2D to 3D dynamic modeling of architectural plans in Augmented Reality

Priyanka Shinde[1], Abhijit Turate[2], Kesha Mehta[3], Renuka Nagpure[4]

[1,2,3,] Student, Dept. of Information Technology, Atharva College of Engineering, Maharashtra, India
[4] Assistant Professor, Dept. of Information Technology, Atharva College of Engineering, Maharashtra, India

---***---

Abstract- *The research paper develops an augmented reality application "AR-CHI-TECH" which proposes to develop a dynamic 3D model from a 2D floorplan drawing. It takes input from a 2D floorplan drawn by an architect and dynamically generates a corresponding 3D model of the same. Later this model is displayed in Augmented Reality for a better visual understanding. This is achieved using a 3D game engine Unity3D and an Augmented Reality (AR) engine Vuforia. The experimental results confirm that the system obtains accurate dynamic 3D models using reliable 2D floor plans.*

Keywords: Augmented Reality; floorplan; dynamic 3D modeling; 2D; image processing; AR;

I. Introduction

Augmented reality uses the existing real-world environment and puts virtual information on top of it to enhance the experience. It is often observed that ordinary, non-technical people find it hard to follow the complicated 2D floorplan drawings by an architect. As these drawings include a lot of technical jargon and symbols, the knowledge of which is unfamiliar to ordinary people. These 2D images of a 3D space (like a house floorplan) do not give a clear understanding of how the finished product will look.

ARchiTech helps to solve all the above-mentioned problems under a single roof. It takes input as an image of a floorplan. That dynamically creates a 3D model of the same at runtime. Lastly, displays the model in AR.

This helps the user to see the floorplan thoroughly which helps them to visualize how the house/floorplan might look before its completion.

Mobile devices nowadays have become pretty powerful and have the impressive processing power. This fact can be leveraged to deploy Augmented Reality applications on modern mobile devices. Also, the increased network speeds help in reducing the delay between the client and server communication required for the app.

II. Proposed System

The floorplan that architect designs are hard to distinguish by ordinary people. Also, 2D images of a 3D space do not give a clear idea of how the final product will look.

AR is basically used to digitally place an object on the surface of a real-life existing body. AR lets us interact with virtual stuff like it is present in front of us. The process of augmented reality is preliminary.

The device supporting the AR feature uses a lens to scan the object of which the digital impression is to be created. The algorithm and the software are then used to measure the distance of the object. AR is, in fact, an enhancement version of the real physical world.

III. Literature Review

After doing thorough research we have noticed that there are a few systems that exist under the same domain. But these models have drawbacks and are not very optimized. Several such products have some errors like image processing, and inefficient image cleaning while converting the 2D floorplan data to a 3D model.

This project mainly attacks these drawbacks and works on building the model sustainable for a long time and also very efficient. Thus, we put forward a project for dynamically converting the floorplan to 3D. With the optimization, we also offer the consumers an easy-to-use UI as well as high portability, alongside the mobile app.

IV. Methodology

Execution of the project is done in 3 phases, specifically

- Image Processing,
- 2D to 3D conversion
- Displaying in AR.
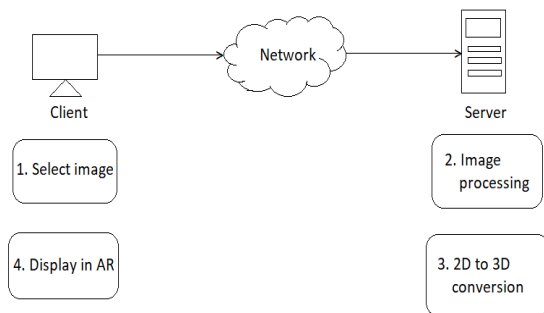
This is how the project architecture will look:

Figure 1: Project Architecture.

Initially, the image must be selected from the mobile device that is to be sent to the server for processing. This is achieved with the help of a plugin called AndroidNativePlugin [9]. This plugin uses core Java methods to provide basic Android functionalities like toast messages, gallery access, etc.

A. Image Processing.

Once an image has been captured as shown in figure 2 it is then sent to the server, which is then passed through a series of OpenCV-python functions to extract information. Firstly, a dilate operation is applied to erase finer details like text. This dilate operation can be performed multiple times to extract any unwanted details. The components exceeding a minimum threshold value in the size of pixels are stored.

Figure 2: Before Processing

Image Processing technology and OpenCV 4.5.4 library [7] in Python version 3.9[4] are used to detect walls and clean noise for optimal visualization. As the image is being processed as a grayscale the dilated floor plan with minimum distortion is inverted (negated) with the help of the ImageOps Module of Pillow Library [8] available for python. Later, another round of denoising is carried out resulting in a noise-free, well-processed floor plan.
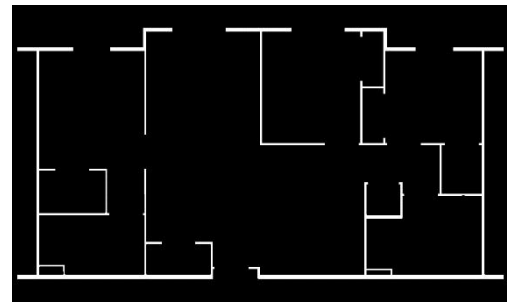

