
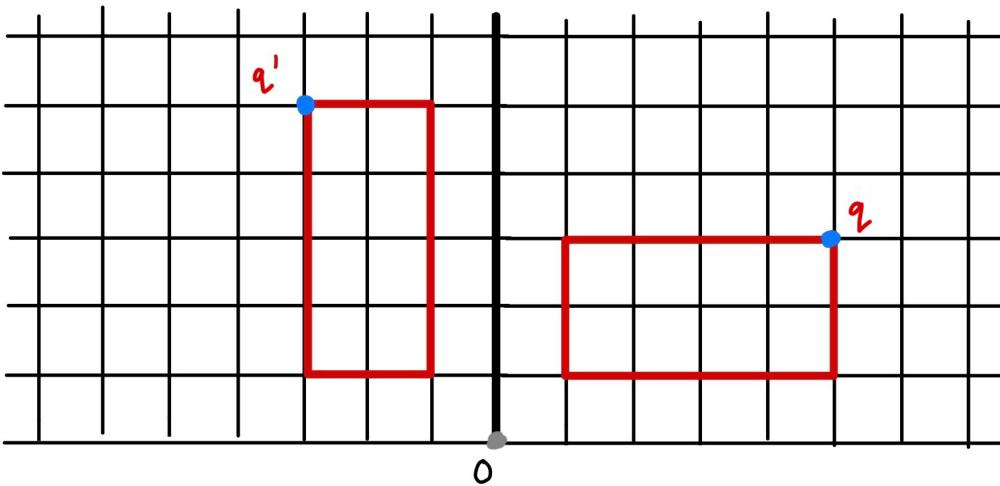
$$\begin{aligned}
q &= \begin{bmatrix} 5 \\ 3 \\ 1 \end{bmatrix} \implies q' = \begin{bmatrix} \cos(90^\circ) & -\sin(90^\circ) & 0 \\ \sin(90^\circ) & \cos(90^\circ) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 5 \\ 3 \\ 1 \end{bmatrix} \\
&= \begin{bmatrix} 5(\cos(90^\circ)) & -3(\sin(90^\circ)) & +1(0) \\ 5(\sin(90^\circ)) & +3(\cos(90^\circ)) & +1(0) \\ 5(0) & +3(0) & +1(1) \end{bmatrix} \\
&= \begin{bmatrix} 5(0) & -3(1) & +1(0) \\ 5(1) & +3(0) & +1(0) \\ 0 & +0 & +1 \end{bmatrix} \\
&= \begin{bmatrix} 3 \\ 5 \\ 1 \end{bmatrix}
\end{aligned}$$

The above performs the calculation for a point  $q$  at the coordinate  $(5, 3)$  specified in homogeneous coordinates. This transformation is shown in Figure 2.6.

Finally, we can translate the points on a two-dimensional object whose corner is at the origin by the following  $3 \times 3$  matrix calculations:

$$q = \begin{bmatrix} 5 \\ 3 \\ 1 \end{bmatrix} \implies q' = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 2 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 5 \\ 3 \\ 1 \end{bmatrix} = \begin{bmatrix} 5+1(1) \\ 3+1(2) \\ 1+0(1) \end{bmatrix} = \begin{bmatrix} 6 \\ 5 \\ 1 \end{bmatrix}$$

The above performs the calculation for a point  $q$  at the coordinate  $(5, 3)$  specified in homogeneous coordinates. This transformation is shown in Figure 2.4.

Figure 2.4: Translation of a point  $q$  to  $q'$ Figure 2.5: Scaling of a point  $q$  to  $q'$ Figure 2.6: Counter-clockwise rotation of a point  $q$  to  $q'$  by  $90^\circ$

## 2.5 Camera Parameters

We can be more specific about what  $M$  is for a camera taking a picture.  $M$  is a combination of two matrices that represents different parameters that belong to the camera. Those parameters are classified as camera intrinsics and extrinsics. The intrinsics represent the various distortions and qualities intrinsic to the camera itself. The main values we care about are the focal length, field of view, and optical center, that is, the distance between the lens of the camera and the sensor, the field of view for the camera, and the pixel location for the center of the image. This allows us to take an object in three-space and map it onto pixels in two-space.

We aren't finished yet, because we have no frame of reference to where the camera exists in the world; thus, we need the extrinsics. The extrinsics describe the camera's relation to the world origin. We want to be able to describe position in the world with coordinates that are convenient for our application. We imagine, then, an origin point in our world with the three orthogonal directions of the coordinate axes. Were we only dealing with the mathematics of a single pinhole camera, it would be most convenient to place the origin at the pinhole, and have the  $X$  and  $Y$  directions parallel to the wall of the projected image, and the  $Z$  direction perpendicular to the  $X$ - $Y$  plane. Generally, for camera placement, the extrinsic parameters describe the translation and rotation necessary to transform a description of world points to this frame of reference. The camera extrinsics are then a rotation and translation matrix. We will cover the calibration part later in this chapter; but the takeaway from this moment is that with these intrinsics and extrinsics that we can calculate through calibration, we can treat the cameras as a “black box” and toss in information about objects in 3D space into the appropriate matrices and the camera will spit out a corresponding image (or collection of pixels representing 3D space). More importantly, we can do this process in reverse, pointing out areas of interest on the cameras' images, and retrieve information on where those areas are represented in 3D space. Extrinsic

parameters are commonly represented as  $[R \mid t]$ , as in, the concatenated matrices of the rotation matrix (which is  $3 \times 3$ ) and the translation matrix (which is  $1 \times 3$ ). The resulting matrix  $[R \mid t]$  is a  $3 \times 4$  matrix.

