GPMS: Generative Projection Mapping System

Francisco Soriano, Declan Carter, Victoria Moreno

Dept. of Electrical Engineering and Computer Science, University of Central Florida, Orlando, Florida, 32816-2450

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

***Index Terms* — Generative Projection Mapping System, Consumer-Grade Projection Technology, Generative AI, Computer Vision, Real-Time Image Alignment, Structure Recognition**

I. Introduction

GPMS was 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 structures, aligning with their unique features. The device was engineered to recognize a structure's vertices, edges, and designs, which ensured that generated images perfectly complemented the physical space. GPMS combined a projector with a responsive touchscreen monitor for efficient use and advanced image processing algorithms for seamless image integration.

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

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, which offers a blend of creativity, innovation, and personalization that directly enriches everyday experiences.

II. Overview

The following sections will take you through the entire lifecycle of the implementation of GPMS. Section III will cover the *System Components*, followed by Sections IV & V which cover the *Hardware & Software* respectively. Then, Section VI will cover *Testing* right before Section VII which covers the *Conclusion/Closing Remarks*.

III. System Components

A. MCU

The authors extensively weighed the options regarding the use of an MCU (Microcontroller Unit) on a custom PCB (Printed Circuit Board), a Development Board, and an SBC (Single Board Computer) for GPMS. To make an informed selection, they evaluated aspects including Hardware Integration, Processing Power, Peripheral Support, Software Development Environment, and Community Support.

SBCs, like development boards, offer significant advantages, including pre-integrated components such as high-quality cameras, simplifying hardware sourcing and ensuring compatibility. Additionally, SBCs and development boards provide more processing power than MCUs on a custom PCB, which enables faster image processing, AI inference, and real-time interactions. They support a wide range of peripherals (cameras, touchscreens, projectors) and protocols (SPI, I2C, HDMI), simplifying the connection of external devices. These platforms also come with comprehensive SDKs and libraries for rapid prototyping, debugging, and software optimization. [1]

Furthermore, popular SBCs and development boards benefit from large communities that provide resources like documentation, tutorials, and forums, which aid in innovation. Given these advantages, the authors ultimately chose an SBC for the GPMS project to streamline hardware setup and ensure compatibility, making it the final decision for this application.

B. PSU for Hardware

The authors evaluated three potential options for the Power Supply Unit (PSU) to power the Raspberry Pi 5: using a Boost Regulator on a custom PCB, a Power Bank, or a 5V Wall Outlet AC Converter. Initially, they had designed a custom PCB with a Boost regulator to elevate AA Alkaline batteries to 5 Volts (V) at 3 Amps (A). However, this design encountered challenges due to the

Pi's high current draw and power instability. Additionally, lithium batteries posed safety risks such as overheating and thermal runaway, leading to the rejection of this option.

Alternatively, they considered using a power bank as their PSU, which provides a stable 5V output and meets the 3A current requirement. Power banks also feature built-in protection circuits, ensuring consistent power delivery and portability, making them suitable for the Raspberry Pi 5. Consequently, the 'RoyPow 30W PD' portable charger/Power Bank was identified for its stable 5V output, sufficient current supply, and versatile features. However, this solution proved extraneous as the projector would also have to be plugged in. Also the battery solutions that offered sufficient power provision were unwieldy and deemed too large for the planned enclosure.

Furthermore, the authors explored the use of a 5V Wall Outlet AC Converter as a direct power source for the Raspberry Pi 5. This approach utilizes a reliable AC-to-DC adapter that converts mains electricity to a stable 5V DC output capable of supplying the necessary 3A current. Using a wall outlet AC converter eliminates concerns related to battery life and portability constraints, providing a continuous and stable power supply ideal for stationary setups. Additionally, AC converters often come with robust safety features, such as overcurrent and short-circuit protection, enhancing the overall reliability of the system.

