GPMS: A Generative Projection Mapping System

[1]

University of Central Florida
Department of Electrical and Computer Engineering

Senior Design I
Dr. Chan

Group B
Victoria Moreno - Computer Engineering
Declan Carter - Computer Engineering
Francisco Soriano - Computer Engineering

Committee:
Reza Abdolvand, Saleem Sawahneh, George Atia

Table of Contents

Chapter 1: Executive Summary

The project introduced by the authors is a Generative Projection Mapping System (*GPMS*), an intriguing technology aimed at delivering projection mapping for everyday use. The authors desire to make immersive storytelling and entertainment accessible to hobbyists and entertainers alike. The GPMS initiative seeks to bridge the gap between high-end, professional projection mapping displays utilized by entertainment giants and the consumer market. The motivation is fueled by the potential to enhance personal and communal experiences through the innovative use of generative AI and computer vision algorithms.

GPMS is designed as a portable system capable of transforming any surface into a dynamic visual storytelling canvas. By accepting user inputs, GPMS leverages generative AI to produce images that are then projected onto surfaces, aligning with their unique features. The device is engineered to recognize the structure's vertices, edges, and designs, ensuring that generated images perfectly complement the physical space. It combines a projector with a responsive touchscreen monitor for easy use and advanced image processing algorithms for seamless image integration.

The primary goal of GPMS is to offer a user-friendly, customizable projection experience that allows for a high degree of personalization in how images are matched and displayed against targeted structures. Key objectives include ensuring the projector accurately displays images based on user prompts and allowing for precise alignment of generated images with the structure's features.

The GPMS system has broad applications that promise to revolutionize how stories are told and how experiences are shared among the general public. Furthermore, this system will make projection mapping an accessible tool for creativity. The project's ambition extends beyond its current scope, with plans to collaborate with local theme parks and potentially establish a startup, significantly impacting the entertainment industry and consumer technology.

GPMS aims to set a new and unique standard in immersive visual storytelling. This system represents a significant step towards making advanced projection mapping accessible, offering a blend of creativity, innovation, and personalization that has the potential to enrich everyday experiences.

Chapter 2: Project Description
2.1 Motivation and Background

The authors' interest in projection mapping stems from their passion for immersive storytelling and entertainment. However, due to the artistry, time, and technical nuance often required to tailor-make projection mapping shows, such displays are typically beyond the grasp of everyday hobbyists or entertainers. While theme park giants like Disney and Universal continue to improve large, multi-projector systems to render grand scenes onto their existing architecture, more needs to be done to put this technology into everyday consumers' hands. Individuals already look to purchase Sky Projectors to throw the night sky onto their ceiling, Christmas lights to celebrate the holidays, or disposable paper products to decorate their walls for festive occasions. However, by combining state-of-the-art computer vision algorithms and generative AI models, the authors look to introduce a new and unique product to allow people to share the joy of any occasion.

The authors propose a portable generative projection mapping system, *GPMS*, with the capability to map generated images onto any unique surface that will create a themed and immersive environment. This system will use generative AI to create these themes that precisely fit user input. This is done by GPMS aligning the major image features with the existing vertices on the surface. This process will be scalable using a central and distinct server to generate these images. As a result, the system can be employed relatively cheaply. The generative AI aspect of GPMS allows users to let their audiences experience immersive storytelling.

The authors take inspiration from the historical presence of Projection Mapping in local entertainment venues. It was not until 2011 that Disney World, located in Orlando, Florida, started projecting images onto Cinderella's castle with the help of 3D scanning and projection mapping. The emerging projections ran from 2011 to 2012 and even displayed visitors' photos of Cinderella's castle throughout select shows. Disney World has continued to incorporate more creative and new projection shows for years to come in light of projection mapping increasing in versatility and stability [2].

Additionally, Universal Orlando has transformed its projection mapping techniques through two breakthroughs within the last four years. The first breakthrough was the development of high-resolution projectors that can beam bright and crystal-clear images over long ranges. The second breakthrough was projection mapping itself, which ensured that projected images were not distorted when they were beamed onto uneven structures such as the Hogwarts Castle, one of Universal's more notable attractions. [3] As entertainment venues increase their use of projection mapping, a unique business opportunity arises for the authors. More specifically, after our final demonstration, the authors plan to sell GPMS to local entertainment venues in Orlando, FL with the condition that the projector must be returned unharmed. Additionally, the authors foster a hope to create their own startup from scratch shortly after the demo.

Projection mapping, being relatively recent, has been refined to project a multitude of images that can wrap themselves around various surface types for the images to look like they are part of the surfaces themselves. As projection mapping advancements continue to grow and expand, GPMS aims to directly engage and provide storytellers in all environments with a fresh way of delivering messages through the use of Generative AI for years to come.

2.2 Related Work

2.2.1 Dynamic Projection Mapping Research

A projection mapping solution presented by Addison Sandvik of the University of California Polytech State University outlines a similar approach to GPMS. This project demonstrates a proof-of-concept for a flexible projection mapping technique, allowing a surface to be placed anywhere within the projector's field of view for accurate image projection. The technique employs a surface equipped with infrared lights and a camera outfitted with an infrared filter to identify the location of the surface. Then, it utilizes a projective transformation to convert points from the camera's perspective into corresponding points in the projector's output [4].