The intrinsic parameters of a camera are also represented with a matrix:

$$\begin{bmatrix} \text{focal}_x & \text{skew} & \text{center}_x \\ 0 & \text{focal}_y & \text{center}_y \\ 0 & 0 & 1 \end{bmatrix}$$

We call this matrix the “camera matrix” and it is represented as  $K$ .

The mathematics to map a point in three-space to pixel coordinates is as follows:

$$\begin{aligned} p' &= Mp \\ p' &= K \begin{bmatrix} R & | & t \end{bmatrix} p \\ \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} &= \begin{bmatrix} f_x & s & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \end{aligned}$$

Cameras are not perfect, however. Lenses contain distortions as light travels through the lens unevenly, especially when going from the center of the lens to the edge of the lens. There are two typical distortions present in our calculations: radial distortion—often dubbed the “fish-eye effect” or, inversely, the “pincushion effect”—and tangential distortion, created by a misalignment of the lens and the image plane. This can be calibrated for, and corrected relatively easily. Lenses have three radial distortion coefficients and two tangential distortion coefficients that can be calculated during calibration and corrected through scaling the  $x$  and  $y$  coordinates of pixels to

compensate.

The final modification we will make to our camera model is to establish a “projection plane” for each camera (this will come up again). In short, we create a plane that is a focal length away from the lens in the opposite direction of the camera sensor. Unlike the example in Figure 2.1, where the resulting image is flipped, we are creating a plane for the image to appear “unflipped”. This allows us to visualize and explain higher-level concepts more simply, as we don’t have to internally flip every image we see.

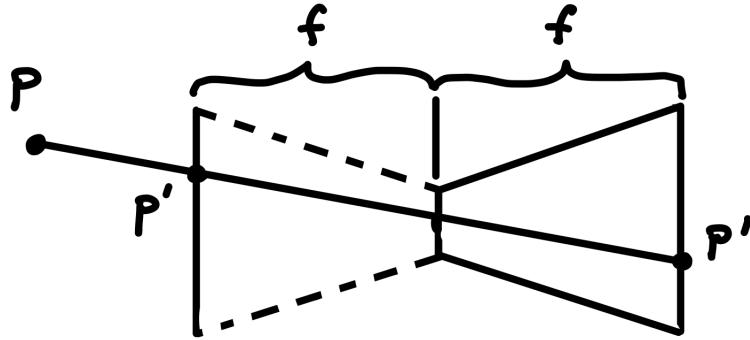


Figure 2.7: Flipping the Projection Plane

## 2.6 Stereo Normal Triangulation

Triangulation is relatively straightforward if the cameras are stereo normal (the cameras are adjacent, and on the same plane of the  $X$  and  $Z$  axes). Professor at University of Bonn, Cyrill Stachniss provides a great example, which I will modify for this use case.<sup>3</sup>

<sup>3</sup>. Cyrill Stachniss, dir., “Lecture: Photogrammetry I & II (2021, Uni Bonn, Cyrill Stachniss) - YouTube” (Uni Bonn, 2021), accessed April 4, 2025, <https://www.youtube.com/playlist?list=PLgnQpQtFTOGRYjqjdZxTEQPZuFHQa7O7Y>.

Imagine two cameras  $c_1$  and  $c_2$  which share the same  $Z$  and  $Y$  axis, pointing in the same direction (specifically the  $Z$  axis), with a distance  $d$  being the distance between the two cameras on the  $X$ -axis. Now, imagine a projection plane that is infinitely large, is perpendicular to the view direction to the cameras, and located a distance  $f$  from the cameras on the  $Z$  axis. To finish setting up, draw lines  $z_1$  and  $z_2$  perpendicular to the projection plane that go through both cameras respectively. A top-down view of this setup would result in Figure 2.8.