In deciding among these options, the authors prioritized stability, safety, and ease of integration. While the power bank offered portability, the 5V Wall Outlet AC Converter provided a more reliable and maintenance-free solution for their application. Therefore, the authors concluded to utilize a 5V Wall Outlet AC Converter instead of the battery-based Power Bank for powering the Raspberry Pi 5 within GPMS, ensuring a consistent and dependable power supply. [2]

The Panseba projector within GPMS, an integral component carefully selected based off its capabilities/specifications, requires 120V and 60W to operate. The authors considered two options for powering the projector. The options were to either use a charger/power bank or plug the projector into a wall outlet which they concluded that the ladder would be more efficient given that this method of power supply provides a stable and consistent power supply, crucial for the projector's high-power requirements. Power banks cannot sustain the projector's necessary power levels for extended periods, potentially leading to interruptions and reduced performance. Wall outlets also ensure continuous power without the risk of running out of battery, keeping the Panseba projector operational throughout GPMS. This stable and reliable power supply is essential for the projector's optimal performance, especially when interfacing with the Raspberry Pi 5. [2]

C. OS

Every computer requires an Operating System (OS) to function effectively. Given that the computer that the authors were working with was brand new, selecting the appropriate OS became a crucial decision. Their familiarity with Linux prompted them to investigate whether there were compatible drivers for an AMD GPU, as Linux was their preferred choice over Windows.

While both operating systems have their strengths, Linux stood out in several key areas. Firstly, Linux is mostly free and open source, providing a cost-effective solution. Secondly, it offers extensive compatibility with AMD processors and has a large community with resources dedicated to running Stable Diffusion. In contrast, Windows requires paid licenses and has a smaller community compared to Linux when it comes to Stable Diffusion support. Additionally, Linux's open-source nature allows for greater customization and flexibility. Additionally, considering the author's existing familiarity with Linux and the compelling features it offers, the authors confidently selected Linux as the operating system of choice for the project. [3]

D. Server

The authors needed to choose a Linux distribution for their project. They focused on distros optimized for server hosting, narrowing it down to Ubuntu and CentOS. After comparing both, they chose Ubuntu for its server hosting capabilities, user-friendly interface, strong AI support, compatibility with AMD GPUs, and regular updates. Ubuntu's robust community support and extensive software range also made it ideal for the GPMS project.

Considering that the Raspberry Pi 5 cannot run Stable Diffusion, they explored server options: local versus cloud. Due to the University's network policies, setting up their machine as a cloud server accessible outside of UCF WAN would involve significant paperwork. Therefore, they opted for a local server setup, allowing the Raspberry Pi 5 to communicate with the AMD computer over the local wireless network. This approach simplified the setup, reduced latency, and avoided exposing ports on the University's network. [4]

E. Front-End Framework

Developing a tablet application requires careful consideration of various factors, such as compatibility with the target platform (Raspberry Pi 5), ease of network communication, and seamless OpenCV integration. The

team prioritized frameworks that offer straightforward OpenCV implementation, as it is crucial for the core functionality of the project to work without excessive additional effort. Other important capabilities included multi-touch support, performance, and ease of use. While the team was proficient in C++ and Python, the choice of programming language was not a limiting factor.

'Qt' stood out from the other frameworks due to its built-in support for OpenCV. This native integration made it highly convenient to incorporate OpenCV functionality into Qt applications without the need for additional bindings and/or libraries. Qt's extensive documentation and resources further facilitated the implementation process. In the context of OpenCV integration, bindings refer to the software layer that enables communication between the programming language and the OpenCV library. Bindings allow developers to access OpenCV functions and data structures from within their chosen programming language.

For example, PyQt/PySide and Kivy benefit from the availability of Python bindings for OpenCV (OpenCV-Python), which makes it easy to utilize OpenCV in Python-based applications. While Qt and Kivy share many similarities, they differ in their programming languages. Qt is built on C++, known for its performance and low-level control, while Kivy is based on Python, renowned for its simplicity and rapid development capabilities. Although Kivy excels in creating visually stunning and interactive user interfaces, Qt has a larger and more vibrant community. This extensive community support is highly advantageous, providing access help when issues arise.

After careful consideration, the authors decided to use Qt as their UI framework. The combination of Qt's native OpenCV support and extensive community resources made it the most suitable choice for GPMS. By leveraging Qt's strengths, the authors were then able to efficiently develop a robust and feature-rich application while benefiting from the framework's performance, cross-platform capabilities, and the vast ecosystem of tools and libraries available within the Qt community. [5]

F. API

The API's (Application Programming Interface) role within GPMS was to facilitate secure communication between a Raspberry Pi 5 and an AMD machine. This allows GPMS to send a text prompt and an image to the machine housed in the DRACO Lab within the University of Central Florida (UCF).

