3D MODEL OF CHENNAI CENTRAL

A PROJECT REPORT

Submitted by

Mithin Jain S, 20BAI1161
Puja Sai Thandavan, 20BAI1008
Akshara Ramprasad, 20BAI1070

in partial fulfillment for the award of the degree of

Bachelor of Technology
in
Computer Science and Engineering
with
Artificial Intelligence and Machine Learning

School of Computer Science and Engineering
Vellore Institute of Technology
Vandalur - Kelambakkam Road, Chennai - 600 127

November - 2022

CONTENTS

LIST OF FIGURES

1. INTRODUCTION

Architectural rendering, also known as architectural visualization, is the process of developing two-dimensional and three-dimensional visuals that show the attributes of a proposed architectural design. This technology is revolutionizing the way we design in ways that would have been unthinkable just 15 years ago. Architects and designers can use simple interactive 3D modeling to analyze "proportions" and "scales," as well as 3D rendering and architectural visualization tools to mimic the effects of lighting, ventilation, and acoustics in indoor environments. Architects, designers, and 3D artists can use modern 3D rendering technologies to change ideas in real-time, quickly moving from concept to concrete and exploring various choices, as well as making changes and manufacturing multiple copies of the designs.

In recent years, 3D graphics has become an increasingly important part of the multimedia web experience. Following on from the advent of the X3D standard and the definition of a declarative approach to presenting 3D graphics on the web, the rise of WebGL has allowed lower-level access to graphics hardware of ever-increasing power. In parallel, remote rendering techniques permit the streaming of high-quality 3D graphics onto a wide range of devices, and recent years have also seen much research on methods of content delivery for web-based 3D applications. All this development is reflected in the increasing number of application fields for the 3D web. Dr. M.C. Ramachandran Central Railway Station, commonly known as Chennai Central, is the main railway terminus in the city of Chennai, Tamil Nadu, India. It is the busiest railway station in South India and one of the most important hubs in the country. Chennai Central is a hub for suburban trains. We plan on making a 3d model of Chennai Central Railway station. We will be using Blender to create the model. Blender has a wide variety of tools making it suitable for almost any sort of media production. People and studios around the world use it for hobby projects, commercials, and feature films.

1.1 BACKGROUND

George Harding, an architect, used the 12th-century Romanesque style to create the Chennai Central Railway Station, which has been around for a century. It is undoubtedly one of Chennai's most well-known monuments and, along with the 2 km away Egmore Railway Station, serves as the principal rail entry point to the rest of the nation. By way of roads and suburban connectivity, these two stations are well connected. To the west of this Station is the Buckingham canal, which the British built to transport goods by waterway as far as Nellore in Andhra Pradesh. The primary hub of the suburban trains, which include MRTS and Metro, is located close to this main station. The multi-level shopping center and the underground metro station now being built will rank among Asia's biggest. To the east and west of this station, respectively, are the Southern Railway's corporate headquarters and the Ripon Building, a building constructed during the British Raj. The primary General hospital is to the south as well as Madras Medical College. To its credit, throughout its long existence, the station has undergone two renaming’s: first, in 1996, when it changed its name from Madras to Chennai to honor the late chief minister of Tamil Nadu, M. G. Ramachandran, it became Puratchi Thalaivar Dr. M.G. Ramachandran Central Railway Station. Perhaps the station in the Indian network with the longest name right now is this one. Together with Chennai Egmore, this terminus sees about 6 lakh passengers per day, making it the busiest rail hub in South India. Chennai Central holds the distinction of being one of the cleanest, receiving over 180 marks, even though Indian trains' cleanliness is never a strong suit 300 total points. It boasts 17 Platforms of which 12 is for long distance, & 5 for suburban commute, and has 30 tracks for interchanging. We are preserving Chennai Central by building a 3D model of it because it is a significant location within the city.

For this purpose, we are using the Blender software. With Blender, you can create 3D visualizations such as still images, 3D animations, and VFX shots. You can also edit videos. It is well suited to individuals and small studios who benefit from its unified pipeline and responsive development process. Being a cross-platform application, Blender runs on Linux, macOS, as well as Windows systems. It also has relatively small memory and drives requirements compared to other 3D creation suites. Its interface uses OpenGL to provide a consistent experience across all supported hardware and platforms. Its tools allow the user to model complex shapes using a variety of techniques. One can create landscapes through terrain displacement using digital elevation models, as well as populate the environment with any 3D objects. Complex geometries can be generated within the program using the numerous modeling tools available or imported from other software (imported as .obj, .fbx, .3ds, .stl). Each object can be given a material, which is controlled through a shader editor that allows great freedom on its physical properties and its interaction with the light. It is also possible to control the lighting of the scene using a variety of elements such as spotlights, sun-like sources, as well as emissive objects that allow for extremely realistic lighting conditions. All the elements introduced above allow the user to create photo-realistic renderings of the scene. The render options are controlled by the camera properties. The camera focal length, the resolution, and the sensor size can all be adjusted. The renderer takes into account the physics of both the materials and the camera to compute multiple light bounces coming from all the sources in the scene as they interact with the materials to obtain physically accurate results

1.2 STATEMENT

The station, which boasts stunning red architecture and a history spanning more than 140 years, was constructed during the British colonial era. Chennai Central Railway Station, is the busiest station in the city, with excellent connections to the suburbs of other Indian cities. Because Chennai Central is such an important landmark, making a 3D model of it would aid students in understanding 3D geometry and keep them interested in practical knowledge.

In this project, we'll be making a 3D Visualisation of Chennai Central. This can be used as an example to show how 3d and graphics designing can be used in the field of architecture and engineering. To collect pictures and create a 3D model of Chennai Central, we took a trip to MAS. We referred to these images to get to know about the colors, textures, and field of depths so that it will be easier to create a 3D model of the building.

1.3 MOTIVATION & CHALLENGES

MOTIVATION:

Chennai Railway station is listed under the heritage building sites at Chennai. It is one of the iconic buildings where all passengers enter and explore Chennai city by offering Indian railway services. The railway station is equipped with amenities like water machines, toilets, dormitory rooms, medical emergency help, and food stalls. However, no need to go out for food Rail-Recipe will deliver your favorite dish directly to your seats. We are creating a 3D Model of Chennai central to preserve this heritage. Good and timely documentation is one of the most effective means of facilitating the recovery and restoration of urban cultural heritage. If well documented, even destroyed individual buildings, ensembles of buildings and entire neighborhoods can be reconstructed a long time after they were destroyed. Recording pre-conflict conditions can also be crucial for maintaining the collective memory of a city or place as a living environment. Even in difficult or hostile political conditions, documentation can be used to rally and mobilize support for recovering lost heritage. 3D models are useful for documentation, visualization, and reconstruction purposes. They can be produced to represent remains and/or illustrate a hypothetical reconstruction. 3D models are a means of documenting historic monuments. They can be used as tools for scientific research, to preserve the memory of a site, or to give access – either digitally or in printed form – to a physical structure that cannot otherwise be seen.