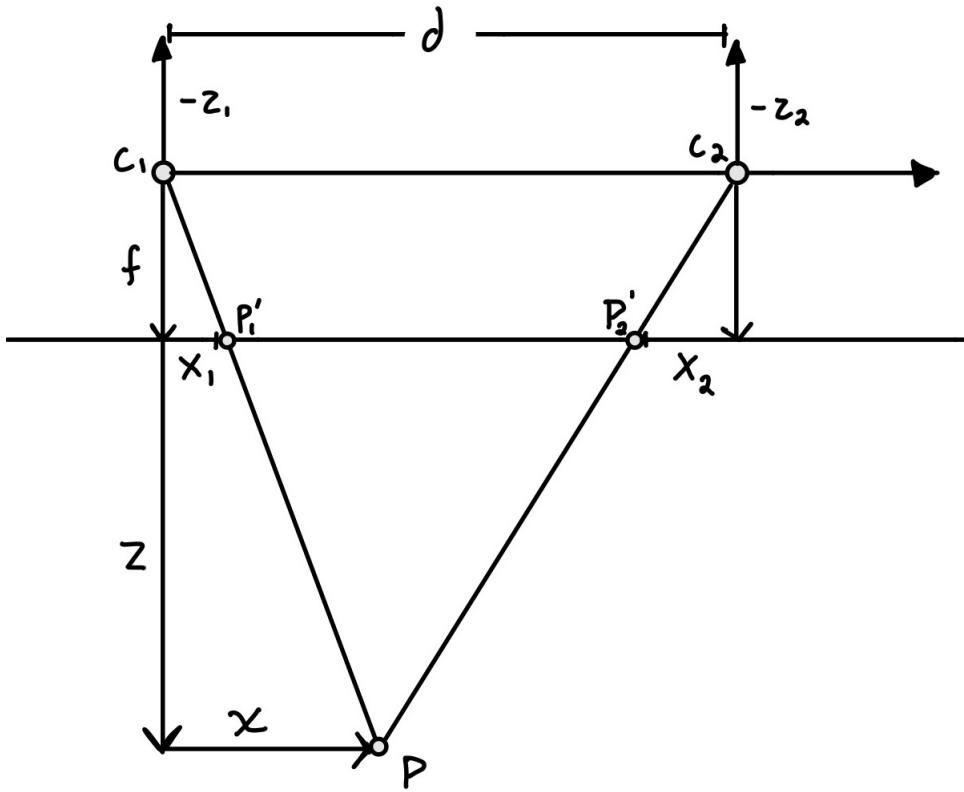


Figure 2.8: A Stereo Normal Example

For our stereo normal example, we are going to continue to use this view in our calculations, as we can assume any  $y$  or height information in the image will remain consistent between the two cameras (and in 3D space). This allows for a more simplistic base case, and we can elaborate on how further examples require more thought.

Now, imagine that, on the projection plane, there exists two points that map to

where  $c_1$  and  $c_2$  each detect a point  $p$  on each camera's sensor. Because we imagine the projection plane to be infinitely large, we map the pixel locations onto this plane using the projection, translation, and rotation matrices we were given in the initial calibration process. We will name these two points  $p'_1$  and  $p'_2$  which map to cameras  $c_1$  and  $c_2$  respectively. If we draw a line from  $c_1$  through  $p'_1$  and the same for the other camera, the point at which both cameras intersect will be the location of  $p$ . To get that result, the math is pretty straightforward.

First, note the distance from  $p'_1$  to  $z_1$  and  $p'_2$  to  $z_2$  on the projection plane which we will label  $x_1$  and  $x_2$ . Through the intercept theorem, we can find the  $z$  coordinate of  $p$  by solving the following relation:

$$\frac{z}{f} = \frac{d}{-(x_2 - x_1)}$$

Once that's done, through Law of Similar Triangles, we can find the  $x$  coordinate of  $p$  using  $z$ . That relation is as follows:

$$\frac{x}{x_1} = \frac{z}{f}$$

Finally, we can assume that the  $y$  value of the coordinates is whatever the  $y$  pixel value maps to the projection plane. This value is less important for placing the light as we will assume performers stay within the same height for this example.

## 2.7 Camera Calibration

Camera calibration is largely done automatically through OpenCV. We use Zhang's method.<sup>4</sup> The goal here is to gather the extrinsic and intrinsic parameters of the

---

<sup>4</sup> Z. Zhang, "A Flexible New Technique for Camera Calibration," *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22, no. 11 (November 2000): 1330–1334, ISSN: 01628828, accessed April 30, 2025, <https://doi.org/10.1109/34.888718>, <http://ieeexplore.ieee.org/document/888718/>.

camera. Remember, the extrinsic and intrinsic parameters of a camera allow us to map an object in three-dimensional space onto a two-dimensional plane. To start, we want to determine the intrinsic parameters for each camera. We can assume that each camera is at the origin of their its world. The high-level explanation of this process is to choose points whose locations are known in two-dimensional space (the image) and in three-dimensional space (the world). Using those locations in both two and three-dimensional space, and the knowledge that the camera is at the origin, we can reverse engineer the camera matrix (and its parameters). However, this method is easier said than done. The current issues we face is that we want a fast way to calibrate a camera, and we may not have points measured out (in three-space where the camera is the origin).