This work demonstrates a few basic implementations of computer vision aspects, such as lens distortion correction using a pre-calculated matrix to flatten the rounded image and image calibration/alignment by transforming the corner points of the projected image to where the camera is able to view them. Both of these are essential aspects of the computer vision stack in this project. Additionally, in a few key ways, our implementation simplifies a few critical aspects as our projection surface is static as opposed to the dynamic mapping proposed by Sandvik. So, some pieces of his project can be well adapted to suit the needs of our calibration issues.

2.2.2 Industry Products: Luma Box

Many products in the industry do projection mapping, and some are even AI-generated. The best existing example of a product like this is Luma Box. Luma Box introduces an innovative approach to projection mapping through its product, Luma Map. This tool is designed to facilitate the creation of visually engaging projections that can be tailored to fit the unique contours of any surface, such as a house. The versatility of Luma Map lies in its ability to enhance physical features with light and shadows while integrating AI-generated themes. Luma Map presents a relevant case study for a project aimed at employing AI to generate images for projection onto buildings, thereby augmenting their appearance or introducing thematic elements. It exemplifies how technology can merge seamlessly with generative AI to redefine spaces, offering insights into the potential applications and benefits of similar AI-driven projects in projection mapping. [5]

2.2.3 Stable Diffusion with Control Nets

In August 2022, a breakthrough in machine learning was achieved with the release of the Stable Diffusion model. This innovative generative AI model has the unique ability to transform text into corresponding images. [6] A mechanism known as Control Nets is employed to generate related images from an existing one. These Control Nets enable the Stable Diffusion model to identify and learn the critical features of an image as determined by another model. Subsequently, a new image is generated based on these learned features. [7]

Control Nets is significant in our application of the Stable Diffusion model. This combination allows us to preserve the essential features of a structure and generate an image that incorporates these features along with user input. The combination opens up exciting possibilities for user-guided image generation. [7]

2.3 Project Overview

2.3.1 Basic Goals

Our project is centered around the creation of a device with the primary goal of creating an immersive storytelling environment. The overarching objective is to dynamically generate and project images onto a designated structure in alignment with its unique features. These structures' features may be pillars, windows, bricks, or even just a drawing. Therefore, the camera must be of good quality to best capture the features. The device adeptly detects vertices, edges, and designs on the target structure by leveraging sophisticated computer vision techniques.

The device integrates a responsive touchscreen monitor to facilitate user input and display any generated content. Users can input prompts, initiating the generation of images that will then be intelligently projected onto the unique structure/surface. Our cutting-edge device harnesses the power of generative AI to interpret user inputs and craft images that align seamlessly with the structure's features. An example would be asking for a Christmas design; the image could include candy canes for the pillars and snowflakes on the featureless areas. Alternatively, if the surface was a drawing of a dog, the dog might have a Santa hat on.

The device must have a high-powered projector for optimal visibility. Once the image is generated, the user must manually calibrate the projected image onto the structure. User calibration is achieved through the touch screen by moving the image's corners to the proper areas on the structure. The user will know if it is correct because the image is projected on the structure while they are calibrating.

2.3.2 Advanced Goal

In our pursuit of elevating user customization, our overarching goal is to increase the control users have over image representation, aiming for a significant level of precision.

Our specific objective is the implementation of sensitivity and threshold controls to achieve this goal. These controls will empower users, allowing them to intricately adjust the outline of the structure according to their unique preferences. For example, if the structure was a brick wall, but the user did not want their projected image to be based on each brick's outline, they can turn the sensitivity down. However, if the structure is a drawing of someone's face, the user can increase the sensitivity, allowing for the inclusion of each feature on the face.

This transformative experience is facilitated through a user-friendly interface, where individuals can precisely determine the desired similarity between the generated image and the structure. Our innovative approach ensures a seamless and personalized customization experience, setting a new user engagement and satisfaction standard.

2.3.3 Stretch Goals

As part of our stretch goals, we aim to allow users to highlight specific structural features. Our primary objective is to design an intuitive interface that allows users to effortlessly select and emphasize particular elements, ensuring a personalized visual experience. Users can achieve this by choosing features on the picture of the structure through the monitor. The image generated will closely align with the selected features.

Our second goal is to streamline the user experience. The objective is to integrate an automatic calibration process in the device, adapting to the features of walls or selected structures with minimal user input. Autocalibration ensures optimal image display without unnecessary complexity. Now, all the user has to do is set up the device, and the image properly aligns with all the features of the structure.

Below is a summary of our objectives.

Basic Objectives:
1. The projector seamlessly displays an image corresponding to the user input.
2. The generated image intelligently aligns with distinct features on the targeted structure.

Advanced Objectives:
1. Implementation of an input method to adjust sensitivity and threshold for image outlines.