CHALLENGES:

3D image rendering will take longer time than 2D image. The 3D rendering process can take quite a long time, depending on the complexity of the objects and scenes, because the software essentially takes photographs of every pixel contained in the image. Even on relatively fast computers, a complete rendering of a 3D scene may require hours (if not days) of processing time.

A common challenge many 3D modelers face with Blender is that to fully realize their artistic vision, they have to create highly complex shapes which are very time-consuming to render. While having a detailed and accurate model is critical for the final product, the added detail creates a lot of downstream problems such as slow rendering times and models being difficult to manipulate.

On some machines, Blender may run very slowly, or you may see weird screen glitches around the mouse pointer or menus. The first thing to check is the drivers for your video card. Go to the website of the manufacturer of your video card to see whether any updates are available.

Often when modeling, you run into a situation where a strange black crease goes along some edges. The stripe is usually most apparent when modeling with the Subdivision Surface modifier turned on and you're looking at your mesh in Solid viewport shading. What's happening here is that the normals for one of the faces adjoining this edge are pointing in the wrong direction.

Occasionally, you might run into a problem where not everything shows up in your 3D View, even though you're positive you didn't delete anything. The first thing to do is to make sure that nothing is hidden. Pressing H in the 3D View hides whatever you've selected, and it's easy to accidentally hit it when you're trying to press G and grab an object. Fortunately, you can unhide all hidden objects pretty quickly by pressing Alt+H.

1.4 LITERATURE-REVIEW

[1] https://www.sciencedirect.com/science/article/abs/pii/S0165027010000087

The nervous systems of animals evolved to exert dynamic control of behavior in response to the needs of the animal and changing signals from the environment. Understanding the mechanisms of dynamic control requires a means of predicting how individual neural and body elements will interact to produce the performance of the entire system. AnimatLab is a software tool that provides an approach to this problem through computer simulation. AnimatLab enables a computational model of an animal's body to be constructed from simple building blocks, situated in a virtual 3D world subject to the laws of physics, and controlled by the activity of a multicellular, multicompartment neural circuit. Sensor receptors on the body surface and inside the body respond to external and internal signals and then excite central neurons, while motor neurons activate Hill muscle models that span the joints and generate movement. AnimatLab provides a common neuromechanical simulation environment in which to construct and test models of any skeletal animal, vertebrate or invertebrate. The use of AnimatLab is demonstrated in a neuromechanical simulation of human arm flexion and the myotatic and contact-withdrawal reflexes.

[2] https://asmedigitalcollection.asme.org/IDETC-CIE/proceedings-abstract/DETC99/1029/1097430

With the advancement of computer techniques, a greater emphasis has been placed on intuitive human-computer interactions (HCIs). Virtual Reality systems can offer a novel way for users to interact with the objects in the computer-generated environment (the so-called Virtual Environment, VE). Through VR technology, we can replace the traditional input device, such as a keyboard and mouse, with other modes such as speech and gesture. In our research project, we use CyberGlove, developed by Virtual Technology Inc., as an input device to develop a desktop CAD modeling system for conceptual designers. We elaborate on the limitations of Dataglove and use gestures to support an intuitive human-computer interface. To develop this gesture interface, we emphasize that conceptual designers are allowed full freedom to use different kinds of gestures to conduct various geometric shape operations instead of depending solely on the keyboard and 2D mouse. The designers can indicate objects or directions simply by pointing with the hand and manipulate the position and orientation of an object by grasping and turning. The "virtual tools" can be used for shaping, cutting, and joining objects. We employ 3D GUIs for enhancing the gesture interface. In the VE, the 3D menu and "virtual hand" float over the objects rather than being part of the scene. Various 3D cursors can be used to select a menu or manipulate the object.

[3] https://dl.acm.org/doi/abs/10.1145/1015706.1015782

Valuable 3D graphical models, such as high-resolution digital scans of cultural heritage objects, may require protection to prevent piracy or misuse, while still allowing for interactive display and manipulation by a widespread audience. We have investigated techniques for protecting 3D graphics content, and we have developed a remote rendering system suitable for sharing archives of 3D models while protecting the 3D geometry from unauthorized extraction. The system consists of a 3D viewer client that includes low-resolution versions of the 3D models, and a rendering server that renders and returns images of high-resolution models according to client requests. The server implements several defenses to guard against 3D reconstruction attacks, such as monitoring and limiting request streams and slightly perturbing and distorting the rendered images. We consider several possible types of reconstruction attacks on such a rendering server, and we examine how these attacks can be defended against without excessively compromising the interactive experience for non-malicious users.

[4] https://dl.acm.org/doi/abs/10.1145/1186822.1073277

We present an efficient approach for end-to-end out-of-core construction and interactive inspection of very large arbitrary surface models. The method tightly integrates visibility culling and out-of-core data management with a level-of-detail framework. At preprocessing time, we generate a coarse volume hierarchy by binary space partitioning the input triangle soup. Leaf nodes partition the original data into chunks of a fixed maximum number of triangles, while inner nodes are discretized into a fixed number of cubical voxels. Each voxel contains a compact direction-dependent approximation of the appearance of the associated volumetric subpart of the model when viewed from a distance. The approximation is constructed by a visibility-aware algorithm that fits parametric shaders to samples obtained by casting rays against the full-resolution dataset. At rendering time, the volumetric structure, maintained off-core, is refined and rendered in front-to-back order, exploiting vertex programs for GPU evaluation of view-dependent voxel representations, hardware occlusion queries for culling occluded subtrees, and asynchronous I/O for detecting and avoiding data access latencies. Since the granularity of the multiresolution structure is coarse, data management, traversal, and occlusion culling cost are amortized over many graphics primitives. The efficiency and generality of the approach are demonstrated with the interactive rendering of extremely complex heterogeneous surface models on current commodity graphics platforms.

[5] https://dl.acm.org/doi/abs/10.1145/588272.588279

As the number of 3D models available on the Web grows, there is an increasing need for a search engine to help people find them. Unfortunately, traditional text-based search techniques are not always effective for 3D data. In this article, we investigate new shape-based search methods. The key challenges are to develop query methods simple enough for novice users and matching algorithms robust enough to work for arbitrary polygonal models. We present a Web-based search engine system that supports queries based on 3D sketches, 2D sketches, 3D models, and/or text keywords. For the shape-based queries, we have developed a new matching algorithm that uses spherical harmonics to compute discriminating similarity measures without requiring the repair of model degeneracies or alignment of orientations. It provides 46 to 245% better performance than related shape-matching methods during precision-recall experiments, and it is fast enough to return query results from a repository of 20,000 models in under a second. The net result is a growing interactive index of 3D models available on the Web (i.e., a Google for 3D models).

