IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 2, May 2026
Impact Factor: 8.2

DIMENDRAFT: Assist Beyond Sight

Gagan S
Junior Software Engineer, Nirmith AI Labs, Bengaluru

Abstract: *DIMENDRAFT is an innovative software application designed to simplify the conversion of 2D floor plans into 3D models, a process traditionally complex and time-consuming for architects, designers, and engineers. Utilizing a blend of Python, OpenCV, Blender API, PyQt5, Matplotlib, and Django, the application offers a user-friendly solution that significantly reduces the effort and time required for such conversions. The core functionality includes creating 2D floor plans using the PyQt5 canvas, which provides a flexible and intuitive interface for users to design and edit their floor plans. These plans can be saved in PNG and points in .bp format for easy sharing and further editing. The application integrates Blender APIs to automate the conversion of these 2D plans into detailed 3D models, ensuring accuracy and efficiency. Future enhancements include the development of walkthrough capabilities for 3D building models, enabling users to create immersive virtual experiences*

Keywords: 2D to 3D conversion, Floor plan design, Blender API, Computer-aided design, Architectural visualization

I. INTRODUCTION

The conversion of 2D blueprints to 3D models is a critical process in various fields such as architecture, engineering, and design. Traditional methods are often time-consuming and require specialized software and skills. DIMENDRAFT aims to streamline this process by providing an integrated solution that allows users to create, edit, and convert 2D floor plans into 3D models efficiently.

For architects and designers, the ability to quickly visualize and iterate on 2D designs in a 3D space can significantly enhance productivity and creativity. The current process involves manual conversion using complex CAD software, which can be cumbersome and prone to errors. DIMENDRAFT leverages modern technologies such as Python, OpenCV, Blender API, PyQt5, Matplotlib, and Django to create a user- friendly and efficient conversion tool.

Existing methods for converting 2D blueprints to 3D models often involve a series of manual steps that are both time-consuming and error-prone. These methods typically require users to have a high level of technical expertise in using specialized CAD software. This not only limits the accessibility of these tools to non-technical users but also significantly increases the time and effort required to complete the conversion process.

II. LITERATURE SURVEY

The transition from 2D to 3D conversion has undergone significant advancements, particularly with the development of computer-aided design (CAD) software, 3D modelling tools, and machine learning algorithms.

Joint 3D Scene Reconstruction and Class Segmentation by C. Hane et al. [1] introduces a novel framework for joint 3D scene reconstruction and semantic segmentation. By integrating geometric and semantic information, the authors propose a method that simultaneously recovers the 3D structure of a scene and classifies objects within it.

A Novel Technique for Converting Images from 2D to 3D using Deep Learning [2] explores the application of deep learning, specifically convolutional neural networks (CNNs), to convert 2D images into 3D representations. The authors demonstrate how deep learning can be used to infer depth information from 2D images, enabling the generation of 3D models.

Make3D: Learning 3D Scene Structure from a Single Still Image by Ashutosh Saxena et al. [3] is a system that uses machine learning to infer 3D scene structure from a single still image. The authors train a model to predict depth information from 2D images, enabling the reconstruction of 3D scenes.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 2, May 2026 Impact Factor: 8.2

Normalized Cuts and Image Segmentation by Jianbo Shi and Jitendra Malik [4] introduces the normalized cuts algorithm for image segmentation. This technique has been widely adopted in various image processing and computer vision applications, including 3D reconstruction.

SYSTEM ARCHITECTURE

The overall architecture of DIMENDRAFT integrates multiple technologies to provide a seamless and efficient conversion process. The key components include:

- PyQt5 Canvas: For creating and editing 2D floor plans
- Blender APIs: For converting 2D floor plans to 3D models
- OpenCV: For image processing tasks
- Matplotlib: For data visualization
- Django: For backend support and data management

The system operates through three main modules:

A. 2D Floor Plan Creation Module

This module provides an intuitive user interface using PyQt5, allowing users to create and edit 2D floor plans without requiring extensive technical knowledge. The PyQt5 canvas supports various drawing tools and features, enabling users to design detailed floor plans with precision. Users can make real-time changes with immediate visual feedback, and the system supports undo and redo functionalities.

B. 3D Model Conversion Module

The system leverages Blender APIs to automatically convert 2D floor plans into 3D models, significantly reducing the manual effort required for this process. The conversion process is designed to be efficient and accurate, ensuring that the resulting 3D models are of high quality and meet the design specifications. Users can view real-time 3D renderings of their floor plans and edit the 3D models directly within the system.

C. Data Management Module

The system manages various files and data in a centralized manner, including configurations, user input, system outputs, and logs. It stores user preferences and settings securely, records detailed logs of all activities, and includes robust backup and recovery mechanisms.

