Exterior Attribute Extraction and Interior Layout Speculation of 3D Structures

Ronald G. Moore, Matthew J. Reilly
Leidos, Inc.
Orlando, FL
ronald.g.moore@leidos.com, matthew.j.reilly@leidos.com

Tony Pelham
GameSim, Inc.
Orlando, FL
tony.pelham@gamesim.com

ABSTRACT

Automated collection-to-construction of terrain databases is a critical capability envisioned for future U.S. Army training systems. The challenge is how to automatically produce terrain data that supports both visual rendering and simulated reasoning with content sufficient to train ground forces in dense urban environments.

The process of automated terrain construction begins with surface capture. Drones and ground-robots are deployed, capturing large amounts of raw surface data. Processing the surface data yields point clouds or 3D polygonal meshes, providing an initial 3D terrain model, typically with very high point/polygon densities and large raster memory requirements. While certain applications may be able to utilize these terrain models directly, most visualization applications, require additional processing to generate well-formed model geometry, sharp textures, door and window apertures, and material classifications. This additional processing, performed on the point cloud or 3D polygonal mesh, extracts point, line, and polygon feature geometries along with descriptive feature attributes (e.g., height, roofline, roof-type). A bare earth elevation model is generated to provide a ground surface in which to place the extracted 3D features. The final enabler of the terrain construction process is the automated generation of 3D models from the feature and attributed data.

This paper reports on research which expands automated extraction of attributes from images through deep-learning and image processing techniques, identifying structural dimensions, apertures (e.g., doors, windows), appendages (e.g., A/C-Unit, chimneys), colors, and materials. From this set of enhanced attributes, geo-representative 3D models are procedurally generated. In addition, from the same set of enhanced attributes, geo-representative building-interiors are speculated and procedurally generated. This paper details these image processing and deep-learning techniques, describes the enhanced feature attributes that are extracted, explains the methods for interior speculation, and details the techniques for procedural 3D model generation. The paper provides lessons-learned and recommends a new standard for procedural model generation.

ABOUT THE AUTHORS

Mr. Ronald Moore is currently the Chief Architect on SE Core CVEM. He has over 35 years of experience in the modeling, simulation and training industry with expertise in software development, computer graphics, computer image generation, simulation geospatial terrain database production, sound simulation, streaming audio and video, and PC and console game development.

Mr. Matthew Reilly is currently a senior software engineer on SE Core CVEM. He has over 20 years of experience in software and hardware development with a focus on the computer graphics domain. Previous experience includes CCTT and AVCATT development for the SE Core Architecture and Integration (A&I) Program, the WARSIM Program, development of console video game titles, avionics instrumentation, and cellular technology.

Mr. Tony Pelham is currently a senior software engineer at GameSim. Tony has over 30 years of experience in the modeling, simulation, and training industry with expertise in computer image generation, visual simulation, distributed simulation, 3D database development tools, virtual world development, game design, procedural road generation, and procedural model generation.

LPCO1-18-ASG-0622-0039

Exterior Attribute Extraction and Interior Layout Speculation of 3D Structures

Ronald G. Moore, Matthew J. Reilly
Leidos, Inc.
Orlando, FL
ronald.g.moore@leidos.com,
matthew.j.reilly@leidos.com

Tony Pelham
GameSim, Inc.
Orlando, FL
tony.pelham@gamesim.com

INTRODUCTION

This paper reports on research to use image processing and deep learning technologies to extract structural dimensions, architectural elements, apertures (e.g., doors, windows), appendages (e.g., A/C-Unit, chimneys), colors, and materials from building pictures, drone captures terrain meshes and point clouds. From these extracted feature attributes, geo-representative exteriors are identified, plausible interiors are speculated, and 3D building models are automatically procedurally generated.

Our approach expands the traditional geospatial database production process to incorporate new technologies without abandoning current training systems and without compromising current production capabilities. These new technologies provide incredible automation and fantastic geospatial content, but they are not yet universally scalable due to the limited availability and affordability of source data.

The paper begins with a short background on Automated Feature Extraction (AFE) and procedural model generation (PMG) technologies, and highlights references to related research. Next, we review our use-cases—focusing on the costliest functions within the current terrain database production processes. Following the use-cases, we discuss the reasons for constraining our approach to fit within the existing database production process, discussing the technology limitations, explaining our procedural model generation technique, and introducing our automated processing pipeline. We next present our approach for extracting information from images, terrain meshes and point clouds, and explain our method of coupling the image attribute extraction with our procedural model generation. We then introduce our method for interior layout speculation. We end with a short lessons-learned and recommendations for further research.