[6] https://ieeexplore.ieee.org/abstract/document/923386

Measuring the similarity between 3D shapes is a fundamental problem, with applications in computer vision, molecular biology, computer graphics, and a variety of other fields. A challenging aspect of this problem is to find a suitable shape signature that can be constructed and compared quickly, while still discriminating between similar and dissimilar shapes. In this paper, we propose and analyze a method for computing shape signatures for arbitrary (possibly degenerate) 3D polygonal models. The key idea is to represent the signature of an object as a shape distribution sampled from a shape function measuring the global geometric properties of an object. The primary motivation for this approach is to reduce the shape-matching problem to the comparison of probability distributions, which is simpler than traditional shape-matching methods that require pose registration, feature correspondence, or model fitting. We find that the dissimilarities between sampled distributions of simple shape functions (e.g. the distance between two random points on a surface) provide a robust method for discriminating between classes of objects (e.g. cars versus airplanes) in a moderately sized database, despite the presence of arbitrary translations, rotations, scales, reflections, tessellations, simplifications and model degeneracies. They can be evaluated quickly, and thus the proposed method could be applied as a pre-classifier in an object recognition system or an interactive content-based retrieval application.

[7]https://www.researchgate.net/profile/Philip-Shilane/publication/200018879 Stratified_Point_Sampling_of_3D_Models/links/555b7b5b08aec5ac22323eab/Stratified-Point-Sampling-of-3D-Models.pdf

In this article, we present a stratified sampling strategy for 3D models. Stratified sampling is a technique that generates evenly spaced samples by subdividing the sampling domain into non-overlapping parts and sampling independently from each part. It has been shown to decrease the variance of the numerical estimation of integrals in several applications, including the antialiasing of ray-traced images [Mit96]. Our technique behaves like voxelization but generates samples on the surface of the model. The idea is to voxelized the model and output one sample for each voxel, choosing a position from the part of the model surface that is contained in the voxel's bounding box. The sample is selected according to a probability that decays as its distance to the center of the voxel increases. We allow the user to control the sampling resolution, the regularity of the sampling, and the minimum distance between samples. Most previous sampling strategies consider only sample density over the surface area of the original model. Among them, uniform sampling is by far the most common strategy. Samples are spread such that the probability of a surface point being sampled is equal for all surface points. Uniform sampling is popular because it is simple, efficient, and unbiased. However, artifacts such as those seen in random dithered images also appear in uniformly sampled models, and for many applications, these artifacts are aesthetically undesirable. Other applications, such as point rendering systems, demand bounded maximum or minimum distance between samples. For these applications, uniform sampling is not an option.

[8] https://dl.acm.org/doi/abs/10.1145/3173574.3173772

Tactile maps are widely used in Orientation and Mobility (O&M) training for people with blindness and severe vision impairment. Commodity 3D printers now offer an alternative way to present accessible graphics, however, it is unclear if 3D models offer advantages over tactile equivalents for 2D graphics such as maps. In a controlled study with 16 touch readers, we found that 3D models were preferred, enabled the use of more easily understood icons, facilitated better short-term recall, and allowed the relative height of map elements to be more easily understood. Analysis of hand movements revealed the use of novel strategies for systematic scanning of the 3D model and gaining an overview of the map. Finally, we explored how 3D-printed maps can be augmented with interactive audio labels, replacing less practical braille labels. Our findings suggest that 3D-printed maps do indeed offer advantages for O&M training.

[9] https://ieeexplore.ieee.org/abstract/document/232096

Model-based encoding of human facial features for narrowband visual communication is described. Based on an already prepared 3D human model, this coding method detects and understands a person's body motion and facial expressions. It expresses the essential information as compact codes and transmits it. At the receiving end, this code becomes the basis for modifying the 3D model of the person and thereby generating lifelike human images. The feature extraction used by the system to acquire data for regions or edges that express the eyes, nose, mouth, and outlines of the face and hair is discussed. How the system creates a 3D model of the person by using the features extracted in the first part to modify a generic head model is also discussed.

[10] https://www.sciencedirect.com/science/article/abs/pii/S092359651000113X

In this paper, we propose a view-based 3D model retrieval algorithm, where the many-to-many matching method, weighted bipartite graph matching, is employed for comparison between two 3D models. In this work, each 3D model is represented by a set of 2D views. Representative views are first selected from the query model and the corresponding initial weights are provided. These initial weights are further updated based on the relationship among these representative views. The weighted bipartite graph is built with these selected 2D views, and the matching result is used to measure the similarity between two 3D models. Experimental results and comparisons with existing methods show the effectiveness of the proposed algorithm.
Recently, large databases of 3D models are rapidly increasing, and 3D models have been widely used in CAD, virtual reality, medicine, and entertainment. Effective and efficient 3D model retrieval algorithms are required in wide applications. Early 3D model retrieval

methods employed low-level features, and high-level structure-based methods to describe 3D models. Recently, view-based 3D model descriptors came out. These view-based 3D model descriptors represent 3D models using 2D views, and 3D model comparison is based on 2D view matching.

The state-of-the-art view-based 3D object retrieval methods are highly dependent on the methods of view acquisition. The light field descriptor (LFD) was computed from 10 silhouettes obtained from the vertices of a dodecahedron over a hemisphere. This image set described the spatial structure information from different views. In LFD, Zernike moments and Fourier descriptors of the 3D model were employed as the features of each image. This method found the best match between two LFDs as the similarity between two 3D models. The elevation descriptor (ED) was a global spatial information descriptor. ED represented 3D models by the spatial information from six directions. It was invariant to the translation, rotation, and scaling of 3D models. The comparison between the two EDs was based on the distance between two groups of six elevation views. Five circular camera arrays, including four vertical and one horizontal camera arrays, are employed to acquire representative views of 3D models. Each group of views (acquired by a circle set of cameras) was modeled as a Markov chain (MC). In MC, 3D model comparison included two stages: comparison in the view set level and comparison in the model level. In the MC framework, 3D model retrieval was to find the maximal posterior (MAP) in the 3D database given the query model.

2. PLANNING & REQUIREMENTS SPECIFICATION

2.2 SYSTEM REQUIREMENTS

Although Blender is free to download, It has many features that can’t be used by everyone. Like most software, Blender has minimum standard requirements for the hardware and software (mostly hardware) on your computer to operate effectively.

2.2.1 HARDWARE REQUIREMENTS

MINIMUM:

- 64-bit quad-core CPU with SSE2 support
- 8 GB RAM
- Full HD display
- Mouse, trackpad, or pen + tablet
- Graphics card with 2 GB RAM, OpenGL 4.3
- Less than 10 years old

RECOMMENDED:

- 64-bit eight-core CPU

- 32 GB RAM
- 2560×1440 display
- Three-button mouse or pen + tablet
- Graphics card with 8 GB RAM

A full HD display or higher is recommended. Multi-monitor setups are supported, and workspaces can be configured to span multiple monitors.

