3D Buildings from Floorplan

Michael Cardenas, Ronald Moore, John Mericle, José Orozco
One World Terrain
Orlando, Florida
michael.w.cardenas@leidos.com; ronald.moore@maxar.com; john.p.mericle@leidos.com; jose.l.orozco8.civ@army.mil

ABSTRACT

Constructing 3D geospatial terrain databases for live, virtual, and constructive simulation applications is costly and time consuming. To support the US Army's Synthetic Training Environment (STE) terrain database production requirements many new automated processes have been developed to lower costs and reduce timelines. One of these automated processes is procedural 3D building model generation.

Creating a procedural 3D building model generation capability for virtual simulation applications poses many complex challenges. Four main requirements were identified for the creation of 3D buildings: 1) ingest a variety of source data types and input formats, 2) handle missing or incomplete building descriptions, 3) generate 3D building geometry with desired appearance for both exteriors and interiors, 4) fulfill a broad range of fidelity and functional requirements for the runtime simulation systems.

This paper describes a methodology for procedurally creating 3D building models using a centralized floorplan-based common source input into a 3D building model generation process. This approach, 3D Buildings from Floorplans (3DBFF), utilizes a floorplan-based schema for defining a building in a common Drawing Exchange Format (DXF). This floorplan interface is assembled from a variety of disparate data sources including from real-world reference data such as building footprints, CAD floorplans, building blueprints, satellite captured imagery, and drone captured imagery; and can be generated from manual, procedural, machine learning, and generative AI processes. The paper will describe why the DXF format was chosen as the central format type, including advantages from CAD software support, metadata structures, and common workflow. The paper will also describe the processes to transform this common data format into a fully functional 3D model of real-world buildings, with functioning windows and doors, multistory structures and traversable stairs and ladders. Last, we will show the application of this approach in multiple use cases, and enumerate our successes, failures, and lessons learned.

ABOUT THE AUTHORS

Michael Cardenas is currently a Software Engineer working on the Synthetic Training Environment (STE) One World Terrain (OWT) program. He has over 10 years' experience working in the modeling, simulation, and training industry with expertise in software development, software integration and test, terrain database production, constructive training simulation, and virtual training simulation.

Ronald Moore is currently Maxar's Solutions Architect and OWT Chief Engineer on the STE OWT program. He has over 45 years of experience in the modeling, simulation, and training industry with expertise in software development, computer graphics, computer image generation, terrain database production, sound simulation, streaming video and audio, curated vectors, and game development.

John Mericle is currently a Software Engineer on OWT. John has worked in the simulations and training industry for over 30 years, with expertise in software development, software integration and test, terrain database production, constructive training simulation, and virtual training simulation.

José Orozco is the One World Terrain Lead Engineer. Previously, Lead Engineer for One Semi-Automated Forces with 30 years of experience supporting in various lead engineering and acquisition capacities within the M&S community. He was instrumental in development and integration of constructive capability to live, virtual, and mission command assets for OneSAF enabling platform to simulation training. He has received his BS/MS in Computer Engineering from Mercer University and has MBA from Wright State University and BBA from Polytechnic University of Puerto Rico.

3D Buildings from Floorplan

Michael Cardenas, Ronald Moore, John Mericle, José L. Orozco
One World Terrain
Orlando, Florida
michael.w.cardenas@leidos.com; ronald.moore@maxar.com; john.p.mericle@leidos.com; jose.l.orozco8.civ@army.mil

INTRODUCTION

This paper describes an approach for procedurally creating variable fidelity 3D building models, for virtual simulations, and from a variety of disparate sources. The approach is to normalize the various 2D and 3D sources into a standardized form and format that can be used by a procedural 3D floorplan building model generation tool.

The normalization approach enables a broad range of inputs including hand created Computer Aided Design (CAD) floorplans, blueprint-based floorplans, smart phone hand scan floorplans, Geographic Information System (GIS) footprints, and 3D polygon mesh geometry buildings from photogrammetric 3D reconstruction to be the source for the procedural model generation. All these inputs are transformed into a common floorplan-based source form and format. Using this floorplan-based approach, both the exteriors and the interiors of buildings can be described in the single data format and created in the floorplan-based procedural building generation tool.