The AMD machine will generate two images using Stable Diffusion XL, a machine learning model for image generation, and return the images back to the Raspberry Pi 5. The API ensures efficient, secure transfer of data between the devices to carry out the image generation process. This API will be accessed exclusively by the three Senior Design 2 authors. Additionally, the API will only be accessible via the Raspberry Pi 5, which will use secure hashing for authentication.

The API will use any available port within the dynamic port range (50,000+). The API communication will be secured using 'HTTPS' (Hypertext Transfer Protocol Secure), a TCP (Transmission Control Protocol) protocol. A self-signed SSL (Secure Sockets Layer) certificate will be used to ensure secure connections, with unique hash-based handshakes. The API's service description begins with it built using Flask, a Python framework for creating web applications. It then accepts POST requests (data sent to the server) that include an image file (in JPEG or PNG format) and a text prompt. In response, it generates two images using Stable Diffusion XL and returns them as binary PNG files.

The data being transmitted starts with the input and ends with the output. The inputs are the image files (PNG) and text prompts. The outputs are the two generated images as binary JPEG files.

Regarding security measures, HTTPS encryption ensures that data, including sensitive information like password hashes, are securely transmitted. Authentication is handled using hashed passwords, which are stored securely and never transmitted in plaintext. SSL encryption prevents unauthorized access, even within a Wide Area Network (WAN). Additionally, the duration of port usage will be only operational from September 5, 2024 to December 14, 2024.

Regarding security considerations in the realm of authentication and access control, the system will establish an HTTPS connection secured with a self-signed SSL certificate. Both devices (Raspberry Pi 5 and AMD machine) will share a private API key to generate a unique time-based hash, ensuring secure communication. The hash will be transmitted during the API request and authenticated using the private key and timestamp. This method prevents unauthorized access, especially from potential attackers on the UCF network who might scan for open ports to misuse system resources or generate inappropriate content.

The data being transmitted (images and prompts) is not inherently sensitive. However, the hash used for authentication must be protected. Therefore, as later discussed, the authors utilize SSL to create a secture pipe between the device and server. Without proper authentication, the server could be exploited to generate malicious/inappropriate content or waste compute resources. The authentication and access control

mechanisms mitigate this risk, ensuring that only authorized users can access the system.

The following software dependencies are required: Python 3.x, HMAC and hashing libraries, OpenSSL. The final demo will take place to hardwire the Raspberry Pi 5 directly to the AMD machine via Ethernet, to ensure functionality of GPMS in case the UCF Wi-Fi is not functional. [6]

IV. Standards

A. HDMI

HDMI (High-Definition Multimedia Interface) is a standard developed to facilitate high-bandwidth links among digital devices, supporting 1080p video and up to eight channels of uncompressed audio. It simplifies connections by reducing the number of cables and remote controls needed. All components in a system must be HDMI-compatible to fully utilize its features.

The Panseba projector includes an HDMI port for receiving and transmitting audio and video signals. HDMI can transfer high-bandwidth data via a single cable, making it efficient for high-definition signals.

HDMI uses Transition Minimized Differential Signaling (TMDS) to maintain signal integrity and High-Bandwidth Digital Copy Protection (HDCP) to prevent piracy. HDCP ensures that only authorized devices can decode transmitted data through a “handshake” process. Common issues include “handshake” failures and the length limit of HDMI cables. The standard requires a minimum length of 32 feet, but shorter cables are often used for various applications such as GPMS. [7]

B. TCP

The authors integrated TCP to establish a reliable client-server connection for GPMS. This connection is crucial for relaying packets from the Raspberry Pi 5 to the local server, where the generative AI pipeline processes the data and then sends it back to the Pi and then from there to the projector to display the image. TCP’s reliability and orderliness in data transmission are critical for the seamless operation of the project.

TCP ensures accurate and ordered delivery of data packets between networked devices. Its error-checking capabilities guarantee data integrity, while sequence control maintains the order of data packets, preventing incorrect visual outputs. TCP’s flow control mechanism helps prevent network congestion, ensuring smooth and responsive interaction between the user device and the server.

The three-way handshake that TCP establishes is a reliable connection, creating a stable pathway for data flow. Its congestion control algorithms dynamically adjust the rate of data transmission, preventing packet loss and minimizing latency. This ensures that the Generative AI pipeline remains responsive and efficient, providing seamless experiences for GPMS users. [8]

C. SSL

Integrating Secure Sockets Layer (SSL) is crucial for establishing a secure HTTPS connection between the client and server within GPMS. The use of a self-signed SSL certificate ensures that data integrity and confidentiality are maintained during transmission, essential for protecting sensitive information, such as API keys, from potential interception on UCF’s network.