III. METHODOLOGY

The implementation follows a systematic approach:

A. 2D Floor Plan Creation Algorithm

Input: User inputs on the PyQt5 canvas Algorithm:
1. Initialize PyQt5 canvas for drawing interface
2. Process user drawing commands (lines, rectangles, etc.)
3. Apply real-time rendering and feedback
4. Save floor plan in PNG format
5. Store blueprint points in .bp format

Output: Saved 2D floor plans in PNG and .bp formats

B. 3D Model Conversion Algorithm

Input: 2D floor plans (PNG and .bp files) Algorithm:
1. Load 2D floor plan data

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 2, May 2026 Impact Factor: 8.2

2. Parse blueprint points and geometric data
3. Initialize Blender environment via API
4. Create 3D geometry based on 2D coordinates
5. Apply extrusion and modeling operations
6. Add materials and textures
7. Render 3D model
8. Export in standard 3D formats

Output: Rendered 3D models (.blend, .obj, .fbx formats)

SYSTEM IMPLEMENTATION

The system was implemented using Python as the primary programming language due to its versatility and robust library ecosystem. Key implementation details include:

A. Software Requirements

Python 3.8+ with required libraries
Blender 2.8+ for 3D modeling and rendering PyQt5 for GUI development
OpenCV for image processing Django for backend support PostgreSQL for data storage

B. Hardware Requirements

- Processor: Intel i5 or higher
- RAM: Minimum 8 GB
- Storage: At least 256 GB
- Graphics: Integrated GPU (optional for enhanced rendering)

C. System Integration

The modular design allows for easy integration of components. Python scripts coordinate between the PyQt5 interface, Blender API calls, and database operations. The system maintains data consistency through centralized file management and logging mechanisms.

IV. RESULTS AND TESTING

A. Functional Testing

The system underwent comprehensive testing across all modules:
2D Floor Plan Creation: Successfully validated the PyQt5 canvas functionality for creating and editing floor plans, with accurate saving in PNG format.
3D Model Conversion: Verified accurate conversion of 2D floor plans to 3D models using Blender APIs, with proper rendering and export capabilities.
Data Management: Confirmed efficient management of logs, user data, and media files with reliable access and recovery features.

B. Performance Results

- 2D floor plan creation: Real-time response with smooth drawing operations
- 3D conversion time: Average 30-60 seconds for typical floor plans
- File management: Efficient storage and retrieval of project data
- User interface: Intuitive navigation with minimal learning curve

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 2, May 2026
Impact Factor: 8.2

C. User Interface Screenshots

The system provides a comprehensive web-based interface including login functionality, project dashboard, blueprint management, 2D canvas for design, and 3D model viewing capabilities. The generated 3D models display properly in Blender with accurate geometric representation of the original 2D floor plans.

V. FUTURE ENHANCEMENTS

Several areas can be explored to enhance DIMENDRAFT's functionality:

A. Advanced AI Integration

Implementing deep learning techniques to improve conversion accuracy and handle more complex architectural drawings automatically.

B. Walkthrough Capabilities

Development of immersive virtual reality experiences allowing users to navigate through generated 3D models.

C. Cloud-Based Collaboration

Enabling real-time collaboration features for distributed teams working on architectural projects.

D. Enhanced Material Library

Expanding the system to include comprehensive material and texture libraries for more realistic 3D rendering.

E. Mobile Application

Developing mobile versions for on-site architectural visualization and editing capabilities.

VI. CONCLUSION

DIMENDRAFT successfully demonstrates its potential to enhance the efficiency and accessibility of 2D to 3D conversion processes. By leveraging modern technologies including Python, OpenCV, Blender API, PyQt5, Matplotlib, and Django, the system provides a comprehensive and user-friendly solution for converting 2D blueprints to 3D models.

The integrated approach significantly reduces the time and effort required for architectural visualization while making the technology accessible to users with varying levels of technical expertise. The successful implementation and testing validate the system's capability to serve as an effective tool for architects, designers, and engineers.

Future developments focusing on AI enhancement, cloud collaboration, and mobile accessibility will further establish DIMENDRAFT as an essential tool in the architectural design and visualization domain.

ACKNOWLEDGMENT

The authors express sincere gratitude to Dr. Sapna B Kulkarni, Professor, Department of Computer Science and Engineering, for her valuable guidance and support throughout this project. Special thanks to Dr. T. Hanumantha Reddy, Principal, and Dr. H. Girisha, Head of Department, Computer Science and Engineering, RYMEC, Ballari, for providing the necessary resources and encouragement for this research work.

REFERENCES

[1] C. Hane, C. Zach, A. Cohen, R. Angst, M. Pollefeys, "Joint 3D Scene Reconstruction and Class Segmentation," IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 97-104, 2013.

[2] "A Novel Technique for Converting Images from 2D to 3D using Deep Learning," IEEE Xplore, 2023.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 2, May 2026
Impact Factor: 8.2

[3] A. Saxena, M. Sun, A. Y. Ng, "Make3D: Learning 3D Scene Structure from a Single Still Image," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 31, no. 5, pp. 824-840, 2009.
[4] J. Shi, J. Malik, "Normalized Cuts and Image Segmentation," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 22, no. 8, pp. 888-905, 2000.
[5] J. Xie, R. Girshick, A. Farhadi, "Deep3D: Fully Automatic 2D-to-3D Video Conversion with Deep Convolutional Neural Networks," arXiv preprint arXiv:1604.03650, 2016.
[6] A. Criminisi, I. Reid, A. Zisserman, "Single View Metrology," International Journal of Computer Vision, vol. 40, no. 2, pp. 123- 148, 2000.
[7] D. Hoiem, A. A. Efros, M. Hebert, "Automatic Photo Pop-up," ACM Transactions on Graphics (TOG), vol. 24, no. 3, pp. 577-584, 2005.
[8] W. Xia, J.-H. Xue, "A Survey on Deep Generative 3D-aware Image Synthesis," arXiv preprint arXiv:2210.14267, 2022