The floorplan-based procedural building generation tool enables the creation of 3D building models with a range of capabilities. Some of the capabilities are defined in the floorplan data and some are driven by execution control. In the output terrain data, the 3D building models can vary in fidelity based on location within the terrain, training need, and target system performance.

This paper discusses our approach to creating 3D building models from a variety of sources. First, we discuss our 3D model requirements and our evaluation of the required sources. Next, we describe our standardized process for 3D Model Generation, introduce the normalized data format and summarize our floorplan-based procedural 3D building model generation tool. Also, we talk about our selection of our normalizing format. Last, we show the use of the tool in our use case and enumerate our successes, failures and lessons learned.

BUILDING MODEL REQUIREMENTS

Our procedural building model generation approach is driven by four requirements: 1) the synthetic training terrain functional requirements, 2) the requirement to accept a large variety of source data, 3) the capabilities of the target training systems rendering and reasoning systems, and 4) the performance limitations of these training systems.

Use Case

The Use Case entails selecting small geographic areas within the training ranges where buildings require interiors. Our initial capability focused functionality on creating "training buildings" with interiors with functioning windows and doors. A polygon is created in the curated vector process to identify the enhanced area where buildings with interiors are desired. Attributes can be included in the control vector feature to select building functionality, appearance, and language. Building models from floorplan are then created using one of three methods.

The first method involves generating buildings models from a DXF drawing of a specifically desired building. A DXF CAD floorplan is created manually from a myriad of source data formats. That floorplan is then used to generate a building model placed at a specific point location. Figures 6 and 7 showcase buildings generated in this way.

The second method involves creating building models from procedurally generated floorplans derived from GIS building footprints. Building footprints are input into our Floorplan Architect Tool to procedurally produce a floorplan that is then used to create a model. Figure 10 showcases buildings generated in this way.

The third method involves procedurally identifying building footprints that are good candidates for replacement with a floorplan model built from our library of general use DXF floorplans. Currently, this method is utilized for Conex container buildings that conform to standardized dimensions and layouts. See Figure 1.

Terrain Functional Requirements

The 3D building models in the terrain data must support the training objectives to:

- efficiently fill the background scene with rich content
- accurately act as obstacles to mobility in the maneuver and training ranges
- effectively provide landmark models for visual navigation
- selectively enable enterable structures for specific training exercises

Regionality

Procedurally created Buildings from Floorplan (BFF) models must accept regionality attributes such as building function, appearance and language. These attributes are then used to determine appropriate building visual styles and feature layouts. For some appearance types such as Conex container buildings, footprints identified as good candidates are replaced with a building model generated from a library of general use DXF floorplans. See Figure 1.

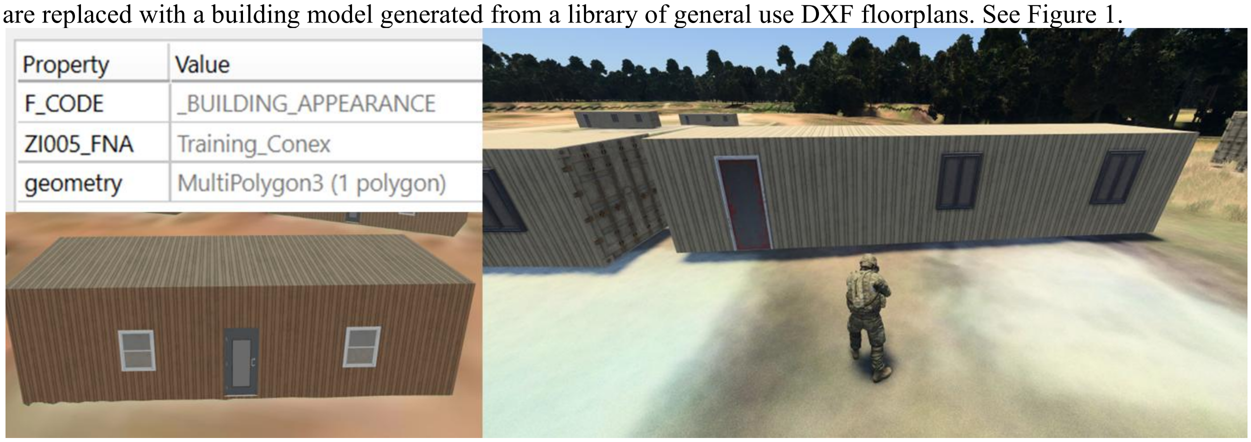


Figure 1. A "Training_Conex" appearance is attributed to a set of building footprints. Left: A Conex (container building) visual style is chosen, and a model is generated from a procedurally created floorplan. Right: These footprints are chosen as good candidates for replacement with a model produced from our DXF library

Source Data

To use the best available source data, we need an affordable way to accept the different sources and create a terrain database that looks consistent. To accomplish this, we evaluated the data sources to determine their structure and attribution.