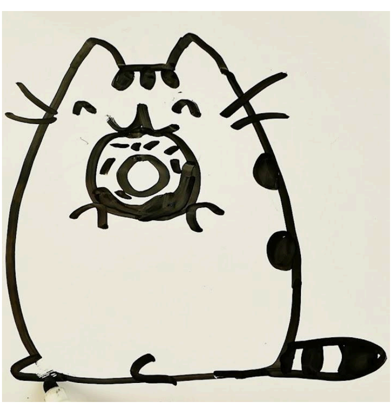
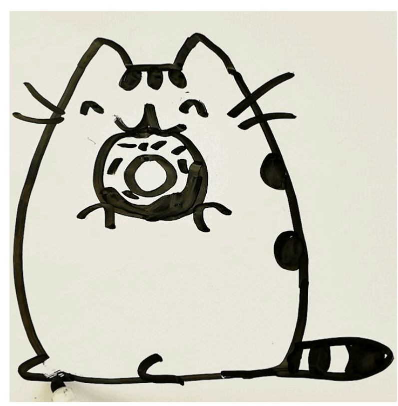
Stretch Objectives:
1. Develop a user-friendly frontend that empowers users to highlight features on the structure for emphasis.
2. Automatic calibration of the device to the specific features present on the wall.

2.4 Progress

2.4.1 Software Design Progress

Currently, we have developed early-stage image calibration and image generation prototypes. On a Raspberry Pi 3B model, we developed a basic image calibration implementation using OpenCV in Python. This initial prototype allows the user to select the corners that the projector is able to reach, effectively calibrating the image to be superimposed over what the camera is viewing. From there, Canny Edge detection is done on the camera view and displayed through the projector, now highlighting the edges and vertices seen via the edge detection algorithm. This process still needs to be tested, adjusted for barrel distortion in the camera lens, and automated with a better visualization of the process.

Next, concerning image generation, we have developed an initial pipeline utilizing Control Nets + Stable Diffusion XL with the open-source GUI Automatic 1111 [5] [6]. The table below, on the left, is an example of an original image. This particular image is used due to its explicit edges and cartoonish style. Next is an example of an image generated using this new pipeline, using the prompt “Cat eating a donut.” Finally, on the right is an animated GIF showing the tightness and accuracy of the edges in the original photo. This accuracy will be vital in aligning our images with existing features. This process needs further exploration to enable ease of use for the final user. Much of this will come from prompt engineering algorithms, whether a Large Language Model (LLM) or a custom database we design to find keywords in user prompts and expand on what we believe the user is seeking in their image. Finally, work should also be done to detect poorly generated images and regenerate them before they are returned to the user. Reprinted from *How to Draw Pusheen with Donut*. [8]

		
Original Image [8]	**Image generated w/ SDXL**	**Animated GIF to display similarities**

2.5 Requirements Specification
2.5.1 Technical Overview

To begin, the device is a projector with a camera and touch screen monitor attached and hardware to allow onboard calibration of the image to the target surface. This is done by taking a still frame of the surface, such as a wall, and taking another frame of the wall with a calibration image projected onto it, and moving the vertices of each corner such that the image is aligned properly; however, this would be a chance to utilize a custom PCB that would supply power to the selected MCU which in this case would be a Raspberry Pi 3B. Furthermore, this PCB would have buttons to control each of the corners to calibrate the image. From there, we use the camera and perform Canny edge detection on it, using an onboard Raspberry Pi 3B, to produce a line drawing, as seen in the image below. Next, connecting to a cloud or edge-based server, we send this image to be put through a generative AI model such as Stable Diffusion XL. On top of this model, we add layers for a control net; this ensures that the process we follow keeps in place the major features or vertices from the image and, subsequently, the wall. Finally, this data is sent back to the Raspberry Pi via the TCP connection we previously established, and the image is displayed on the projector using its newly calibrated set-up.

2.5.2 MCU and PCB Specifications

The main functionality of the custom PCB is to supply power to a microcontroller development board. As power is being transmitted to the MCU, LEDs will shine different colors on the custom PCB to indicate the on/off state of the power supply. Before the custom PCB is created, the proper and most suitable microcontroller must be carefully selected. After narrowing down the options, the authors compared an ESP32 and a Raspberry Pi. Each MCU’s specifications are briefly yet comprehensively listed below:

The ESP32 has the following features to integrate with the custom PCB [9]:

- Single or Dual-Core 32-bit LX6 Microprocessor with clock frequency up to 240 MHz.
- 520 KB of SRAM, 448 KB of ROM, and 16 KB of RTC SRAM.
- Supports 802.11 b/g/n Wi-Fi connectivity with speeds up to 150 Mbps.
- Support for both Classic Bluetooth v4.2 and BLE specifications.
- 34 Programmable GPIOs.
- Up to 18 channels of 12-bit SAR ADC and 2 channels of 8-bit DAC
- Serial Connectivity includes 4 x SPI, 2 x I2C, 2 x I2S, 3 x UART.
- Ethernet MAC for physical LAN Communication (requires external PHY).
- 1 Host controller for SD/SDIO/MMC and 1 Slave controller for SDIO/SPI.
- Motor PWM and up to 16 channels of LED PWM.
- Secure Boot and Flash Encryption.
- Cryptographic Hardware Acceleration for AES, Hash (SHA-2), RSA, ECC and RNG.