SSL provides an encrypted channel over an insecure network, ensuring that both data and API keys remain confidential and secure. This comprehensive encryption framework mitigates risks of data exposure or tampering, thereby enhancing privacy. Furthermore, SSL authentication mechanisms, using certificate validation, ensure that data exchanges occur only with the intended server. By using HTTPS with SSL, GPMS securely manages various data streams without compromising security, a vital feature for operating the AI pipeline efficiently and safely. [9]

D. Projector

Projectors come in various shapes and sizes but adhere to key standards. Resolution, such as SVGA (800x600) to Full HD (1920x1080), determines image detail. More specifically, higher resolutions will provide sharper images.

Standards also define connection options like HDMI, VGA, and USB. Additionally, the lamp life indicates how long the projector lamp lasts, affecting ongoing costs with regard to the number of lamps that might be required for the projector. Moreover, the throw ratio determines the distance needed for the desired image size and color gamut refers to the range of colors a projector can display which are both important for photo projections.

Aspect ratio, such as ‘4:3’ or ‘16:9’, should match the content’s aspect ratio. Lumens measure brightness - meaning that higher lumens are better for brightly lit rooms. Understanding these standards assisted the authors in choosing the most optimal projector that met the needs of GPMS in terms of brightness, resolution, and compatibility. [10]

E. Cameras

There are two main standards for cameras: Measurement Standards and Environmental Standards. Measurement Standards, set by organizations like 'ISO', ensure consistency in reporting camera performance aspects like resolution, noise, and sensitivity. Environmental Standards define a camera's resistance to dust, water, and impact, indicating how well it withstands a variety of conditions.

'ISO 12233' measures resolution, 'ISO 15739' defines image noise, and 'ISO 12232' refers to ISO Speed Rating. The 'IK Rating' (IEC 62262) rates impact resistance, while the 'IP Rating' (IEC 60529) defines dust and water resistance. *NEMA* Standards in North America also indicate dust and water resistance levels. Understanding these standards assisted the authors to best select the most appropriate camera for the needs of GPMS. [11]

V. Design Constraints

A. Economic

The economic constraints of the GPMS project primarily revolve around the cost of the hardware components and the available budget. The total estimated cost for the project is approximately $3,544.37. However, thanks to component lending from AMD housed in the DRACO lab space, the out-of-pocket expenses for the authors were reduced to roughly $299.93.

The authors recognize that additional funding could have potentially enhanced the project's capabilities. For instance, a higher budget would have enabled the authors to acquire a high-end projector capable of projecting images at greater distances with increased visibility. Such a projector would have expanded the potential applications of the GPMS, allowing for larger-scale projections in various settings. Careful financial planning and management was carried out throughout the completion of GPMS within the given economic constraints while striving to maximize its capabilities.

B. Time

The GPMS project is subject to the time constraints imposed by the Senior Design joint course schedule, which presents several challenges for the project team. The entirety of the GPMS project was divided into two courses/phases: Senior Design 1 (EEL4914) and Senior Design 2 (EEL 4915L), each with its own set of deadlines and deliverables.

Senior Design 1 focused on completing all preliminary administrative tasks, such as project planning, research, and documentation, crucial for laying the foundation of the project and ensuring that all necessary research and planning are completed accurately before moving on to the implementation phase.

Senior Design 2 was dedicated to prototyping, testing, adjusting when needed, and debugging the entire project along with milestones which required significant effort and coordination from the team, as these milestones involved integrating various hardware and software components, conducting rigorous testing, and addressing issues that arose during the entire process. To have met these deadlines and effectively manage time constraints, the authors adhered to a strict schedule which involved regular communication involving progress checks with our faculty advisor.

C. Ethics

Ethical constraints existed in the development of GPMS, particularly with generative AI in image generation. Significant concerns included copyright infringement, plagiarism, and biases in AI-generated images. These issues highlighted the need for careful consideration of social implications.

For GPMS, user input has prohibited inappropriate and misleading images to ensure it remains suitable for all audiences. These measures maintain the integrity of GPMS, especially in family-friendly environments.

D. AMD Machine

During the research phase, the team explored various methods to establish communication between the Raspberry Pi 5 and the AMD machine, aiming for efficient and secure data transfer and execution of computationally intensive tasks on the AMD machine. However, several constraints and challenges emerged.