BLENDER SUPPORTS VARIOUS TYPES OF INPUT DEVICES:

- Keyboard (recommended: keyboard with numeric keypad, English layout works best)
- Mouse (recommended: three-button mouse with scroll wheel)
- NDOF Device (also known as 3D Mouse)
- Graphic Tablet

2.2.2 SOFTWARE REQUIREMENTS

Latest Version of Blender
OS that supports Blender

The supported platforms are detailed below:

- Windows 2000, XP, Vista,
- Mac OS X (PPC and Intel)
- Linux (i386)
- Linux (PPC)
- FreeBSD 5.4 (i386)
- SGI Irix 6.5
- Sun Solaris 2.8 (sparc)

3. SYSTEM DESIGN

The 3D model of Chennai Central was built using the software – Blender.

Blender is a powerful open-source tool to be used for 3D graphics. The current version is 3.3.1 and the application's splash screen is presented below.

Blender is a free and open-source 3D creation suite. It supports the entirety of the 3D pipeline—modeling, rigging, animation, simulation, rendering, compositing and motion tracking, and even video editing and game creation. Advanced users employ Blender's API for Python scripting to customize the application and write specialized tools; often these are included in Blender's future releases. Blender is well suited to individuals and small studios who benefit from its unified pipeline and responsive development process. Blender is cross-platform and runs equally well on Linux, Windows, and Macintosh computers. Its interface uses OpenGL to provide a consistent experience. Blender has a wide variety of tools making it suitable for almost any sort of media production. People and studios around the world use it for hobby projects, commercials, and feature films. Blender has a wide variety of tools making it suitable for almost any sort of media production. People and studios around the world use it for hobby projects, commercials, and feature films.

Blender is chock full of useful tools, but some will be more relevant to beginners than others. For many coming to Blender, the most popular tools are modeling, sculpting, and texturing, as well as animation. Those creating objects for 3D printing may not even go beyond modeling and sculpting. However, for those who are interested in exploring the more advanced 3D techniques and tools, you'll want to check out the 2D/3D hybrid Grease Pencil, physics simulations, scripting, and visual effects.

The main Blender features are related to the following areas:

- Animation
- Rigging
- UV unwrapping
- Physics and particles
- Modeling
- Rendering
- Shading
- Imaging and compositing
- Realtime 3D/Game Creation

Beyond the primary tools, there's a lot to discover in Blender!

- The Grease Pencil is a fascinating and popular tool that allows you to paint in 3D space using 2D brushes. You can create 2D animations using a hybrid workspace.

- Physics simulations such as gravity, cloth, and generated hair are fun to use. Other cool ones to play with are fire, smoke, or liquid particle effects.
- Blender’s powerful rendering engine lets you output your images and designs in a variety of formats and resolutions.
- There are lots of video editing and visual effects that can turn your renders into fully-qualified animations.
- Blender’s built-in scripting lets you shape the program to your needs.

4. IMPLEMENTATION:

- The model of Chennai Central was made with the help of a blender using pictures of Chennai Central as a reference.

Figure 1 – Front View

As shown in Figure 1, the front view of Chennai Central was thus made using various small ingredients such as the clock on the top of the tower, the U-shaped windows/exits, the trees, the bushes, etc.