The Raspberry Pi 3B has the following features to integrate with the custom PCB [10]:

- Quad Core 1.2GHz Broadcom BCM2837 64bit CPU
- 1GB of RAM
- BCM43438 wireless LAN and Bluetooth Low Energy on board
- 100 Base Ethernet
- 40-pin extended GPIO
- 4 USB 2 ports
- 4 Pole stereo output and composite video port
- Full-size HDMI
- CSI camera port for connecting Raspberry Pi camera
- DSI display port for connecting Raspberry Pi touchscreen display
- Micro SD port for loading operating system and storing data
- Upgraded switched Micro USB power source up to 2

It is important to note that the overall project deliverable will benefit most from the microcontroller, which produces higher processing power, contains greater memory capacity, allows for direct video output, provides an operating system and multimedia support, and hosts various peripheral options. The comparisons of these factors are listed in the table below.

Main Factors	**Raspberry Pi 3B Features**	**ESP32 Features**
Higher Processing Power	Quad-Core 1.2 GHz Broadcom BCM2837 64-bit CPU	Single or dual-core LX6 microprocessor
Greater Memory Capacity	1GB of RAM	520 KB of SRAM
Direct Video Output	Full-Size HDMI port	LAcks build-in video
Operating System and Multimedia Support for image and video processing	Supports a wide range of multimedia software and libraries	Software environment is more limited and not optimized for multimedia processing
Peripheral Options in addition to WiFi and Bluetooth capabilities	Contains additional USB ports and extended GPIO for more peripheral connections	Only WiFi and Bluetooth capabilities

In summary, despite the ESP32 being a versatile and powerful microcontroller for a range of IoT applications, it does not suit the GPMS sufficiently. On the other hand, with a vast amount of available documentation on the Internet, the Raspberry Pi 3 B lends itself to me as the more suitable MCU for the GPMS. Its superior processing power, direct video output capabilities, and broader support for operating systems and multimedia applications make it a strong candidate for this project.

The custom PCB must have a reliable form of power supply input. Therefore, it will utilize two batteries to supply power to the Raspberry Pi and surrounding peripherals. After narrowing down the options, Alkaline batteries and Nickel Metal Hydride are the two battery types under consideration. Each of their specifications is listed in the table below. [11]

Chemistry	Description	Fully Charged Voltage (V)	Fully Depleted Voltage (V)	Capacity (mAh)
Alkaline	Non-rechargeable AA battery	1.65	1.4	1800 - 2700
NiMH	Rechargeable AA battery	1.45	1.2	1700 - 2000

Non-rechargeable batteries, such as Alkaline, have a lower initial purchase cost than rechargeable ones, making them more affordable for immediate use. They also exhibit a meager self-discharge rate, ensuring they retain their charge for a long time when unused, contributing to a long shelf life. Alkaline batteries are readily available in many stores and outlets, which makes them easily accessible for the authors to attain. Regarding pricing, those of type NiMH run at much higher costs. Therefore, fully charged Alkaline batteries are the clear choice regarding functionality and cost, given that they are moderately priced. [12]

It is important to note that the input voltage to the regulator must be between 2.4V and 3.3V. Therefore, the Alkaline battery is most suitable for the custom PCB since two Alkaline batteries in series will be able to supply 3.3V to the regulator for further use. Using two Alkaline batteries allows more current to be produced throughout the custom PCB, rather than the NiMH batteries, as shown in the table above. Furthermore, using Alkaline batteries allows the regulator's output current to reach 5V at 2.5A, which the Raspberry Pi 3B requires. [13]

2.5.3 Target Project Specifications

Category	Target
Size	12 inches x 12 inches x 5 inches
Weight	8 pounds
Power Transmission Efficiency	~ 2.5 hours
Display Accuracy	100% accuracy rate
Response Time	< 45 seconds
Visual Quality	1920 x 1080 pixels
Functionality	< 10% failure rate
Projection Range	> 6 feet
Calibration Autonomy	100% Fully Autonomous[1]
Recovery Time	< 10 seconds