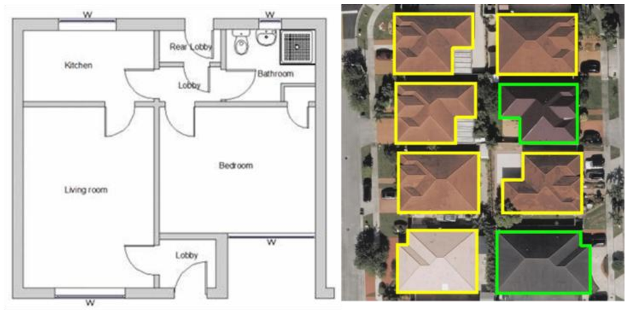
Footprints

The most common source for procedural model generation is 2D footprints. (This source is used in the Base Globe data stratum.) It is generally the roof outline projected to the ground. The footprint feature is a list of "x and y" vertices forming a closed polygon/surface. The footprint feature provides a generalized size and shape of the building, typically has a few map-friendly attributes including building height and building function, and perhaps includes the dominate color and primary building material. If it is a 3D footprint, it will include the "z" value at the intersection of the projected roof outline to the ground surface for each of the vertices. For tiered structures, they can include multiple 3D footprints representing the multiple roof lines of the structure.

CAD Floorplans

CAD floorplans are detailed, scaled drawings that illustrate the layout of a building, showing the relationship between different rooms, floors, walls, windows, and doors for the purpose of enabling real world building construction. CAD drawings provide much more detail than is typically used in modeling, simulation, and training (MS&T) applications.

Only a small subset of a CAD drawing content is required for our procedural model generation. Figure 2 provides an example of a CAD floorplan and GIS footprints.



CAD Floorplan GIS Footprint

Figure 2. CAD Floorplan and GIS Footprints Example.

Blueprints

In architecture, a blueprint is a detailed guide for constructing a building, and outlines the dimensions, materials, construction methods, and other important information necessary for construction. Blueprints are generally available for all buildings because they are required by land use administrators. Today, most are still stored in paper form or digitized as raster images. For our use, blueprints must be converted to CAD drawings. To-date this has been a manual process. But once converted, blueprints become CAD floorplans.

Satellite Captured Imagery

The process for a satellite capture of a building requires photogrammetric 3D reconstruction and 3D feature extraction. The extracted building is a 3D polygon mesh depicting the volume of the building. The mesh has the original imagery from the photogrammetry process projected onto the exterior surface. The polygons of the mesh have rudimentary identification of components, identifying wall and roof polygons. The height is captured in the geometry. The extracted building also has a footprint, or stack of footprints. These footprints include attribution from the mesh, like roof type, and adds dominate roof and wall color. Artificial Intelligence/Machine Learning (AI/ML) is being used to extract details from the 3D Polygon Mesh, identifying building function, apertures, and appendages, and extracting materials and colors.

Drone Captured Imagery

Like satellite captured imagery, the process for drone captured building requires photogrammetric 3D reconstruction and 3D feature extraction. The extracted building is a 3D polygon mesh wrapped in the drone imagery with identification of components (wall and roof polygons), but at a much higher resolution than the satellite captured imagery. The drone extracted building has a footprint, or stack of footprints. The mesh may also have building function, apertures, appendages, materials, and colors, again enabled by AI/ML extraction.

System Capabilities

The 3D building models must support a broad range of usage within the training applications, to include rendering of an out the window visual scene, rendering of a sensor scene, including infrared and low light, as well as painting of the building icon or footprint in a Plan View Display (PVD) or map display. Additionally, the 3D building models must support the reasoning systems including supporting collision detection, obstacle avoidance, and in the case of building interiors navigating between rooms, entering through doors and windows, and even breaching walls.