Figure 3: After Processing

B. 2D to 3D conversion.

The pre-processed image from the first phase is used as an input here. The NumPy-STL   Library [6] is used to convert the input image to a 3D STL Model. This is a simple library used to make working with STL files (and 3D objects in general) fast and simple. Due to all operations heavily relying on NumPy's this is one of the fastest STL editing libraries for Python available.
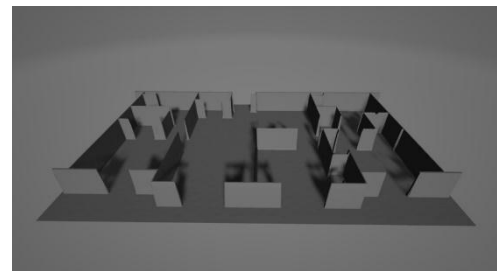

Figure 4: After converting into a 3D model

C. Augmented Reality (AR) display.

The final phase of implementation is the one where the client receives the 3D model from the server. The model is then displayed in Augmented Reality to obtain a better idea of what the actual, finished product would look like.

To achieve this, an Augmented Reality (AR) engine is required. The AR engine we used in this project is named Vuforia [5]. Vuforia uses computer vision technology to recognize and track planar images and 3D objects in real-time and requires two parameters to complete this phase which is a 3D model which is provided by the server and an "image target", upon which it overlays the 3D model.

Vuforia runs multiple scripts in the background that work in conjunction and whenever the target image is detected, it overlays the 3D model upon it and enables the user to experience how the finished construction will look.
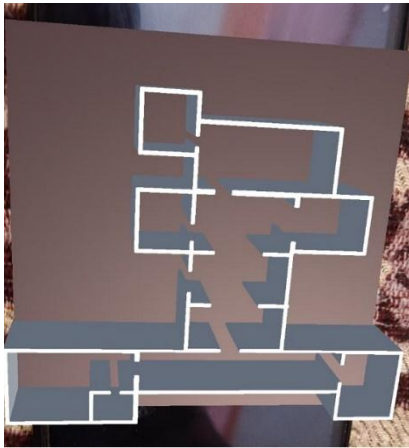
Figure 5: Displaying 3D model in AR

V. Result and Analysis

The end result of this project is a 3D model, which is obtained from a drawing of a 2D architectural floorplan, after undergoing image processing and dynamic 3D modeling. The main thought behind this project was to create an app that provides multiple functionalities like image processing, dynamic 3D modeling, and displaying in AR all under one roof.

This app uses a server running Python scripts, so as to relieve the client's device from all the heavy processing and at the same time acquire the benefits of Python's extensive image processing libraries. The Vuforia SDK uses a tracker-based method which also brings a little delay in the AR display phase.

VI. Conclusion

In this paper, we developed a tool to extract values from a 2D image of a blueprint and created a 3D model in real-time. The major outcome of this research was to create 3D models dynamically and offer customers the ability to customize their prefabricated home interiors while keeping the design of the exterior structure. Augmented Reality offers a real visualization of the final product for customers, enabling clients to create their dream homes.

REFERENCES

[1] **Developing augmented reality-based application (Accessed November 2021); Link:** https://www.researchgate.net/publication/337466471

[2] **Pillow Library (Accessed November 2021); Link:** https://pillow.readthedocs.io/en/stable/

[3] **Dynamic 3D model construction using architectural house plans (Accessed January 2019); Link:** https://ieeexplore.ieee.org/document/7872850

[4] **Vuforia Augmented Reality SDK (Accessed September 2021); Link:** - https://en.wikipedia.org/wiki/Vuforia_Augmented_Reality_SDK

[5] **Numpy-stl Library (Accessed September 2021) Link:** https://numpy-stl.readthedocs.io/en/latest/

[6] **OpenCV Library (Accessed September 2021); Link:** https://en.wikipedia.org/wiki/OpenCV

[7] **Stack Overflow (Accessed August 2021); Link:-** https://stackoverflow.com/questions/54274610/separate-rooms-in-a-floor-plan-using-opencv

[8] **Python (Accessed 2020/21); Link:** https://www.python.org/

[9] **Parsing floor plan images** (May 2017). DOI:10.23919/MVA.2017.7986875. Samuel Dodge University Jiu Xu, Bjorn Stenger. Conference: 2017 Fifteenth IAPR International Conference on Machine Vision Applications (MVA).

[10] **3- Phase Recognition Approach to Pseudo 3D building generation from 2D floorplan** (June 2011). Raj Kishen Moloo, Muhammad Ajmal Sheik Dawood, Abu Salmaan Auleear. International Journal of Computer Graphics & Animation (IJCGA).

[11] **Generating 3D Models from 2D Floor Plan Images Using Ensemble Methods** (December 2020). Sungsoo Park 1,2 and Hyeoncheol Kim, Department of Computer Science and Engineering, Korea University, Seoul, Korea (MDPI).

[12] **Floorplanning for 3-D VLSI Design** (July 2005), Lei Cheng, Liang Deng, and M. D. F. Wong. Asia and South Pacific Design Automation Conference, 2005 (IEEE).