[1] denotes a stretch goal

2.5.4 House of Quality

House Of Quality

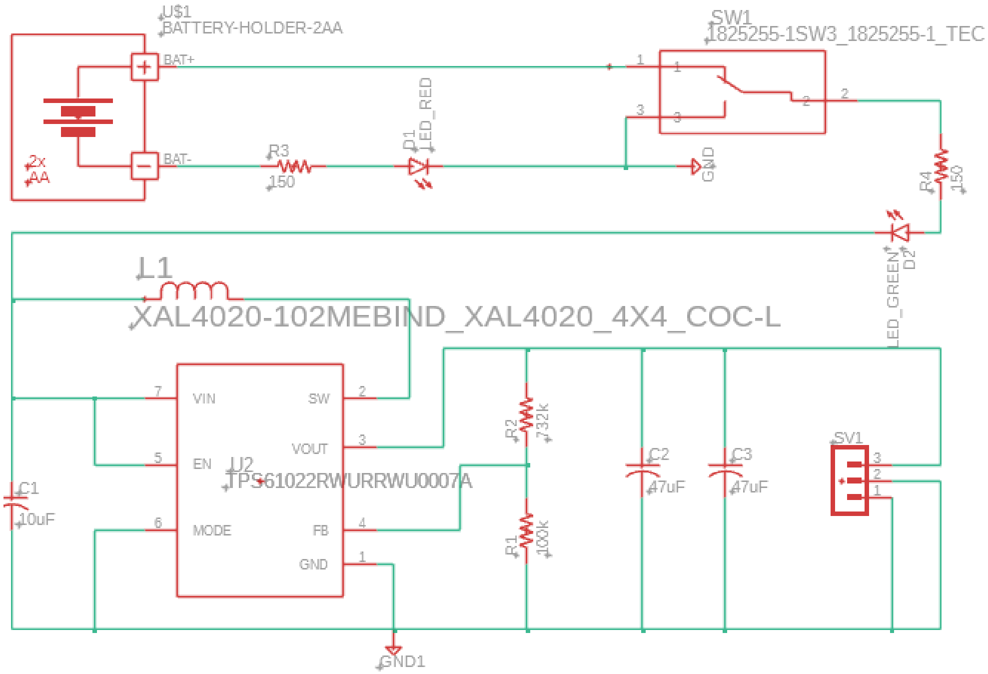
	Direction of Improvement		▼	▼	▲	▲	▲	▲	▲	▲	▲	▲
			Engineering Requiremets									
			Size	Weight	Power Transmission Efficiency	Display Accuracy	Response Time	Visual Quality	Functionality	Projection Range	Calibration Autonomy	Recovery Time
	Marketing Requirements	**Direction**	1	2	3	4	5	6	7	8	9	10
1	Ease of Use	▲	•	•	○	○	○	△	•	○	△	○
2	Cost	▼	•	•	•	•	•	•	•	•	•	○
3	Latency	▼	△	△	•	○	•	△	•	•	△	•
4	Connectivity	▲	△	△	△	•	•	•	○	•	△	○
5	User Safety	▲	○	○	△	△	△	△	△	△	△	△
6	System Display	▲	△	○	○	•	△	•	•	△	△	△
7	Portability	▲	•	•	△	△	△	△	△	△	○	△
8	Communication Range	▲	△	△	△	•	•	△	•	•	△	△
9	Ease of Maintainence	▲	•	•	○	△	△	△	•	△	•	△
10	Data Accuracy	▲	△	△	•	•	○	•	•	•	△	○
		Target	12" x 12" x 5"	8 lbs	~ 2.5 hrs	100%	< 45 seconds	1920 x 1080 pixels	< 10% failure	> 6 ft	Fully Autonomous	< 10 seconds

Correlations:
Positive: +
Negative: -
No Correlation: ~

Relationships:
Strong = •
Medium = ○
Weak = △

Direction of Improvement:
Maximize: ▲
Target: ☐
Minimize: ▼

2.5.5 Significant PCB Design Progress



A battery holder will house the two Alkaline batteries, which will be the power supply to the Raspberry Pi. The batteries are followed by an on/off switch to allow current to flow when switched on. A green LED will indicate when the switch is on, and a red LED will indicate when the switch is off. Following the switch is a DC/DC converter. The DC/DC converter will direct the incoming current to the 5V voltage level that the Raspberry Pi 3B requires into the 3-pin header that follows the regulator.

Given our 3.3V power supply, resistors can be used to limit the current flowing through each LED. In order to determine the resistor value required for each diode to avoid burning out, Ohm's Law was used with each LED's information.