To get a complete picture of the functionality needed in the 3D models we identified the tactical tasks expected to be performed and derived functional requirements for the various components of the 3D buildings based on the tactical tasks. Table 1 provides a top-level summary of this analysis.

Table 1. Functional Requirements Analysis.

Tactical Task \ Runtime Capability	Door Construction	Door Open Method	Window Type	Window Open Method	Room Connections	Floor Transition Type	Level Transition Type	Roof Access	Construction Material	Surface Material	External Wall Thickness	Internal Wall Thickness	Room Function	Model Asset	Surface Asset
Routing															
Pedestrian					■			■							
Vehicle Movement					■	■	■								
Vehicle Storage					■	■	■								
Drone			■	■	■		■	■							
Robot		■			■	■	■	■							
Rappel			■	■	■										
First Person		■		■		■	■	■						■	■
Damage															
Protection	■								■		■	■			
Rubble									■		■				
Analysis															
Radiation Protection		■		■					■		■	■			
Chem/Bio Exposure		■		■	■				■		■				
Chem/Bio Propagation		■		■	■										
EMI	■								■		■	■			
Sniper															
Roof								■							
Window			■	■											
Loop Holes									■		■				
Clear Building	■	■			■								■		
Breach Entry									■		■				
Bunkers									■		■				
Smoke				■	■	■									
Visual Processing	■	■	■	■		■	■			■				■	■

System Limitations

The 3D building models must support the target rendering and reasoning systems that have varying computing capacities and require adjustable fidelity of the 3D building models to work within the performance limitations. This means being able to scale the complexity and functionality in the 3D building model creation and including attributes in the 3D building models that supports build-time and run-time processing load management.

SOURCE COLLECTION AND DATA REPRESENTATION AS FLOORPLAN

Normalized Data Format

For the normalized abstract representation of the buildings, we selected a CAD format. We experimented with multiple CAD tools to ensure that we could use open source or commercial tools to create, view and edit the CAD drawings. A specification was written to describe the required floorplan data and how to encode the required metadata. The goal was to align the minimum data we needed for our MS&T applications with the data typically found in building Blueprints. The specification is written around the DXF format (AutoDesk, 2024). DXF is a CAD data file format designed for sharing drawing data universally across CAD applications. Our Procedural Building Specification describes the minimum information required to build a shell structure and provides the details if very specific exteriors and interiors are desired. Figure 3 below provides a sample of the procedural model floorplan.

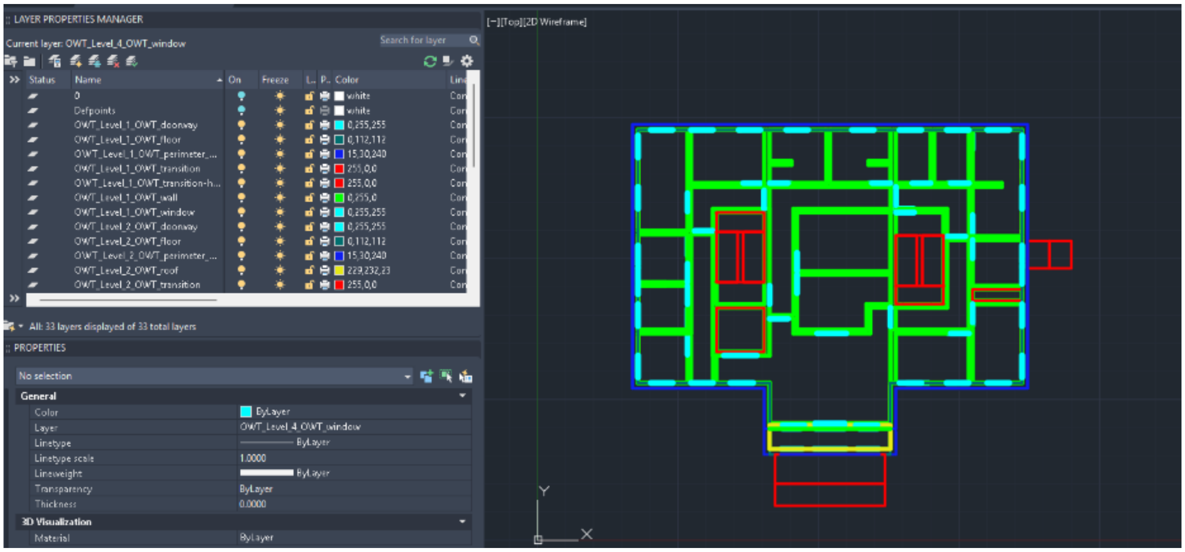


Figure 3. Procedural Model CAD floorplan

Why a CAD Format

Before we selected DXF as our format and before we created our procedural model generation tools, we reviewed previous work. Two previous activities were of note: Computer Generated Architecture (CGAs) (Esri, 2024) from Esri's City Engine and the construction grammar developed by Blackshark.ai (Blackshark.ai, 2024) for the Microsoft Flight Simulator game.