One primary hurdle was limited access to the AMD computer. The team collaborated with the DRACO Lab Principal Investigator and lab manager to gain access, but the process faced significant delays due to unforeseen issues and administrative complexities with UCF IT. This raised concerns about the feasibility of the proposed communication system, particularly the need to expose a port on the network for simplified API calls.

A major constraint was the restricted accessibility to the AMD computer, which could only be accessed remotely through SSH while connected to the school network. This required team members to be physically present on campus or use a VPN to establish a connection.

The most critical issue was the inability to safely open port on the AMD machine. The team initially planned to use API endpoints for seamless communication and data exchange, but this required opening a specific port on the school network. Given the strict security policies and

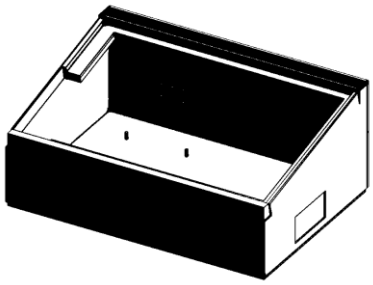
administrative regulations, obtaining permission to open a dedicated port was a formidable challenge. Despite the challenges, the authors were eventually able to gain approval, by UCF IT and the DRACO Lab, to open a port for API communication.

E. Figure & Tables

Section E presents all figures & tables related to the implementation and testing of GMPS which are elaborated in their respective sections.

Figure 1 presents a physical view of the enclosure GPMS. Figure 2 explains the Software 'Main Loop' for GPMS. Additionally, Tables 1, 2, 3, and 4 each summarize their respective data recorded during testing of GPMS.

GPMS Hardware Enclosure – Figure 1



Engineering Specifications – Table 1

Category	Target
Size	7 in x 10 in x 8 in
Weight	~10 lbs
Elapsed Time	**< 5 minutes**
Alignment Accuracy	**< 8 mm**
AI Generation	**< 1.5 minute/image**

Elapsed Time Data – Table 2

Target (300s)	Elapsed Time(s)
< 300	73
< 300	92
< 300	87
< 300	89
< 300	80
< 300	97
< 300	78
< 300	90
< 300	82
< 300	77
Mean without Gen	**84.5**
Standard Deviation	**7.64852927**
AI Image Gen	**180**
Mean with Gen	**264.5**

Alignment Accuracy Data – Table 3

Target (mm)	Average Accuracy (mm)
< 8	1.75
< 8	1.75
< 8	2.25
< 8	2.75
< 8	2
< 8	1.25
< 8	0.75
< 8	1.5
< 8	2.25
< 8	1
Mean:	**1.725**
Standard Deviation:	**0.6175**

AI Generation Time Data – Table 4

Target (min)	AI Generation Time (s)
< 1.5/image	42.6
< 1.5/image	43
< 1.5/image	42.8
< 1.5/image	43
< 1.5/image	43.1
< 1.5/image	42.7
< 1.5/image	43.4
< 1.5/image	42.8
< 1.5/image	42.7
< 1.5/image	42.9
Mean	**42.9**
Standard Deviation	**0.2357022604**

VI. HARDWARE

The Hardware Components of GPMS begin with integrating all the necessary and selected hardware parts into the project. The System includes, most notably, the Single Board Computer (the Raspberry Pi 5) as the brains of GPMS. This Raspberry Pi 5, is powered with the proper power ratings (5V at 3A) by a 5V Wall Outlet AC Converter.

A touch screen monitor what sends data to the Pi via USB and receives Image Data back from the Pi via HDMI. The Camera Module is then connected to the Pi via Serial Camera Interface (SCI). The camera module captures an image and sends it to the Pi while in return receives a 3.3V at 250mA power rating for operation from that same Pi. The Pi then connects via Wi-Fi using the TCP/IP protocols to a local server for the generation of AI images. The Raspberry Pi 5 then sends the retrieved image data via HDMI to the projector which is connected and

powered by the wall outlet. Finally, the projected projects the AI generated image on the surface/structure.

The Hardware Design of GPMS includes a 3D printed enclosure made from PLA filament to sit on top of the projector mounted on top of it using Velcro to prevent unwanted movement and/or instability. It housed the touchscreen monitor, Raspberry Pi 5, and camera. The enclosure accounted for marginal error printed by the 3D printer with regards to the dimensions of each of its components. The touchscreen monitor sat flush into the top of the enclosure as if it were a phone surrounded by a phone screen. The Raspberry Pi 5 was also securely fastened to the enclosure's floor using tape. Holes were used to fasten the camera to the front face of the enclosure using screws and nuts. The camera sat above the projector lens and on the same plane parallel to it.