Zhang's method relies on the usage of a checkerboard with defined parameters; these parameters are the size of squares and the size of the grid (e.g. a  $3 \times 5$  grid). We assume that the checkerboard is planar—or flat—on some surface. If we are able to detect the corners of the squares on the checkerboard, we could pick a single corner and know where all other corners exist in relation to it. Our first trick is setting the world coordinate system to the bottom-left corner of the checkerboard, with the  $X$  and  $Y$  axes aligned with the bottom and left sides of the checkerboard and the  $-Z$  axis being normal to the checkerboard. Then, for all corner locations on the checkerboard, they all have a single thing in common. The  $z$  coordinate of each point in three-space is 0 (or all points lie in the  $X$ - $Y$  plane). This trick simplifies the current matrices by a significant amount. Recall that the way to convert a three-dimensional

point to a two-dimensional one is as follows:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & s & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

The board serves as the  $X$ - $Y$  plane at  $z = 0$ , with the  $Z$ -axis perpendicular to this plane. Thus, we can simplify the above as follows:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & s & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & t_1 \\ r_{21} & r_{22} & t_2 \\ r_{31} & r_{32} & t_3 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

During this process, we take multiple photos of the checkerboard in view of the camera. For each image we take, we find the bottom left corner of the checkerboard and establish it as the world origin. Then, knowing the size of the checkerboard squares in the real world, we generate a version of the above matrices for each corner in the checkerboard, where  $K$  and  $[R | t]$  are the same. At this point, all we care about is what  $p$  and  $p'$  are. The OpenCV library takes the checkerboard locations as input. Using its implementation of Zhang's method—which we will treat as a black box for brevity's sake—we can complete our calibration of both cameras.

## 2.8 Generalized Triangulation

The calculation we use to compute a three-dimensional location  $p$  from two camera images  $p'_1$  and  $p'_2$  is taken from a blog post of Batpurev, a method he refers to as Direct Linear Transform—or DLT.<sup>5</sup> This process is also specifically described in a

---

<sup>5</sup>. Temuge Batpurev, “Stereo Camera Calibration and Triangulation with OpenCV and Python,” Temuge’s webpage, February 2, 2021, 10:00 a.m. (Z), accessed May 1, 2025, <https://temugeb.github.io/stereo-camera-calibration-and-triangulation-with-opencv-and-python/>.

related post.<sup>6</sup> Essentially, we know that if we knew the location of  $p$  and the photo capture process was without error, that

$$p'_1 = M_1 p$$

and that

$$p'_2 = M_2 p.$$

Here  $M_1$  and  $M_2$  are the matrices representing the two cameras. Looking at each of these two equations individually—noting that these representations are in homogeneous coordinates—we seek non-trivial solutions to the following:

$$p'_i = \alpha_i M_i p$$

for  $i = 1, 2$ . We assert that the vectors  $p_i$  and  $M_i p$  are parallel by writing.

$$p'_i \times (M_i p) = 0$$

Batpurev relaxes this, allowing for error in sensor position. By setting the goal to finding  $p$  with  $\|p\| = 1$  where

$$p'_i \times (M_i p) = w_i \quad (2.1)$$

for  $i = 1, 2$ , particularly one that minimizes  $\|w_1\|^2 + \|w_2\|^2$ . According to the Hartley-Zisserman text<sup>7</sup> we can solve this formulation by taking the singular value decomposition (SVD) of a matrix  $A^T A$ , where  $A$  contains the cross products encoded in (2.1).

---

[io/opencv/python/2021/02/02/stereo-camera-calibration-and-triangulation.html](https://temugeb.github.io/computer_vision/2021/02/02/stereo-camera-calibration-and-triangulation.html).

6. Temuge Batpurev, “Direct Linear Transforms (DLT),” Temuge Batpurev, February 6, 2021, 10:00 a.m. (Z), accessed May 1, 2025, [https://temugeb.github.io/computer\\_vision/2021/02/06/direct-linear-transforms.html](https://temugeb.github.io/computer_vision/2021/02/06/direct-linear-transforms.html).

7. Richard. Hartley and Andrew. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. (Cambridge, UK ; Cambridge University Press, 2003), p. 585-587, ISBN: 0-521-54051-8.

In summary, using SVD as part of DLT, we can take our two points on the image planes, and find a point in three-space where  $w$  (or the distance between the two vectors) is minimized. DLT and SVD are provided within the code, which allows us to treat it like a black box.

## 2.9 From Pixels to Position

Finally, we are in a position to cover the pipeline that exists in this project. We begin with setting up the cameras in the performance space and calibrating the cameras to the space. The program contains a calibration mode which is modified from Dr. Batpurev's work in stereo calibration.<sup>8</sup> The script takes advantage of the OpenCV library which has in-built functions to calibrate the cameras individually and as a stereo pair through Zhang's method. We take a couple of photos for each calibration step.

Once that calibration is complete, we need to calibrate the cameras to be sensitive to only the LED that we hope to have tracked. Modern representations of images are as a matrix of three vectors, where each pixel corresponds to a set of red, green, and blue (RGB) values (sidenote, OpenCV defaults to a format of green, red, blue (GRB) values). The webcams that are used in this project operate at a resolution of 1080 pixels by 720 pixels, so to analyze a single frame of video data, the computer needs to process 777,600 three vectors of GRB data. In order to aid the software, we want to filter out the data we don't care about.

I used a blue LED light as a beacon because it was visually distinct. In my testing, red and green LEDs resulted in worse masks. That way, I could filter out pixels that don't appear as shades of blue. This technique is called "masking", and is pretty straightforward: First, we need to discuss a different representation of color referred to as HSV. While BGR (and other representations of RGB color) store the

---

<sup>8</sup>. Batpurev, "Stereo Camera Calibration and Triangulation with OpenCV and Python."

image as representations of the amount of red, green, and blue appear in a pixel, HSV approaches color through hue, saturation, and value. Hue represents the color, and is a value from 0-359.