Esri CGA's - CGA is a programming language that is used to instruct CityEngine how to automatically generate architecture 3D geometry. On the positive side, CGAs are mature and in use by a commercial product, CGAs are well defined, and CGAs can provide a good definition of a building, both exterior and interior. On the negative side, CGAs are very complex and require programmer levels skill to create, the tool to create CGAs are rudimentary, and the GCA specification is owned by Esri. Without broader adoption, CGAs do not meet the criteria for Open Geospatial Consortium (OGC) community standards adoption. We were looking for a format to describe the buildings with common, familiar methods and tools. In the end, CGAs did not allow us to describe the buildings easily from multiple sources.

Blackshark.ia Construction Grammar - From the presentations and articles, the Blackshark.ai grammar language appeared very interesting. Unfortunately, the grammar language is part of a patented process and not publicly shared. In the end, we selected a format that is broadly used, supported by many commercial and open-source tools, and easily encompasses the information necessary to describe a building for MS&T applications.

Collating Source as a DXF Floorplan
Many source data formats can be imported into CAD software directly as reference when creating a floorplan.

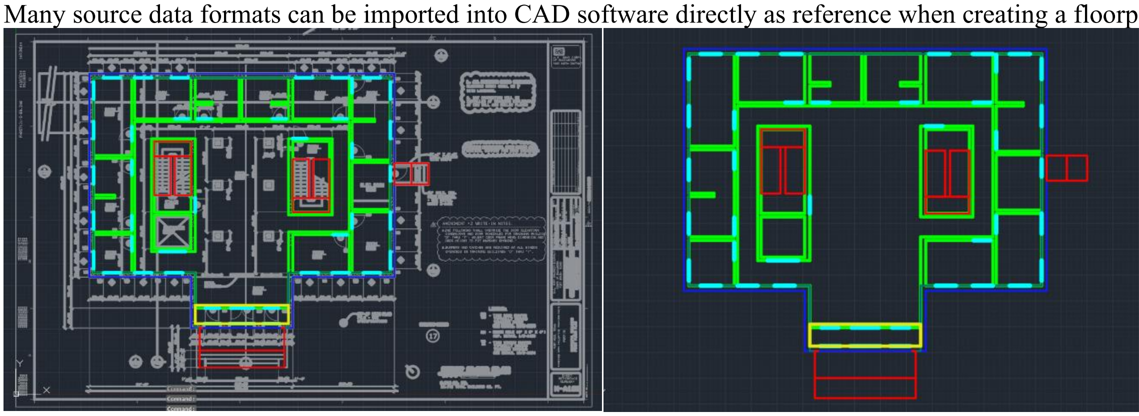


Figure 4. Floorplan created from architectural blueprints imported into CAD from PDF

BUILDINGS FROM FLOORPLAN DETAILED

Floorplans are drawn in 2D with a few exceptions (e.g. transitions between levels are effectively 2D but have some z values for ease of entry). A combination of layer naming conventions, entity types, geometry, XData, default values, and smart assets form the rules and components for conversion to a 3D model. The front of the building is assumed to be parallel to the x-axis. Y values increase moving from the front of the building to the back.

Levels and Layers

Levels are used to group entities and their data by stories of a building. All layers are named according to the level of the building containing that feature and the feature's type. For above ground levels, layer names begin with the prefix "owt_level_#_", where '#' is the level value that must increase with elevation but may skip integer values (i.e. level 13 may be skipped). For below ground levels, layer names begin with the prefix "owt_level_b#" where '#' is the level value that must increase going down in elevation. Level specific layers are named and grouped by level id. They are given the postfix that designates feature type (e.g. owt_level_1_owt_doorway is a layer containing level 1 doorways).