Figure 2 – An attachment of another block of our Chennai Central

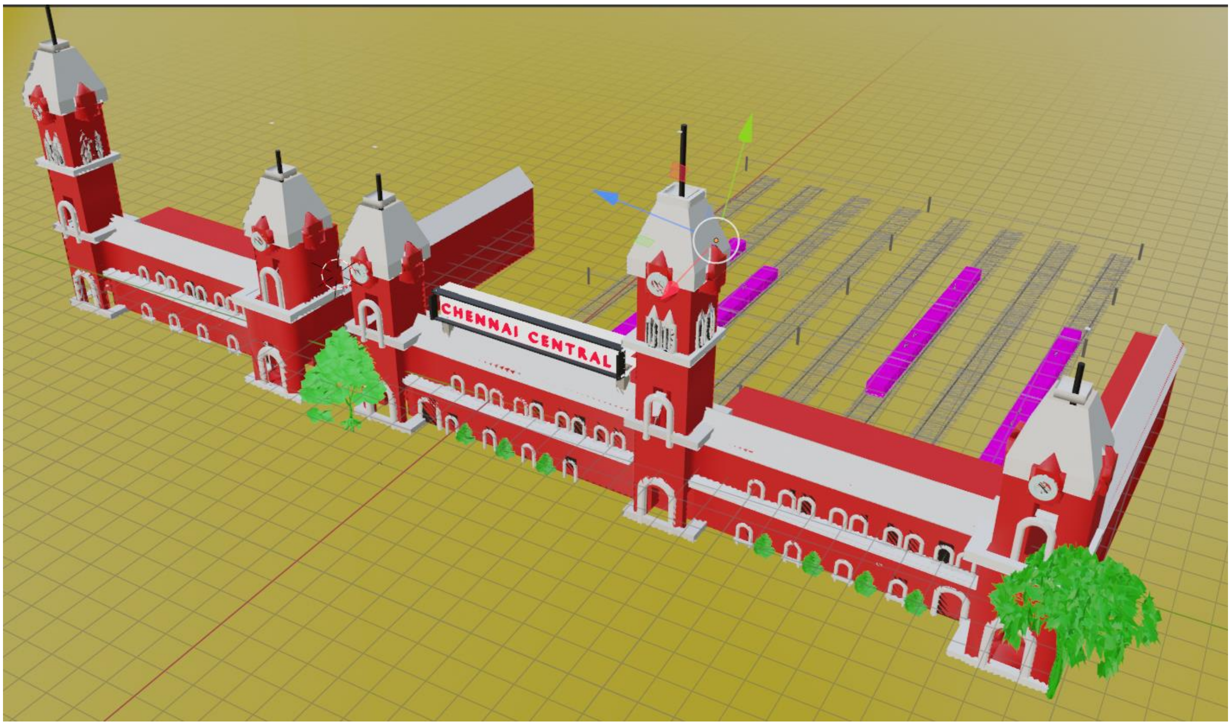


Figure 3 – Side view of our entire model

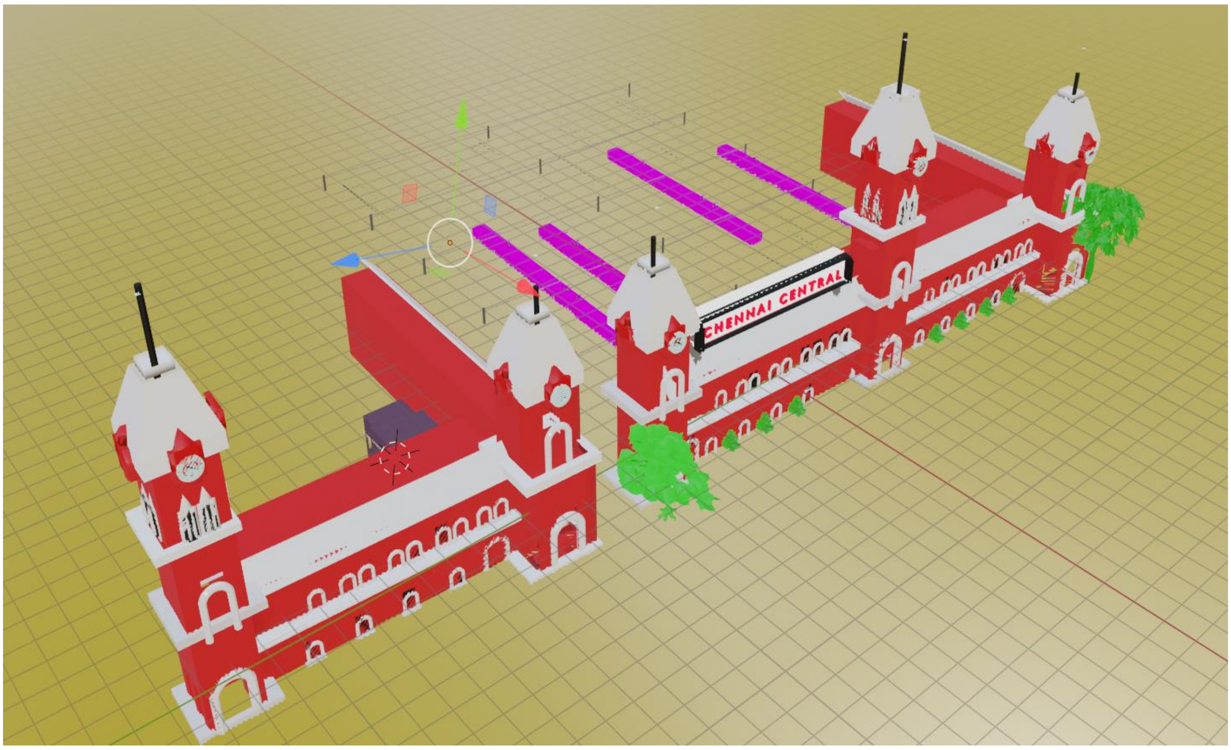


Figure 4 – Viewing our entire model from another viewpoint

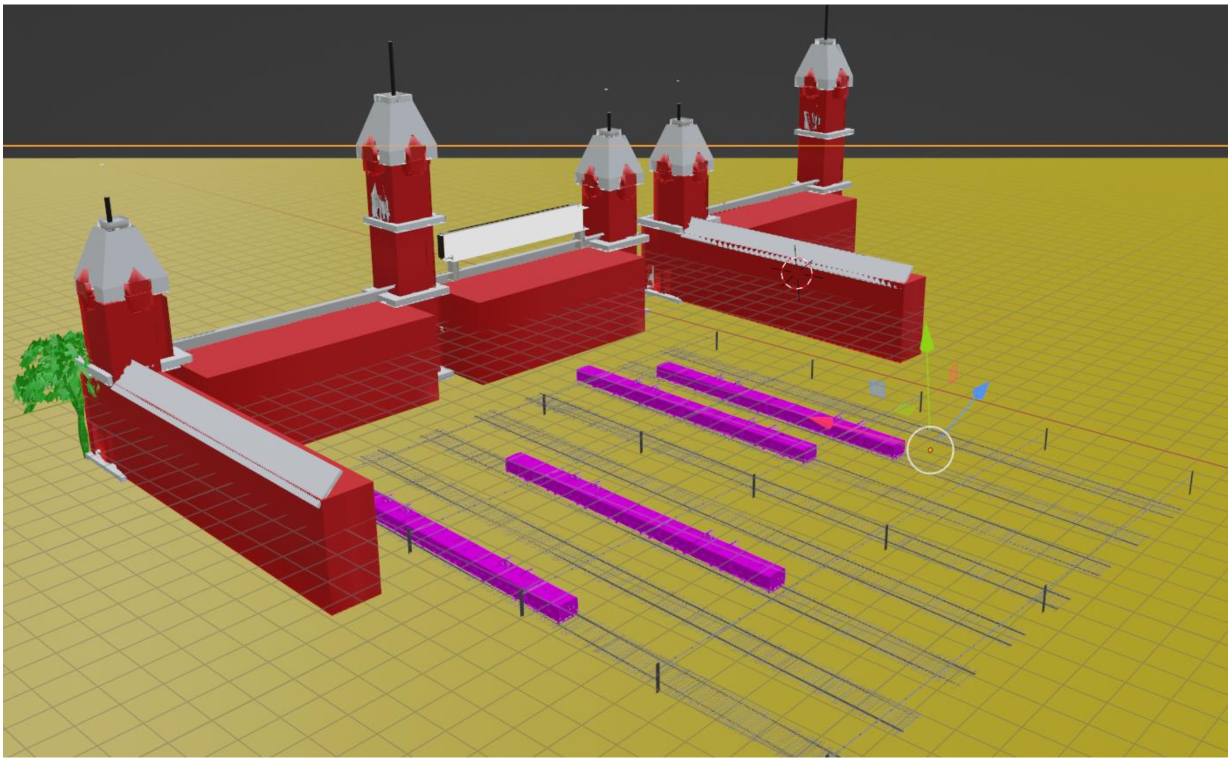


Figure 5 – The view of the back of our model

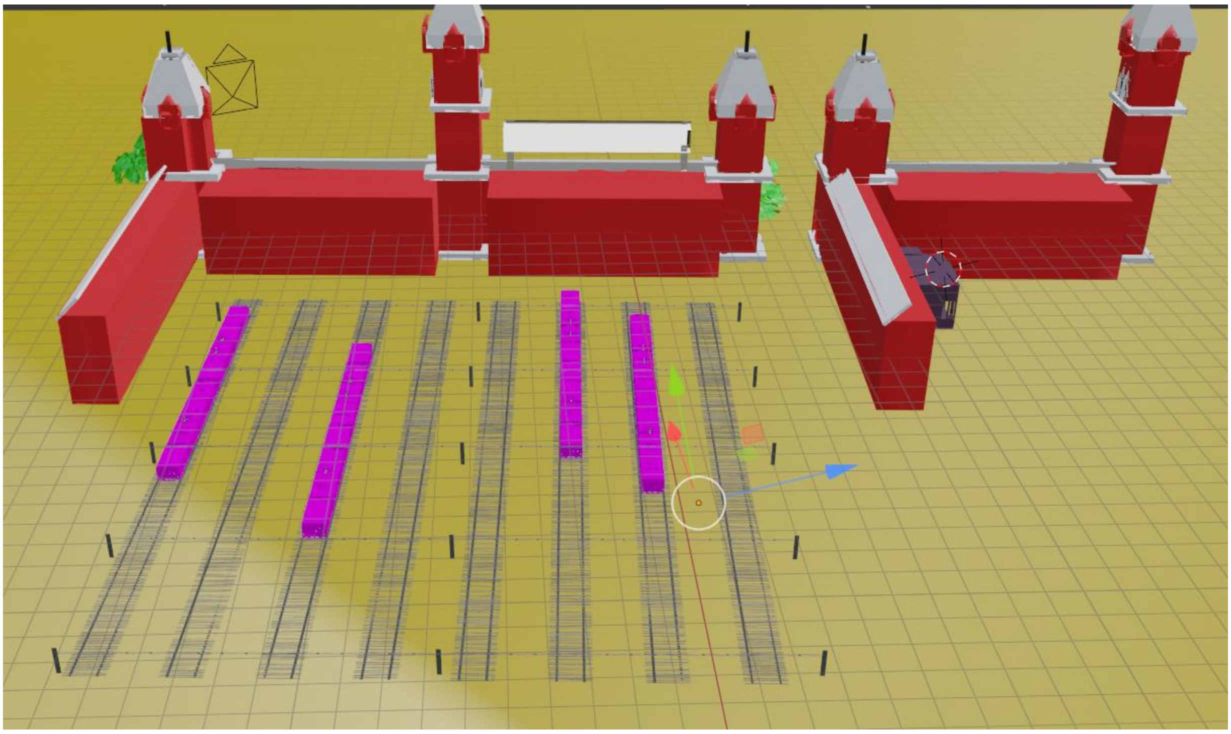


Figure 6 – Complete back view of our model

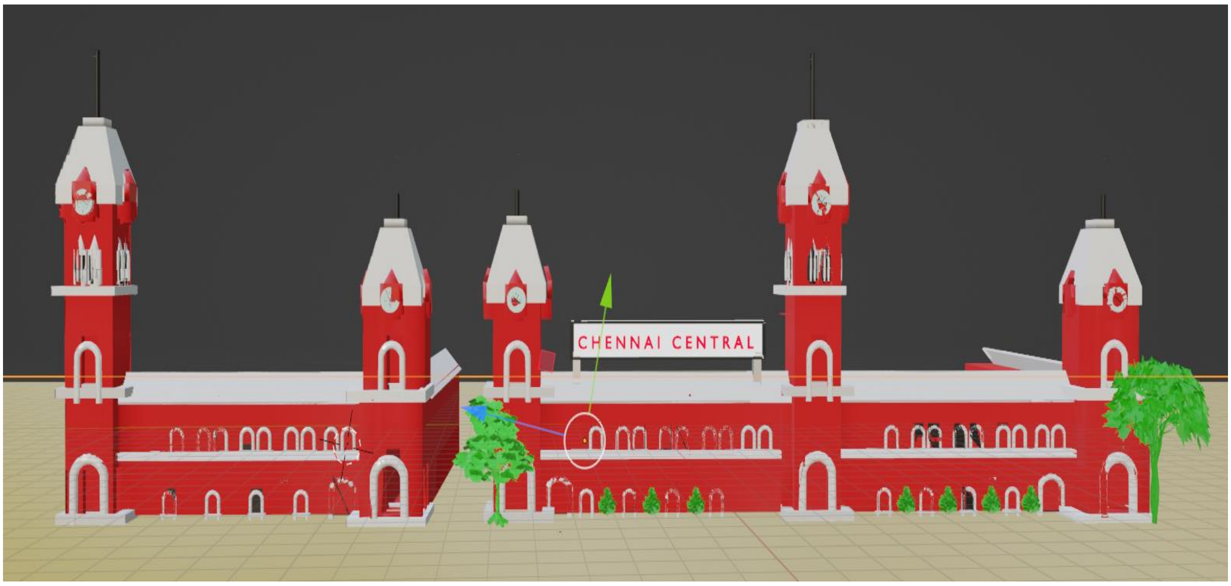


Figure 7 – Complete front view of our model

In the last couple of figures, we notice the inclusion of tracks, electric poles, trains, and also a shop to make our model look more realistic to the actual Chennai Central out there. The trains have been perfectly aligned with the tracks and the electric poles that power them to run on the tracks.

More specific details regarding the shop and the tracks and trains have been provided below,