This is akin to the hue being represented as a circle. Our perception of color is cyclical. The colors that we are sensitive to range from red to violet. However, we perceive violet as a mix of red and blue, which means that we could represent the colors we can see as a circle, with violet connected to red. Colors that feature red range in the hue values of 0-60, green in the range of 60-180, blue in the range from 180-300, and back to red in the range of 300-359. The peak or most pure forms of red, green, and blue are at hues 0, 120, and 240, and allows for easy translation between RGB and HSV. Saturation represents the distance from white a color sits, and is a range from 0-100. Value is the brightness of a color, and is a range from 0-100.

HSV is actually great for image processing because it interprets color similarly to the way the brain perceives color. When looking For example, we do not see a color as a specific mix of red, green, and blue light, and when comparing two shades of the same, or similar color, we may say that one is darker or lighter, which HSV primarily encodes. In this use case, if we want to mask the image down to a certain color, we can first convert all the pixels to an HSV format, then go pixel-by-pixel, and look for pixels that fall within a certain range of hue. Because the LED light reflects off of objects, we can also filter by saturation and values to remove darker blues and focus purely on the brighter shades coming directly out of the LED.

When we have the masked frames, the next step is to run a blob detection algorithm. The basic idea is to search the mask for blobs. A high-level description of the algorithm is to search the mask for pixels of similar color. Because the mask results in an image with only fully black or white pixels based on the hue of the LED, we reduce the workload for the algorithm, as it now only needs to distinguish between two colors. The blob detection algorithm outputs pixel coordinates at the center of



Figure 2.9: Left: Image of LED, Right: Mask of LED by HSV

each and every blob it notices.

We can input these pixel coordinates into our DLT solver. Because we calibrated the cameras, the vast majority of the work is done already. We know the camera matrix and the rotation-translation matrix for each camera, so we can input the pixel location on each camera and get a point in three-space representing the location of the LED.

Now that we have this location, the final step is to pick a location for the instrument and to transmit it to said instrument. Theoretically this is pretty easy. If we know the instrument's position in relation to the world we have defined with the cameras, we can convert the location of the LED to polar coordinates. At this point, there are defined values that exist for the lighting instrument to use, and all that is left is to transmit it. I take advantage of the numerous existing Python libraries that exist that allow someone to transmit a universe of data through a serial port. One of the most common serial ports that people encounter is the USB port (which stands for Universal Serial Bus). We can plug in an off-the-shelf USB to DMX decoder which will output a DMX signal that runs on a five-pin DMX cable to the instrument.



# Chapter 3

## Bringing the Project to Life (?)

The original idea was to demo the tracking software in Beier Li's thesis, *Chronicle of a Planet*. The show explores the various relationships between Belle and the concept of "mother". That took the form of three movements: her relationship with her mother, her motherland as a student from China, and her mother tongue of Mandarin.

The original goal was to include the beacon in the motherland movement. In that movement, Belle moves around a with white umbrella. The intended effect was to track the umbrella with a moving light in certain sections when Belle was criticizing the CCP most, to serve as the "watchful eye of a surveillance state." The show was set to open on February 14th.

### 3.1 The Process Begins

This process began as a dual-pronged approach. On the theater side, I started doing research into the history of lighting control. Meanwhile, I began to experiment with OpenCV, trying to pull the feed off of two cameras simultaneously. By fall break, I had set up the HSV mask and had the beginning of the pipeline. Unfortunately, that is where progress stagnated. I had pixel coordinates, but needed to figure out how to reverse that information into a three-dimensional position. During this time,

I was taking a computer graphics course, and we were focusing on projecting a three-dimensional scene into a two-dimensional image. If I could reverse that process, the triangulation of the beacon should be straightforward. I aimed to have a demo of the triangulation by the end of the semester, but did not meet that goal due to issues with my triangulation script as it output locations that implied me being impossibly far from the cameras.

I had ended the semester feeling a bit disheartened. At that moment I was hoping for a functional demo of the tracking software. Over that winter break, I spent some time digging around the internet to see if someone had done work with stereo vision to see what I was missing; this was when I discovered the work by Temuge Batpurev, Dr. Batpurev had spent time working with stereo vision, and had developed a script to calibrate stereo cameras. I hadn't calibrated for the distortion that a camera experiences through lenses and the fish eye effect, and I tried to rely on perfect placement of the two cameras with a measured distance instead of calibrating the two. Dr. Batpurev's script handled both of those problems.

I began the semester poking around Dr. Batpurev's code. His code relies on calibrating the cameras using a checkerboard with a specific size and row/column count. OpenCV provides infrastructure for detecting a checkerboard for this exact purpose. I integrated my HSV mask and blob detection with his calibration technique. He also had a script for DLT, which meant I could finally get a location out of the two cameras. The next step was to communicate this to a moving light.