Moreover, this enclosure, specifically the touchscreen monitor, hid all the cables to allow users to avoid the cable clutter when using GPMS. This enclosure, which can be viewed in Section V-E, had a side opening made for cables from other external hardware components to be able to enter and connect to the Pi on the inside.

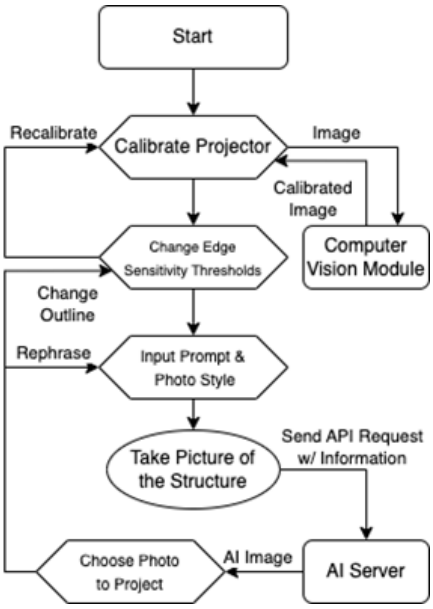
VII. Software

The GPMS software application was designed around three main components — the Main Loop, the Computer Vision Module, and the AI Server — that together enabled users to create context-aware projections from text prompts. When the application was launched, the first step was camera calibration, which corrected for lens distortion and defined the projection area. This initial calibration ensured images aligned accurately with the target surface.

Next, users set edge detection thresholds in the Computer Vision Module, which allows the system to identify surface edges on the projection space. With calibration/edge detection in place, users then provide a descriptive prompt for their desired image, which is then sent with processed surface data to the AI Server. The server then uses Stable Diffusion technology to generate two unique images based on this input. Users then choose the preferred image to be seamlessly mapped onto the surface, resulting in a precise, visually aligned projection.

The Main Loop, shown in Figure 2, is the primary interaction point where users start by selecting the corners of the projection area for camera calibration and then adjusting sensitivity thresholds for edge detection. After verifying alignment with projected edges, users input their prompt and photo style preferences (Realistic/Animated). This information is then sent to the AI Server which then returns two AI-generated images. After selecting an image, users see it accurately projected, mapped precisely to fit the calibrated area.

GPMS Software Diagram, Main Loop – Figure 2



Within the Computer Vision Module, lens distortion is corrected first. The system then projects a calibration image onto the surface, defining the viewable boundaries regardless of GPMS's position and/or angle, by warping the image as needed to fit the surface. Next, edge detection was then computed and projected onto the scene, giving users a chance to customize and fine-tune alignment before finalizing calibration.

The AI Server was the final processing point, where a still frame of the surface, along with the user's prompt and metadata, gets sent to the Python server via HTTPS POST. The server could expand the prompt if needed and establishes a "control net" that guides the Stable Diffusion model (in this case, Stable Diffusion XL) during image generation. The control net ensures that the major edges and features of the original surface were preserved, providing consistency across image generation iterations. Once the final image was ready, it was sent back to GPMS, where it was then projected onto the surface with high accuracy by the projector, creating a seamless experience powered by computer vision and AI.

VIII. Testing

Testing integrated hardware and software to produce results, using festive storylines to evaluate GPMS in an authentic setting. For accuracy in Section V.E tables, the authors focused on three key metrics: elapsed time,

alignment accuracy, and AI generation time—critical for assessing system performance and efficiency.

For elapsed time, GPMS targeted completing setup and image projection in under five minutes. The testing involved ten runs, measuring the total time using a stopwatch from initial setup through final image projection. Each run included setting up the projector, calibrating the projection area, adjusting threshold values, entering a prompt, and selecting an image. Excluding AI generation, times ranged from 1 minute and 13 seconds to 1 minute and 37 seconds, with a mean of 1 minute and 24 seconds. When factoring in AI generation (adding about 3 minutes), the mean total time was 4 minutes and 24.5 seconds (264.5), staying comfortably within the target.