Figure 8 – The Tea Shop at the central

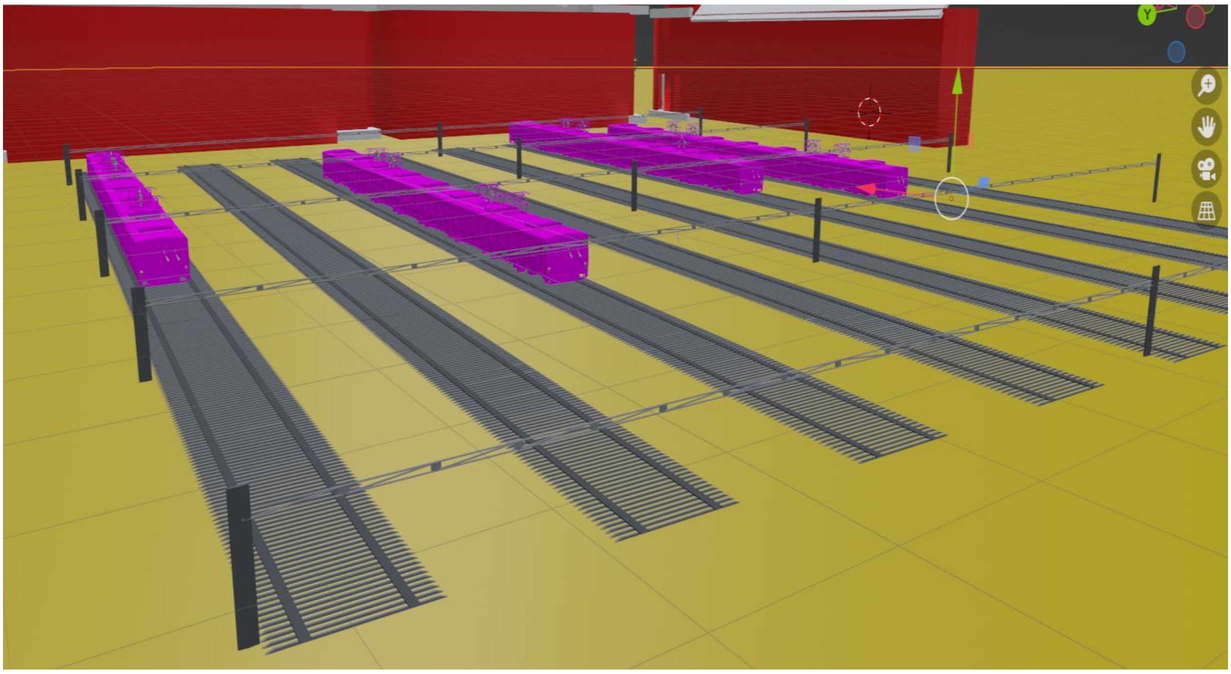


Figure 9 – A view of the tracks and the trains from behind

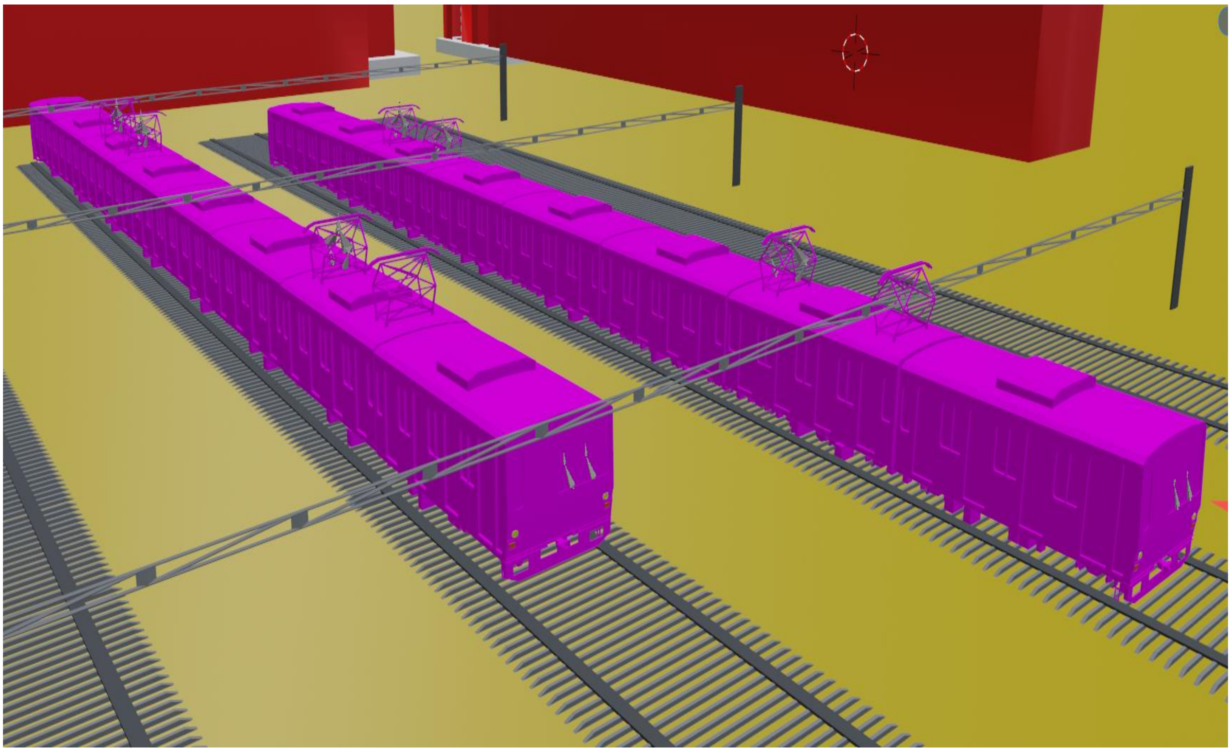


Figure 10 – A more detailed picture depicting the trains and the tracks along with the electric poles

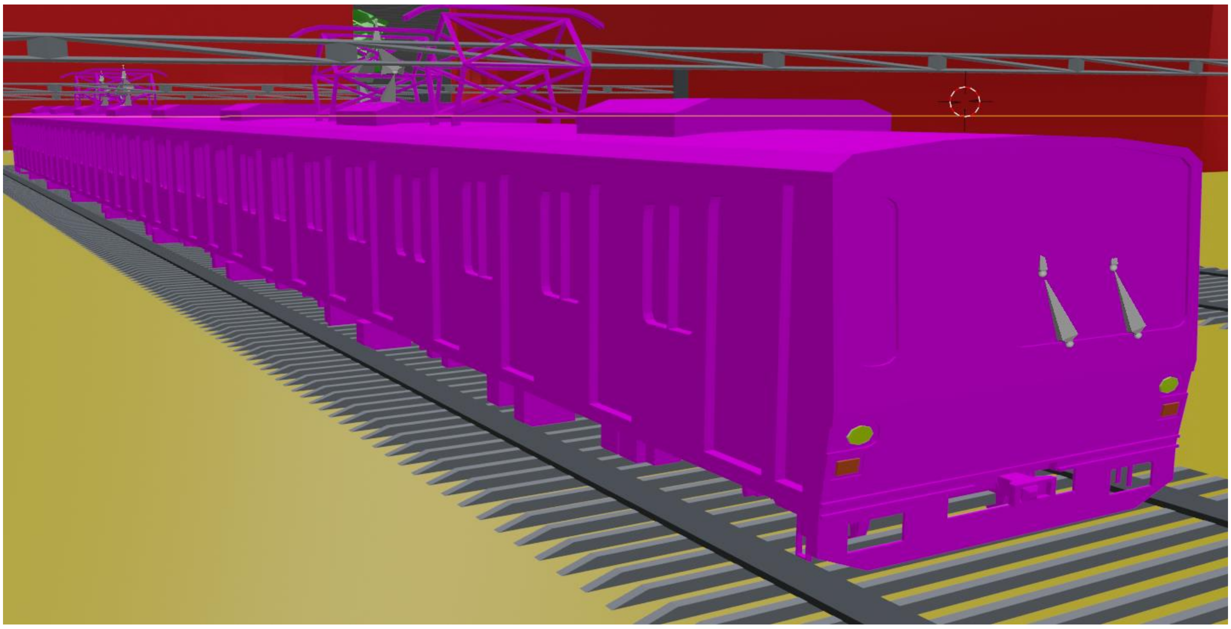
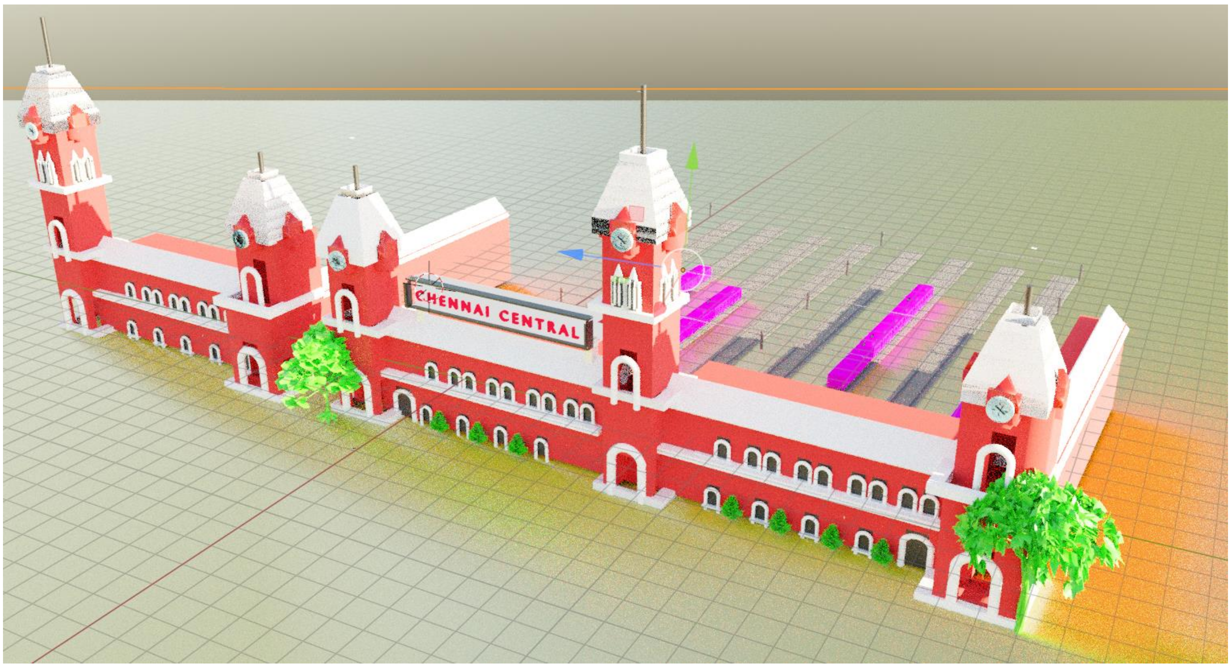
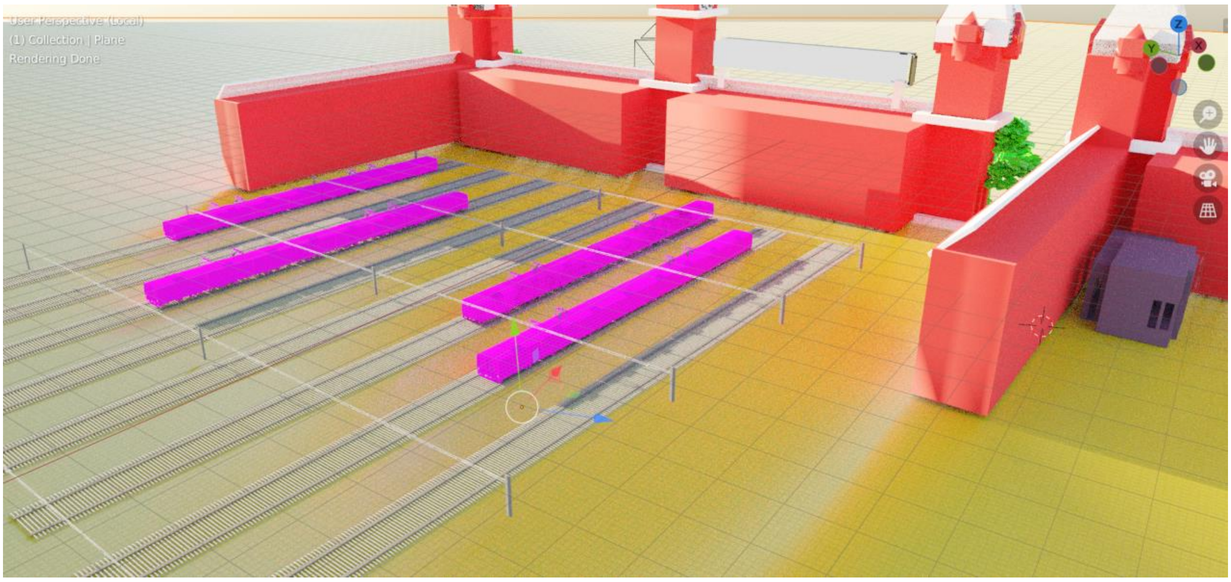