## 3.2 DMX “Hell”

DMX is an interesting protocol. The protocol itself is a serial protocol, which means that bytes are transmitted sequentially, one bit at a time. Serial protocols are quite common in the world of communications; one of the most common known ports that

exist on a computer is the Universal Serial Bus (USB) port. USB is quite versatile, and allows for basically anything to be transmitted across it. There are speed limitations, but for a protocol as basic as DMX, this is a non-issue. While I had done some work in networking and protocols, my knowledge of implementing it in Python wasn't the greatest. Thankfully, through the magic of open-source software, there were multiple people who had created Python modules designed to work in DMX transmission. One of the defining characteristics of DMX was that every packet of data to be transmitted *had* to be 513 bytes; the first one is the *start code*. It denotes the way the signal should be interpreted (usually 0x00). The other 512 bytes communicate the behavior of the following 512 *addresses*, with one byte assigned to each address. Because of this, the majority of open-source software I encountered would have me either create the entire 513 byte packet, or to define what addresses were needed for the light (and what data it used).

The light I experimented with most was a ROBE Esprite, provided by Outlaw Lighting—a local supplier of lighting gear. It required, on average, 49 addresses to control various aspects of the light—my interest was specifically in the positioning of the beam through pan and tilt. That made for an interesting evening of staring at data sheets provided by ROBE, and attempting to create a consistent way to *serialize* all the attributes I needed. I ended up getting it working on its own, but I wasn't let off that easily. One of the main issues was dealing with traffic. An unforeseen challenge was managing the data traffic from two webcams providing input, and a third port providing DMX output.

### 3.3 So how did you get so close?

I felt strong going into the final week before tech for Belle's piece (the period where all technical aspects of a show are combined). I had a way to communicate with

the light, and I had a way to track the LED. Sure, I needed to still find a way to communicate the position of the LED to the Esprite, but even if I could just start to feed junk data, and just see what happens, it would have been awesome. Wishful thinking.

In order to try and speed up the tracking to be as close to real-time as possible, using OpenCV, I was able to downgrade the incoming image resolution from 1920x1080 to 640x480. That's a difference of 2,457,600 pixels to analyze to 307,200 pixels—a factor of eight. The tracking worked great when the beacon was close to the camera, but I was hoping for Belle to be dancing with the umbrella more than 15 feet from the camera, and with that resolution, the LED appeared only a few pixels wide if at all.

However, if I tried to upscale the resolution, it would scale the calculations by a factor of eight. The code as it is currently written struggled to keep pace. Additionally, when at a far distance, the LED would not show up on the camera feed.

Unfortunately, at that point, tech arrived, and I resolved to put the project on hold while I finished designing the non-live-tracking lights for Belle's thesis.

### 3.4 Where does that put us now

After discussing with my advisors, the best step forward was to try to work on the pipeline in sections to figure out the capacity, and to generate demos for each particular step. That way, even if the final, full pipeline never came to completion, I would still be able to showcase my work in the various sections. The three stages we imagined were as follows:

1. Tracking an LED with a stereo pair of cameras recording a series of three-dimensional positions.
2. Convert a series of three-dimensional positions to a series of moving light direc-

tions.

3. Program a moving light with DMX to move in that specified sequence.

If I can get all three stages working, then the pipeline should be trivial. This still means I need to work through calibrating the light, but I have some ideas on how to accomplish that. Although we are facing obstacles, this is a two-step forward, one step back process.



# Conclusions & Reflections

As of writing, this project stands incomplete. I was able to get some mapping of a 3D point onto a graph, but I continued to struggle to translate that position to a moving light. During the production process of *Chronicle of a Planet*, I was able to experiment with a ROBE Esprite, an industry standard moving light with the pan and tilt directly moving the luminaire. As we only had the budget to use the Esprite for the production, the remaining option was a Rosco ICue, which is an attachment that clips onto an ETC Source 4 Ellipsoidal. The ICue is a moving mirror, which uses a two-axis gimbal to move a mirror to reflect/redirect the beam of light. I was not able to write a calibration script for either moving light.

Had I the ability to start this process over, I would have started with Dr. Stachniss's Photogrammetry course, which would have allowed me to spend more time digging around under the hood of the black boxes used to calibrate cameras, and locate objects.

The idea is still a sound one, and absolutely accomplishable in the remaining time in the semester. It is just a manner of getting into the space and doing the work.

There are many places I hope to take this software upon its completion. For starters, the current implementation has communication with a lighting instrument in isolation from the rest of the lighting network. While this is not an issue for my specific project, it means that the instrument cannot be controlled by a conventional lighting console. One of my original dreams for this process was to create an injector/extractor

duo which took two DMX lines as input: one from the console, one from my software; the injector would interweave the signals together, encoding the signal with a different start code, so all other lights would ignore its instructions. The extractor would sit between the DMX chain and the moving light, and listen for instructions destined for the instrument with either start codes. Once receiving those instructions, the extractor would read the start code for the instrument, and forward the instruction.