XData

Extended data (XData) is a DXF tags structure to store arbitrary data in DXF entities. It is used here to override entity default values. If the default values of the building style are correct, it is not required to override them. XData is associated with a drawing entity (e.g. line, polyline, point). Each entity belongs to a layer which defines its function. The XData is organized as [key:value] pairs for each entity's attribution. The allowable attribution is dependent on the entity type (e.g. a doorway has different attribution than a perimeter wall).

Figure 5. (Left) Style using default values for floor. (Right) Replaced floor with a different material using XData

Required Layers for Minimum Building Representation

The following layers represent the minimum required information to represent a building:

Perimeter
A perimeter wall describes the outer surface wall of the building. The perimeter wall layer may contain multiple closed perimeter polygons. Each closed polygon may be formed with a combination of lines, polygons, and 3D polygons (where the z-value represents elevation information). The reason to break a perimeter polygon into components is due to its associated XData. A single surface may need unique surface details. The bottom edge of a polygon may need to follow a non-planar footprint which would be best represented as a 3D polygon. Any changes to the perimeter material between levels is handled by creating a new perimeter wall at that level. Perimeter walls are extended up until there is another perimeter polygon that contains that perimeter or is contained by it. If no other perimeters are found, the surface extends to the top level. Parapets are added if the top level has a nearby parallel interior wall with an offset height. In the case of parapets, a top surface is created to join the perimeter and interior wall surfaces.

Fig. 6. Perimeter wall of a Building from Floorplan with varying materials

Roof

The roof layer may contain multiple closed polygons. The level should be above the floor that it is covering. A one-story building will have a level 1 for the ground floor and a level two with the roof. Our current implementation supports flat, gable, and hip roof types. Roof attributes define the deck surface, and can include soffit, fascia, gable interior and gable exterior surfaces. Gable roofs revert to a hip roof type if all deck surfaces share a common vertex, which happens when all sides are equal (e.g. an octagon). Roofs are allowed point entities for placing models (e.g. dome, spire)

Fig. 7. (Top Left) BFF with hip asphalt shingle roof with white painted wood fascia (Top Right) BFF with gable roof using same defined assets. (Bottom Left) Flat concrete roof with parapets (Bottom Right) BFF with flat roofs and dome models

Floor

The floor layer can contain one or more closed polygons. The entire floor surface should be entirely covered by the combination of these closed polygons. If a hole is needed, a transition hole should be added to the transition layer. Floors can be offset from the level elevation to create raised and lowered floors.

Wall

Wall surfaces form the interior walls of the building. This includes insides of parapets which are considered interior surfaces. Wall surfaces must be able to be combined to create closed polygons. Wall polygons are used to create room cells. Because of room nesting, not all wall polygons form a room, they may represent the external surface of a nested room.

Optional Layers

Structure
Structure layer is a top-level layer not associated with a level. No geometry is used from entities on this layer. Since only XData is used, any entity type can be created (e.g. point, line, polyline) with associated XData. This layer allows the selection of default values defined in the style matching the style id specified in the floorplan buildings catalog.

Ceiling
If no ceiling is supplied, the floors defined in the level's floor layer are offset and used for ceiling surfaces. Default ceiling parameters are used from the current building style. Optionally, the ceiling surfaces may be defined but all ceiling surfaces of the level must then be defined.

Doorway
Doorways are defined by a line that is parallel and close to a wall. Door opening size is determined by line length in conjunction with the default values from the current style or any provided XData values. Doorways create connections between rooms.

Window
Like doorways, windows are defined by a line that is parallel and close to a wall. Window opening size is determined by line length in conjunction with the default values from the current style or any provided XData values.

Transition
The transition layer defines any feature used to travel between levels of a building. Transition types include holes, stairs, ladders, and spiral stairs. Transitions are defined with 2D polygons (with elevation attribution) or 3D polygons. The elevation or z-values are used to determine beginning and end point of the transition. Transitions are assembled by the Floorplan Model Generation Tool from model kits in our building from floorplan asset library.

Fig. 8. A ladder, spiral staircase, and stairwell created by the Floorplan Model Generation Tool using the transition layer information defined in the floorplan

STANDARDIZED 3D MODEL GENERATION PROCESS

Our objective is to have the runtime terrain 3D building models look realistic and consistent regardless of the source. We determined that a common procedural building generation process for creating the 3D structures would be the most-affordable and result in the most-consistent output. We also determined that a common format would be the best approach to feed our procedural building generation process. The diagram below, in Figure 9, captures the process. The GIS Footprints and Satellite 3DSM sources are part of a fully automated production process, with other sources requiring manual actions. When multiple sources are available for a building, the more detailed/specific source "knocks out" the less detailed/specific source.