The alignment accuracy test measured the precision of projected image alignment with the target surface. In ten test runs, alignment was measured at four different points on the target projection surface. The average accuracy across all runs was 1.1725 mm, with a standard deviation of 0.6175 mm, surpassing the target of 8 mm. This demonstrated the system's consistent precision and provided a reliable basis for high-quality image projection.

For AI generation time, GPMS aimed for under 1.5 minutes per image generation. In ten tests, times ranged from 42.6 to 43.4 seconds, averaging 42.9 seconds with a standard deviation of 0.23 seconds—well below the target, indicating stable performance. However, further analysis is needed to optimize overall delivery time, including image transmission back to the user. This ensures GPMS meets efficiency, precision, and speed goals, enhancing user experience and reliability.

Biography

Francisco Soriano will graduate from UCF having earned a B.S. in Computer Engineering on the VLSI Track. Francisco will join AMD Orlando as a full-time Silicon Design Engineer I. Simultaneously, he will continue his studies at UCF to pursue an M.S. in Computer Engineering on the VLSI Track at UCF.

Declan Carter will graduate from UCF having earned a B.S. in Computer Engineering on the Comprehensive Track. Declan will continue his studies at UCF to pursue an M.S. in Computer Engineering.

Victoria Moreno will graduate from UCF having earned a B.S. in Computer Engineering on the Comprehensive Track. Victoria will continue her studies at UCF to pursue an M.S. in Computer Engineering on the Intelligent Systems and Machine Learning track.

Acknowledgement

The authors acknowledge Dr. Chan and Dr. Weeks for their unwavering support throughout the GPMS project. Special gratitude goes to Dr. Borowczak and the DRACO Lab Managers for their steadfast assistance, as well as to the DRACO Lab for access to the AMD Machine.

The University of Central Florida's Engineering Department labs (Senior Design Lab, Texas Instruments Innovation Lab, etc.) also greatly provided valuable resources for the development of GPMS. Additionally, the authors thank the committee reviewers Dr. Sahawneh, Dr. Atia, and Dr. Abdolvand for their contributions.

References

[1] C. Gregersen, "A Complete Guide to Development Boards," *Nabto*, Sep. 6, 2023. [Online]. Available: https://www.nabto.com/development-board-guide/

[2] "Raspberry Pi Documentation - Raspberry Pi Hardware," *Raspberry Pi*, n.d. [Online]. Available: https://www.raspberrypi.com/documentation/computers/raspberry-pi.html

[3] H. Ashtari, "Unix vs. Linux vs. Windows: How they compare," *Spiceworks Inc.*, Mar. 13, 2023. [Online]. Available: https://www.spiceworks.com/tech/tech-101/articles/unix-linux-windows-comparison/

[4] G. Bonuccelli, "Cloud vs server: Learn the key differences and benefits," *Server and Cloud Blog*, Feb. 15, 2022. [Online]. Available: https://www.parallels.com/blogs/ras/cloud-vs-server/

[5] W. Wassim, "Comparing desktop application development frameworks: Electron, Flutter, Tauri, React Native, and...," *Medium*, Sep. 13, 2023. [Online]. Available: https://medium.com/@maxel333/comparing-desktop-application-development-frameworks-electron-flutter-tauri-react-native-and-fd2712765377

[6] "What is HTTPS," *Cloudflare*, n.d. [Online]. Available: https://www.cloudflare.com/learning/ssl/what-is-https/

[7] "How HDMI Works," *HowStuffWorks*, Oct. 8, 2007. [Online]. Available: https://electronics.howstuffworks.com/hdmi.htm#:~:text=HDMI%20uses%20transition%20minimized%20differential%20signaling%20%28TMDS%29%20to

[8] "What is Transmission Control Protocol (TCP)?," *GeeksforGeeks*, n.d. [Online]. Available: https://www.geeksforgeeks.org/what-is-transmission-control-protocol-tcp/

[9] SSL.com, "Self-Signed Certificate Vulnerabilities," Dec. 6, 2023. [Online]. Available: https://www.ssl.com/article/ssl-tls-self-signed-certificates/.

[10] "Projector Specs Explained – Your Ultimate Projector Buyer's Guide!" *TheaterDIY*, Sep. 12, 2023. [Online]. Available: https://theaterdiy.com/projector-specs-explained-ultimate-projector-buyers-guide/

[11] "Camera Measurement Standards," *Imaging.org*, n.d. [Online]. Available: https://www.imaging.org/IST/IST/Standards/Camera_Measurement_Standards.aspx