The project is available on GitHub under an open source license. I have benefited much from the availability of code from my peers and the least I can do is pay it forward. This stereo vision also has more applications beyond just lighting tracking. With calibrated cameras, one could do all manner of responsive designs beyond lighting; there is possibility to use software like Isadora<sup>1</sup> to create video and sound designs that take the position of the performer into account.

As I finish this process, there is the concern that this process is paving the way for more humans to be taken out of the operation of theatre. This is not something to take lightly. However, I keep coming back to a conversation I had with my advisor in the final week. One can imagine a human-operated follow spot engaged in a duet with a performer onstage; timecoding has warped this duet, disconnecting the performer and the operator. While my software will not restore the operator's hand on the light, it at least attempts to bring the light and the person back in sync, this time just as a solo.

---

1. “TROIKATRONIX : ISADORA,” April 19, 2025, accessed May 2, 2025, <https://web.archive.org/web/20250419133332/https://troikatronix.com/>.

# **Appendix A**

## **The Code of the Project**

The code for this project can be found at: <https://github.com/gabeah/thesis>



# Bibliography

*ABBA – Super Trouper.* Accessed May 2, 2025. <https://genius.com/Abba-super-trouper-lyrics>.

“Automated Follow System — Zactrack Lighting Technologies GmbH — Wien,” May 1, 2025. Accessed May 2, 2025. <https://web.archive.org/web/20250501182028/https://www.zactrack.com/>.

Batpurev, Temuge. “Direct Linear Transforms (DLT).” Temuge Batpurev, February 6, 2021, 10:00 a.m. (Z). Accessed May 1, 2025. [https://temugeb.github.io/computer\\_vision/2021/02/06/direct-linear-transforms.html](https://temugeb.github.io/computer_vision/2021/02/06/direct-linear-transforms.html).

———. “Stereo Camera Calibration and Triangulation with OpenCV and Python.” Temuge’s webpage, February 2, 2021, 10:00 a.m. (Z). Accessed May 1, 2025. <https://temugeb.github.io/opencv/python/2021/02/02/stereo-camera-calibration-and-triangulation.html>.

Bentham, Frederick. *Stage Lighting*. In collaboration with Internet Archive. London : G. Allen & Unwin, 1980. Accessed May 1, 2025. <http://archive.org/details/stagelighting0000fuch>.

Birringer, Johannes. “Dance and Media Technologies.” *PAJ: A Journal of Performance and Art* 24, no. 1 (January 2002): 84–93. ISSN: 1520-281X, 1537-9477, accessed September 13, 2024. <https://doi.org/10.1162/152028101753401811>. [https://direct.mit.edu/pajj/article/24/1%20\(70\)/84-93/55164](https://direct.mit.edu/pajj/article/24/1%20(70)/84-93/55164).

“BlackTrax – CAST Group of Companies,” April 13, 2025. Accessed May 2, 2025. h  
[https://web.archive.org/web/20250413233816/https://cast-soft.com/blacktrax/.](https://web.archive.org/web/20250413233816/https://cast-soft.com/blacktrax/)

Bowman, Wayne, and Jean Bowman. *Modern Theatre Lighting*. New York: Harper, 1957.

Cadena, Richard. *Automated Lighting: The Art and Science of Moving Light in Theatre, Live Performance, Broadcast, and Entertainment*. Amsterdam ; Boston: Focal Press, 2006. ISBN: 978-0-240-80703-4.

Craig M. Carver. *A History of English in Its Own Words*. In collaboration with Internet Archive. HarperCollins, 1991. ISBN: 978-0-06-270013-1, accessed May 1, 2025. <http://archive.org/details/historyofenglish00carv>.

Culjak, Ivan, David Abram, Tomislav Pribanic, Hrvoje Dzapo, and Mario Cifrek. “A Brief Introduction to OpenCV.” In *2012 Proceedings of the 35th International Convention MIPRO*, 1725–1730. 2012 Proceedings of the 35th International Convention MIPRO. May 2012. Accessed April 4, 2025. <https://ieeexplore.ieee.org/document/6240859>.

*Entertainment Technology—USITT DMX512-A Asynchronous Serial Digital Data Transmission Standard for Controlling Lighting Equipment and Accessories*. ESTA, 2024. Accessed May 1, 2025. <https://tsp.estas.org/tsp/documents/docs/ANSI%20E1.11%20-%202024.pdf>.

Frenzel, Louis E. *Handbook of Serial Communications Interfaces : A Comprehensive Compendium of Serial Digital Input/Output (I/o) Standards*. Chantilly, UNITED STATES: Elsevier Science & Technology, 2015. ISBN: 978-0-12-800671-9. <http://ebookcentral.proquest.com/lib/reed/detail.action?docID=2189944>.