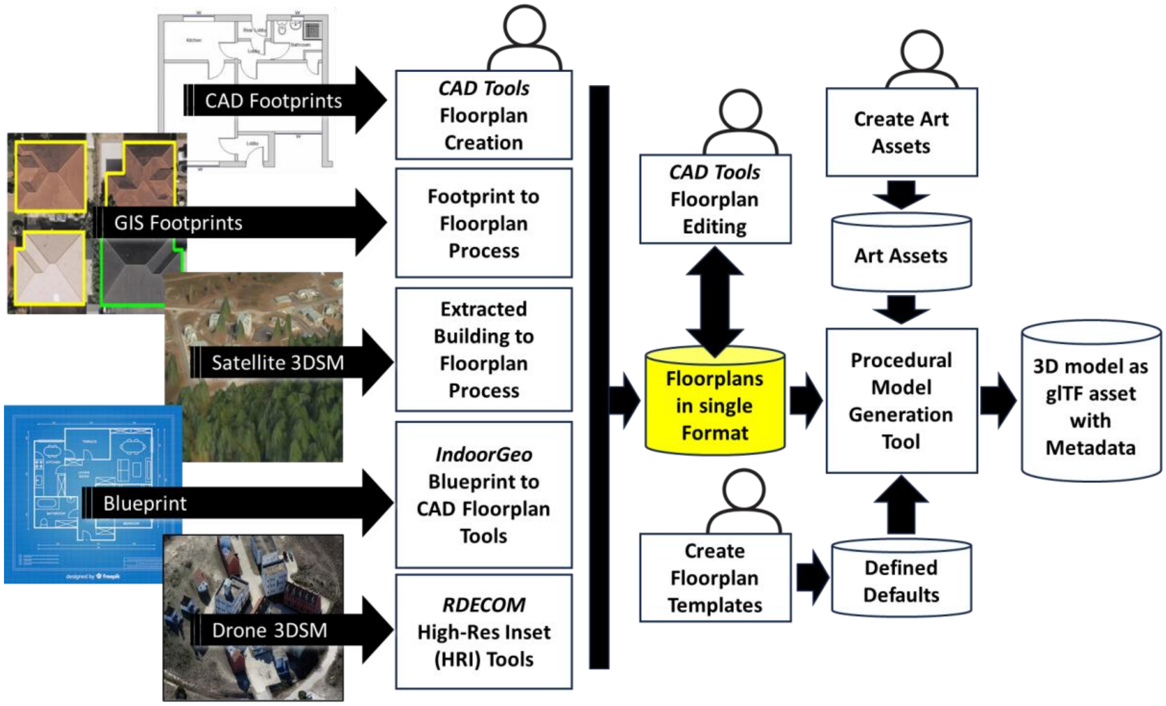


Fig. 9. 3D Model Generation Process.

Floorplan Asset Library

To support the 3D model generation, textures were created to represent exterior and interior surfaces. Also, 3D model kits were built to provide the complex geometry of window frames and windows, door frames and doors, and parts of complex structures like stairs, stair rails. etc. Model kits were also built to provide complete models of small features used on and within a structure, like standpipes and light fixtures. The models are coded to support scaling, duplicating and repeating placement and are also built with material code, articulation, hinges, action volumes, navigation volumes, and collision information for use in the runtime system.

Floorplan Style Catalog

The Floorplan Style Catalog provides a listing of available material and model assets for use in building creation, as well as defines a range of commonly used door, window, roof, ceiling, floor, interior wall, perimeter wall types and associated attributes and default values. These various types are collated together as a building style and assigned a style id as well as tagged with regionality information defining the style's appearance, language, function and more. Floorplans attributed with a style id inherit all necessary metadata information for 3D building generation from the style definition.

Floorplan Model Generation Tool

Our procedural floorplan-based 3D model generation tool is built to accept this normalized data and create 3D buildings representing the structure as described in the floorplan. Templates are created to provide an appearance pattern for the model construction, based on information in the curated vector data regarding regionality.