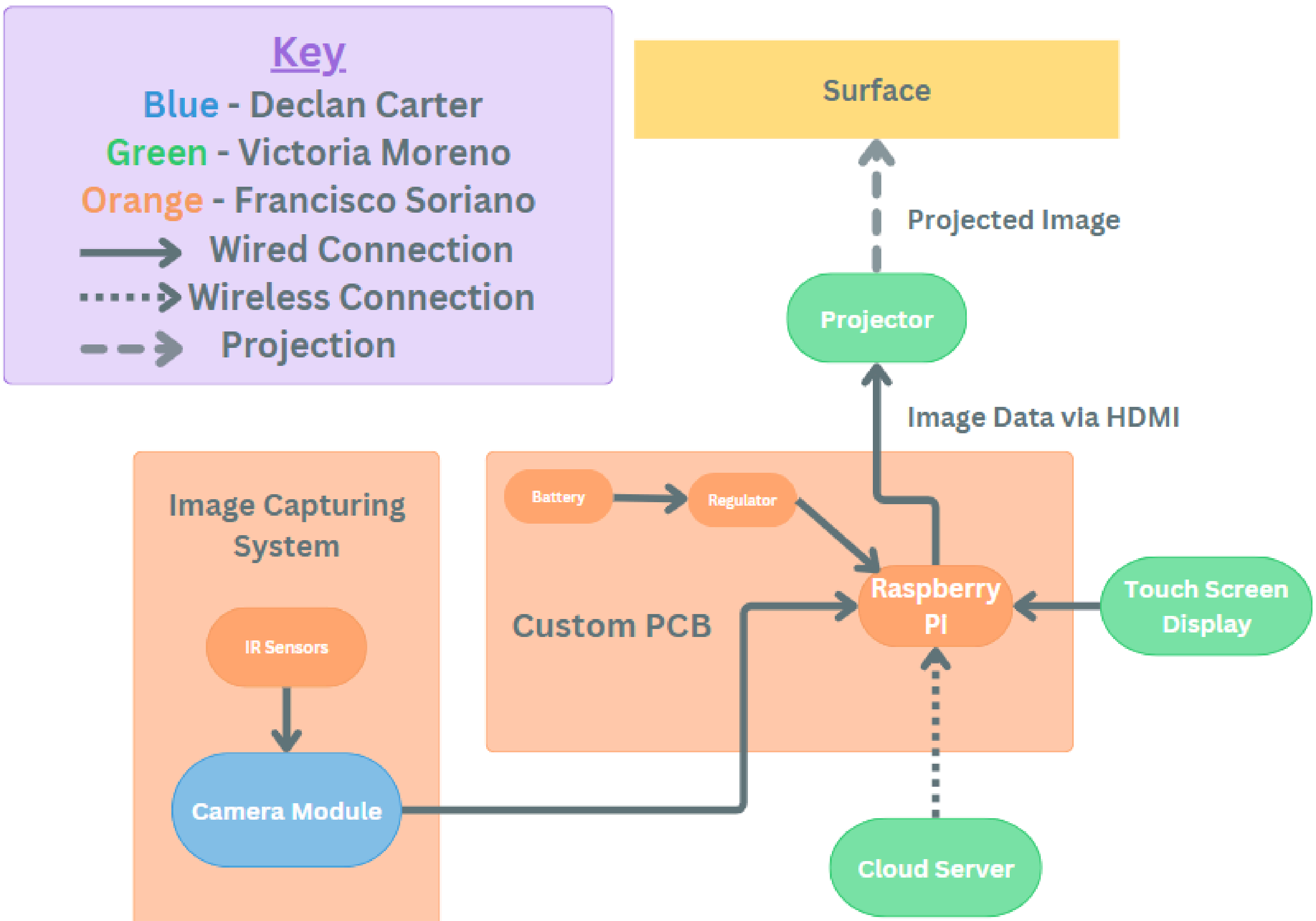
The equation needed to calculate the correct resistance value for each LED to avoid burning is: $R = (V_{source} - V_{forward}) / I_{Forward}$ where V_{source} is the voltage of the two alkaline batteries, $V_{forward}$ is the forward voltage of the diode, and $I_{Forward}$ current of the diode.

The green LED, which indicates that the power is on, has a forward voltage of 2.1V and a forward current of 30mA. The resistor value is computed at approximately 40Ω. [15] The red LED, which indicates that the power is off, has a forward voltage of 2.5V and a forward current of 30mA. The resistor value is computed as approximately 27Ω. [16] To summarize, 3.3V will not fry each LED if each corresponding resistor is used to limit the current to a safe level.

This 3-pin header will direct the output of the DC/DC converter into the Raspberry Pi via a Micro-USB cable. As the Raspberry Pi is powered up and taking in image data, the resulting output will be the input to the projector via HDMI. The Raspberry Pi 3B will be mounted onto the PCB. We will use JLCPCB to order the custom PCB. [14]

2.7 Diagrams

2.7.1 Hardware Diagram



2.7.2 Software Diagram

2.8 GPMS - An Illustration

The image to the right is generated using ChatGPT 4 to fuse the elements of modern technology into a single device. [1] It features a projector with two lenses, capturing multiple views using various sensor arrays.

Our image is sleek and futuristic, with a silver and black color scheme common in contemporary technology devices. The ports and ventilation grills lend to high functionality and processing power, which would be necessary for high-definition projection onto a 3D surface.

This image is a visual representation not available in the current market. It could be used for illustrative purposes during technical presentations of all disciplines and to envision future technology.

Chapter 10 Administrative Content
10.1 Budget and Financing

The prices provided are based on online research and manufacturer quotes and are, therefore, estimates. These costs are subject to change upon implementation of the final product. Currently, the project has no corporate sponsors, and the group will cover the majority of associated expenses aside from those specifically mentioned below. Should we attain further sponsors or donations, we will declare that immediately in the table below.