- Guarnieri, Massimo. "Switching the Light: From Chemical to Electrical [Historical]." *IEEE Industrial Electronics Magazine* 9, no. 3 (September 2015): 44–47. ISSN: 1932-4529, accessed October 4, 2024. <https://doi.org/10.1109/MIE.2015.2454038>. <http://ieeexplore.ieee.org/document/7271159/>.
- Hartley, Richard., and Andrew. Zisserman. *Multiple View Geometry in Computer Vision*. 2nd ed. Cambridge, UK ; Cambridge University Press, 2003. ISBN: 0-521-54051-8.
- Hays, David. *Light on the Subject : Stage Lighting for Directors and Actors –and the Rest of Us*. 1st Limelight ed. New York: Limelight Editions, 1989. ISBN: 0-87910-126-1.
- Hewitt, Barnard, ed. *The Renaissance Stage : Documents of Serlio, Sabbattini and Furtttenbach*. Translated by Allardyce Nicoll, John H. McDowell, and George R. Kernodle. Coral Gables, Fla. : University of Miami Press, 1958. ISBN: 978-0-87024-004-1, accessed December 9, 2024. <http://archive.org/details/renaissancestage0000unse>.
- "Home - Follow-Me," February 13, 2025. Accessed May 2, 2025. <https://web.archive.org/web/20250213084710/https://follow-me.com/>.
- "How Humans See In Color." American Academy of Ophthalmology, 6/8/2017 12:00:00 AM. Accessed May 2, 2025. <https://www.aao.org/eye-health/tips-prevention/how-humans-see-in-color>.
- Huntington, John. *Introduction to Show Control: Connecting Entertainment Control Systems for Live Shows*. NYC, USA: Zircon Designs Press, 2023. ISBN: 978-1-7357638-4-2.

Keller, Max., Johannes Weiss, Michael Robinson, and Andreas Klotz. *Light Fantastic : The Art and Design of Stage Lighting*. 2nd rev. and updated ed. Light. Munich ; Prestel, 2006. ISBN: 978-3-7913-3685-5.

“MIDI Show Control Specification.” Google Docs. Accessed April 29, 2025. [https://drive.google.com/file/d/1Hbix4KbuQRysnEF6L3sXhlwJtc4RE33L/view?usp=drive\\_link&usp=embed\\_facebook](https://drive.google.com/file/d/1Hbix4KbuQRysnEF6L3sXhlwJtc4RE33L/view?usp=drive_link&usp=embed_facebook).

“Opencv/Opencv,” April 4, 2025, 12:44 p.m. (Z). Accessed April 4, 2025. <https://github.com/opencv/opencv>.

“OSC Specification 1.0.” April 7, 2021. Accessed April 29, 2025. [https://opensoundcontrol.stanford.edu/spec-1\\_0.html#introduction](https://opensoundcontrol.stanford.edu/spec-1_0.html#introduction).

Penzel, Frederick. *Theatre Lighting before Electricity*. In collaboration with Internet Archive. Middletown, Conn. : Wesleyan University Press, 1978. ISBN: 978-0-8195-5021-7, accessed May 1, 2025. <http://archive.org/details/theatrelightingb0000penz>.

Pilbrow, Richard. *Stage Lighting Design : The Art, the Craft, the Life*. New York: By Design Press, 1997. ISBN: 0-89676-139-8.

Pulli, Kari, Anatoly Baksheev, Kirill Konyakov, and Victor Eruhimov. “Real-Time Computer Vision with OpenCV.” *Communications of the ACM* 55, no. 6 (June 2012): 61–69. ISSN: 0001-0782, 1557-7317, accessed April 4, 2025. <https://doi.org/10.1145/2184319.2184337>. <https://dl.acm.org/doi/10.1145/2184319.2184337>.

Serlio, Sebastiano. *Five Bookes of Architecture: Translated out of Italian into Dutch and out of Dutch into English*. In collaboration with Glasgow School of Art Library. Robert Peake, 1611. Accessed December 9, 2024. <http://archive.org/details/firstbookeofarch00serl>.

- Seul, Michael., Lawrence. O'Gorman, and Michael J. Sammon. *Practical Algorithms for Image Analysis : Description, Examples, and Code*. Cambridge ; Cambridge University Press, 2000. ISBN: 0-521-66065-3.
- Simpson, Robert S. *Lighting Control: Technology and Applications*. Taylor & Francis, 2003. ISBN: 978-0-240-51566-3. Google Books: gS1EfPUOTUoC.
- Stachniss, Cyrill, dir. "Lecture: Photogrammetry I & II (2021, Uni Bonn, Cyrill Stachniss) - YouTube." Uni Bonn, 2021. Accessed April 4, 2025. <https://www.youtube.com/playlist?list=PLgnQpQtFTOGRYjqjdZxTEQPZuFHQa7O7Y>.
- Stoker, Bram. "Irving and Stage Lighting." *The Nineteenth Century and After: A Monthly Review*, May 1911. Accessed May 1, 2025. <https://www.bramstoker.org/pdf/nonfic/lighting.pdf>.
- Swann, Alex. "Sound Design : A Systems Approach," Reed College, 2016.
- "TROIKATRONIX : ISADORA," April 19, 2025. Accessed May 2, 2025. <https://web.archive.org/web/20250419133332/https://troikatronix.com/>.
- Zhang, Z. "A Flexible New Technique for Camera Calibration." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22, no. 11 (November 2000): 1330–1334. ISSN: 01628828, accessed April 30, 2025. <https://doi.org/10.1109/34.888718>. <http://ieeexplore.ieee.org/document/888718/>.