To manage the runtime performance on the target rendering and reasoning systems, the floorplan model generation tool is embedded in our overall 3D Terrain Pack generation process. This allows the content creator to control where and what functionality is included in the 3D building models. 2D vector polygons are placed within the curated vector data to define where buildings are created and with what functionality they are generated.

Floorplan Architect Tool

As mentioned above, the most common source format for procedural model generation are footprints. The Floorplan Architect Tool is built to ingest these footprints and procedurally create geo-typical floorplans for use by the floorplan model generation tool to generate 3D buildings with "ballroom" interiors. This forgoes the need for manual DXF creation, instead the floorplan is created based on the footprint size and shape as well as other provided vector information such as nearest roadways, appearance, and language. For multi-story buildings the architect will define stairwells or spiral staircases to allow access between floors. For buildings with a flat roof, a ladder can be added to allow roof access. Figure 10 showcases 3D building models created from procedurally created floorplans from the architect tool.

Fig. 10. Buildings from floorplans created by the Floorplan Architect Tool from 2D footprints

SUCCESSES, FAILURES, LESSONS LEARNED

The effort has been successful. We can generate very realistic building models in a fully automated process from the Satellite Captured Imagery, and Geographic Information System (GIS) footprints. We have tested floorplans from smart phone hand scans and Blueprints using IndoorGeo AI/ML footprints (IndoorGeo, 2024). Additionally, we can place manually created floorplans as a referenced model, and procedurally create the buildings in the automated process. Figure 10 shows a picture of some of the buildings that have been generated using the process.

We tested generating drone collected floorplans using Shield.ai's automated exterior/interior reconnaissance drones (Shield AI, 2024). Shield.ai provides an extremely successful drone swarm for building clearing, depicted in Figure 11. Modifications were made to the drone to capture the floorplan and a surface model. Unfortunately, Shield.ai has discontinued drone interior collection efforts.

Fig. 11. Shield.ai Nova 2.

The scanning of a building using a handheld phone is yet to be complete. Most of the smartphone apps we have tested collected one room at a time very well. We have not developed an automated process to collect individual rooms and composite then into a complete floor; composite the floors into a complete building interior; collect the exterior and put it all together into a complete building floorplan. To date, only manual composition has been accomplished.

A few observations were made as we developed the capabilities.

- Bad footprint Geometry - We noted that the fully automated building extraction from satellite imagery does not always create footprints that can be turned into buildings due to malformed geometry. Similar issues were encountered from opensource GIS footprints. The "bad" footprints require modifications from original source to make them valid for producing buildings without geometry errors.

- Additional attributes needed - To generate complete buildings with regional appearance requires more than just building function – it requires secondary attribution for local styles.
- Model kits - Procedural creation of the intricate aspects of doors and windows is better provided with sizable model kits. Models have been built with articulation, collision, navigation volumes, action volumes, hinges, and material encoding information for use by the runtime systems.
- Mix of tintable and full color textures - Tintable monochrome textures is very efficient for the runtime system memory but limits each component to a single color – supporting both full color and tintable textures provides a more realistic look.

Developing the floorplan-based procedural models with the known functional requirements made it much easier to support all the data needed by the target rendering and reasoning systems.

FINAL WORDS

There is no universal solution for creating 3D building models that supports all the requirements – many compromises are required. Our procedural 3D model generation process represents our attempt at addressing this complex problem. With this processing framework in place, adding additional functionality, like damage and destruction, can easily be added.

ACKNOWLEDGEMENTS

Thanks to the One World Terrain engineering team for their hard work on the floorplan-based procedural model generation development project.

REFERENCES

AutoDesk AutoCAD 2012, (2024). *DXF Reference*. https://images.autodesk.com/adsk/files/autocad_2012_pdf_dxf-reference_enu.pdf

Blackshark.ai, (2024). *Announcing the Launch of Metaverse Standards Forum with Blackshark.ai*. https://blackshark.ai/metaverse-standards-forum/

Esri ArcGIS CityEngine, (2024). *CGA reference*. https://doc.arcgis.com/en/cityengine/latest/cga/cityengine-cga-introduction.htm

IndoorGeo, (2024). *Technology*. https://indoorgeo.com/technology/

Shield AI, (2024). *Powered by Hivemind Nova 2*. https://shield.ai/nova-2/