ITEM	QUANTITY	PRICE ESTIMATE ($)
Graphics Card[1]	1	$300
Google Colab[1]	6 months	$10.00 / month; ~$60
PCB	1	$20
Raspberry Pi[2]*	1	$40*
ESP32[2]*	1	$10*
Projector*	1	$200*
Camera	1	$50
Monitor	1	$100
Battery	2	$5
TOTAL (ESTIMATE)	**N/A**	**$505-$775**
AFTER DONATIONS	**N/A**	**$295-$535**

1

denotes 2 options for external compute we're actively researching for image generation

[2] denotes 2 options for onboard compute we're actively researching for calibration/UI

* denotes components gifted from friends or family

10.2 Project Milestones

The following dates in the table below are hard deadlines for our group as we aim to stay on track through weekly meetings to address individual concerns in order to maintain overall design advancements.

Number	**Milestone**	**Planned Completion Week (SD1 & SD2)**
1	Recruit Group Members	1/11/24
2	Project Decision	1/15/24
3	Individual/Collaborative Research	1/24/24
4	Divide & Conquer Document Done	2/2/24
5	45 Page Report Done	2/29/24
6	Finalize a BOM	3/15/24
7	90 Page Report Done	3/29/24
8	Discuss PCB Design	4/15/24
9	Discuss Algorithm Implementations	4/15/24
10	Submit SD1 Final Report After All Revisions	4/23/24
11	Test PCB	5/31/24
12	Prototype & Testing of Entire Project	8/20/24
13	Demonstration	12/1/24

Appendix
Appendix A - Pledge of Originality

We, the members of Senior Design Group B of the University of Central Florida, hereby pledge and affirm that our final report, submitted in partial fulfillment of the requirements for the EEL 4914C and EEL 4915L, represents our own original work.

All ideas and findings presented herein are the result of our collective effort and intellectual endeavor. Where we have consulted the work of others, it is duly acknowledged within the body and Appendix-B of the report, ensuring that credit is given where it is due. This document is a product of our collaboration as a group. Each member has contributed significantly to the conceptualization, research, analysis, and writing processes. We have openly shared knowledge and worked together to overcome challenges, ensuring a comprehensive and cohesive final product. We have adhered strictly to the ethical guidelines provided by our Faculty Mentors and the University of Central Florida's academic honor code. We have made every effort to document our methodologies, data sources, and analysis techniques transparently, allowing for the reproducibility and scrutiny of our work.

By signing this pledge, we commit to the truthfulness of the above statements.

[Signed]

Francisco Soriano Declan Carter Victoria Moreno

Appendix B - References

[1] OpenAI. “ChatGPT.” Chat.openai.com, OpenAI, 30 Nov. 2022, chat.openai.com/.
[2] *EVERY Disney World and Disneyland Castle TRANSFORMATION in HISTORY! | the Disney Food Blog*. 26 May 2020, www.disneyfoodblog.com/2020/05/26/every-disney-world-and-disneyland-castle-transformation-in-history/. Accessed 30 Jan. 2024.
[3] Sylt, Christian. “The Magic behind the Scenes of Universal’s Harry Potter Show.” *Forbes*, www.forbes.com/sites/csylt/2020/03/07/the-magic-behind-the-scenes-of-universals-harry-potter-show/. Accessed 30 Jan. 2024.
[4] Sandvik, A., & Zhang, J. (2023). *Dynamic Projection Mapping A Senior Project presented to*. https://digitalcommons.calpoly.edu/cgi/viewcontent.cgi?article=1377&context=cpesp
[5] Luma Box. (n.d.). Projection Mapping. Luma Box. https://www.lumabox.com/
[6] AUTOMATIC1111. (2022, November 19). *Stable Diffusion web UI*. GitHub. https://github.com/AUTOMATIC1111/stable-diffusion-webui
[7] lllyasviel. (2023, May 16). *ControlNet 1.1*. GitHub. https://github.com/lllyasviel/ControlNet
[8] *How to Draw Pusheen with Donut 🍩 Easy Drawing on a Whiteboard*. (2019, November 8). www.youtube.com. https://www.youtube.com/watch?v=zxxJWL2TZEA
[9] Teja, R. (2021, February 17). *Introduction to ESP32 | Specifications, ESP32 DevKit Board, Layout,.* Electronics Hub. https://www.electronicshub.org/getting-started-with-esp32/
[10] Raspberry Pi LTD. (n.d.). *Buy a Raspberry Pi 3 Model B*. Raspberry Pi. https://www.raspberrypi.com/products/raspberry-pi-3-model-b/
[11] Dr. Samuel Richie, Dr. Avra Kundu (2023, September). EEL 3926L Junior Design Part 2 - WEBENCH Exercise [Lecture notes]. University of Central Florida.
[12] How to Choose Batteries | REI Expert Advice. (n.d.). REI. https://www.rei.com/learn/expert-advice/batteries.html
[13] Negi, A. (2020, December 23). *Raspberry Pi 3 GPIO Pinout, Pin diagram and Specs in detail (Model B)*. https://www.etechnophiles.com/raspberry-pi-3-gpio-pinout-pin-diagram-and-specs-in-detail-model-b/
[14] *Raspberry Pi Documentation - Raspberry Pi Hardware*. (n.d.). Www.raspberrypi.com. https://www.raspberrypi.com/documentation/computers/raspberry-pi.html
[15] Reel, in. (2021). LTST-C190GKT in Reel by LITEON | LEDs | Future Electronics. Futureelectronics.com. https://www.futureelectronics.com/p/9894312
[16] Kingbright. (2021). APT1608SURCK. Retrieved from https://www.kingbrightusa.com/images/catalog/SPEC/APT1608SURCK.pdf

Appendix C - Copyright Permissions

Request For Reference Permission

Francisco Soriano
To: asandvik@calpoly.edu

Reply | Reply all | Forward

Thu 1/18/2024 3:48 PM

Good afternoon,

Hello, my name is Francisco Soriano and I am taking a Senior Design capstone course as a Senior Computer Engineering student at the University of Central Florida.