Figure 11 – A close-up view of the train itself





Figures 12, 13, and 14 give us a glimpse of our model when rendered completely

5. RESULTS AND DISCUSSION

Our model's completion allowed us to emphasize how crucial 3D models are. After building our model, we were able to attract a lot of interest from friends and fellow students. They were able to appreciate Chennai Central's beauty as a result. Moreover, The value of using 3D models to protect important landmarks and cultural artifacts should be stressed. Since Blender is among the most accessible and user-friendly pieces of free software available, we decided to use it as our method. Other techniques might also yield comparable results, but we chose Blender because we found it to be the most practical. It is open-source software that is free. It's accessible and simple for anyone to start using, as was previously mentioned.

6. CONCLUSION AND FUTURE WORK

CONCLUSION:

In this paper, the implementation of computer parametric design to facilitate the design of the 3D model of Chennai Central is successfully realized using modern technological tools such as the free software Blender. We were able to preserve this historic monument using the 3D model.

The user can develop an environment within the Blender GUI using all the available tools for modeling, shading, and illumination. The environment can then be used to render images of the scene at run time. The ease of implementation together with the physically accurate renderer allows for online realistic modeling of complex dynamic environments. Through the convenient interface of the program, the modeling of the desired three-dimensional shapes is greatly facilitated. This also applies to the automated process of constructing a drawing of the resulting polygonal 3D models, which in the future are assembled into three-dimensional paper models. This paper describes in detail the process of fine-designing a 3D model of Chennai Central Railway Station using specific exemplary primitive, text, and with a relatively complex form

FUTURE WORK:

Future developments will include the full rail system and a thorough overview of the paths and passengers by utilizing cutting-edge animation elements. Additionally, we'll put a Chennai Central absolute animation system into place and simulate train action in 3D.

7. REFERENCES

Markova, K., Dovramadjiev, T., & Jecheva, G. (2017, December 28). Computer parametric designing in Blender software for creating 3D paper models. ANNUAL JOURNAL OF TECHNICAL UNIVERSITY OF VARNA, BULGARIA, 1(1), 77-84. https://doi.org/10.29114/ajtuv.vol1.iss1.44

Alecu, F. (2010). Blender institute–the institute for open 3D projects. Open Source Science Journal, 2(1), 36-45.

Cofer, D., Cymbalyuk, G., Reid, J., Zhu, Y., Heitler, W. J., & Edwards, D. H. (2010). AnimatLab: a 3D graphics environment for neuromechanical simulations. *Journal of neuroscience methods*, *187*(2), 280-288.

Zheng, J. M., Chan, K. W., & Gibson, I. (1999, September). A VR-based 3D graphics user interface for CAD modeling system. In *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference* (Vol. 19715, pp. 1029-1036). American Society of Mechanical Engineers.

Koller, D., Turitzin, M., Levoy, M., Tarini, M., Croccia, G., Cignoni, P., & Scopigno, R. (2004). Protected interactive 3D graphics via remote rendering. *ACM Transactions on Graphics (TOG)*, *23*(3), 695-703.

Gobbetti, E., & Marton, F. (2005). Far voxels: a multiresolution framework for interactive rendering of huge complex 3d models on commodity graphics platforms. In *ACM SIGGRAPH 2005 Papers* (pp. 878-885).

Funkhouser, T., Min, P., Kazhdan, M., Chen, J., Halderman, A., Dobkin, D., & Jacobs, D. (2003). A search engine for 3D models. *ACM Transactions on Graphics (TOG)*, *22*(1), 83-105.