I would like to respectfully request permission to use your work - "Dynamic Projection Mapping" in our "Related Works" section in my Senior Design Project documentation report. Is this okay with you?

Best,
Francisco Soriano

Francisco Soriano
AMD Research Fellow
University of Central Florida - Computer Engineering
321-603-9123

Retention: UCF Delete after 10 Years (10 years) Expires: Sun 1/15/2034 9:57 PM

AS

Addison Sandvik <asandvik@calpoly.edu>
To: Francisco Soriano

Thu 1/18/2024 9:57 PM

Hi Francisco,

Yes, you may use my work as a reference.

Good luck!
Addison Sandvik

...

Appendix D - Code

```
# Declan Carter, UCF 2024

# Instructions for using the script:

# 1. Run the script to start the webcam feed in a window titled 'Frame'.
# 2. Select the four corner points on the displayed frame by clicking on them.
#    Do this counter clockwise, starting at the top left corner.
# 3. Once four points are selected, the script automatically calculates the
perspective transformation matrix and applies it to the webcam feed.
# 4. After the perspective transformation is applied, the script switches to
edge detection mode, displaying edges within the transformed area.
# 5. Use the following keyboard commands to interact with the script:
#    - 'q': Quit the program.
#    - 'o'/'p': Decrease/increase high threshold for edge detection.
#    - 'k'/'l': Decrease/increase low threshold for edge detection.
#    - 'r': Reset and select new corner points for transformation.

# Note: Ensure OpenCV is installed (`pip install opencv-python`) before running
the script.

import cv2
import numpy as np

# Initialize the webcam
cap = cv2.VideoCapture(0)

# Default corner points and state
default_pts_webcam = np.float32([[0, 0], [0, 480], [640, 480], [640, 0]])
new_pts = []
reset_mode = True
matrix = None

# Edge detection thresholds
low_threshold = 100
high_threshold = 200

# Mouse callback function
def select_point(event, x, y, flags, param):
   global new_pts
   if event == cv2.EVENT_LBUTTONDOWN and len(new_pts) < 4:
       new_pts.append([x, y])

# Reset points and enable mouse callback
def reset_points():
   global new_pts, reset_mode
   new_pts = []
   reset_mode = True
```

```
    cv2.setMouseCallback('Frame', select_point)

# Calculate the perspective transformation matrix
def get_matrix():
    # Points should be in the order: top-left, bottom-left, bottom-right,
top-right
    pts_projection = np.float32([[0, 0], [0, 480], [640, 480], [640, 0]])
    return cv2.getPerspectiveTransform(np.float32(new_pts), pts_projection)

# Apply Canny edge detection
def apply_edge_detection(frame, low, high):
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    edges = cv2.Canny(gray, low, high)
    return edges

cv2.namedWindow('Frame')
cv2.setMouseCallback('Frame', select_point)

while True:
    # Read frame from the webcam
    ret, frame = cap.read()
    if not ret:
        break

    # Apply transformation or edge detection based on mode
    if reset_mode:
        for pt in new_pts:
            cv2.circle(frame, tuple(pt), 5, (0, 255, 0), -1)
        if len(new_pts) == 4:
            matrix = get_matrix()
            reset_mode = False
    else:
        if matrix is not None:
            transformed = cv2.warpPerspective(frame, matrix, (640, 480))
            frame = apply_edge_detection(transformed, low_threshold,
high_threshold)

    cv2.imshow('Frame', frame)

    # Handle key inputs for threshold adjustments
    key = cv2.waitKey(1) & 0xFF
    if key == ord('q'):
        break
    elif key == ord('o'):  # Decrement high_threshold
        high_threshold = max(high_threshold - 10, 0)
        print("High Threshold:", high_threshold)
    elif key == ord('p'):  # Increment high_threshold
        high_threshold = min(high_threshold + 10, 255)
        print("High Threshold:", high_threshold)
```

```
    elif key == ord('k'):  # Decrement low_threshold
        low_threshold = max(low_threshold - 10, 0)
        print("Low Threshold:", low_threshold)
    elif key == ord('l'):  # Increment low_threshold
        low_threshold = min(low_threshold + 10, 255)
        print("Low Threshold:", low_threshold)
    elif key == ord('r'):
        reset_points()

# Release the webcam and destroy all OpenCV windows
cap.release()
cv2.destroyAllWindows()
```