BACKGROUND

In 2006, Philip Kern, researcher in earth imaging, wrote that “Automation of the feature extraction process has been the ‘Holy Grail’ of the photogrammetric data collection industry for many years.” Kern recounts the many years AFE has been worked, and he concludes that “we are close, but we haven’t found the Grail yet” (Kern, 2006). Five years later, Jarlath O’Neil-Dunne, editor at LiDAR Magazine writes that “Automated feature extraction has long been considered the Holy Grail of remote sensing, but for decades there has been relatively little to show for the untold millions, perhaps even billions, of dollars that were invested in this technology.” Jarlath suggests that sensor limitations are no longer the barrier to AFE, and that new techniques for image classification are improving the feature extraction process (O’Neil-Dunne, 2011). Fast forward to 2016, and Deepakrishna, Ediriweera, and Gunatilake, researchers in surveying and mapping, demonstrate automatic and efficient extraction of feature information from high spatial resolution imagery using an Artificial Neural Network (ANN) (Deepakrishna, 2016).

In 2015, SE Core demonstrated using procedural generation tools to create 3D building models (Eckman, 2015). Figure 1 provides an example of a procedural generated multi-state 3D model.

Figure 1 Procedural 3D Model Generation

For modeling and simulation (M&S) the goal is more than automated feature footprint extraction, procedural 3D model generation, and affordable high resolution photorealistic terrain mesh. For M&S the goal is automated collection-to-construction of terrain data, supporting visual rendering and simulated reasoning with content sufficient to train ground forces in dense urban environments. Our research began by identifying the technology required to support a collection-to-construction pipeline, finding the gaps, and determining the best methods to address those gaps. Consequently, our approach is to leverage drone mapping, photogrammetry, image segmentation and classification and AFE technologies, and combine them with new automated enhanced attribute extraction technologies and use new content with the goal to reduce costly processes within the traditional geospatial terrain database production process.

USE-CASES

SE Core has been producing terrain databases for U.S. Army training systems for more than 13 years. During this time, SE Core engineers have developed a mature knowledge-base of database content requirements for visual rendering systems (plan-view-display, digital maps, out-the-window visual, and sensor visual) and simulated reason systems (semi-automated forces, course-of-action analysis, line-of-sight, and physical interactions). From this experience, three use-cases were identified in which automated processes could achieve significant cost reductions.

Use-Case One: Training Sites

A primary component of an SE Core terrain database are the high-detailed training areas—identified as the 1:12.5K scale area-of-interest (AOI). These are typically Military Operations on Urban Terrain (MOUT) sites within the military installation where live training exercises are performed. Each MOUT sites is faithfully replicated using blueprints, layout drawings, and facility expert discussions. Additionally, SE Core personnel visit each site to capture full-color photographs and videos. Such site data is used by the SE Core modeling team to hand-create each of the 1:12.5K scale buildings, where each building includes modeled interiors. Creation of these 3D models using touch-labor is very costly.

Our first use-case utilized such site-trip data (photos and video), along with our research tools to automatically create geo-specific 3D building models with usable interiors. In addition to the site-trip data, we also obtained from the University of Southern California (USC) Institute for Creative Technologies (ICT), a drone-captured polygon mesh of the West Point MOUT site.

Use-Case Two: Military Installations and Surrounding Cities

The second-most costly area, when producing a military training area terrain database, is the generation of the cantonment areas surrounding training sites. These typically include the built-up-area or region of the military installation and the cities surrounding the military installation. On SE Core this is referred to as the 1:25K scale AOI.

Our second use-case is to create these 1:25K scale AOIs with geo-representative buildings as automatically as possible. For our experiment, we worked with Google to get access to a Google Earth terrain mesh and full-color street view data. The Google data we received encompassed a section of downtown London, England with approximately 0.1-meter resolution imagery projected onto a 3D polygon mesh. The Google Street View data was accessed directly from Google's online repository using a Google-provided application programmer's interface (API).

Use-Case Three: GFT Standalone Training Database

In the course of producing a terrain database for a specific military site, SE Core will often receive requests to produce a companion standalone Games-For-Training (GFT) database to support specialty training. These standalone GFT databases are generated in Virtual Battlespace 3 (VBS3®[1]) format and include 3D building models, complete with interiors. Unlike MOUT-type buildings which require open doorway and window apertures, GFT buildings contain door and window models, which must be capable of being opened appropriately during an exercise. Creating GFT buildings using traditional 3D-modeling tools is costly and time-consuming.

Our third use-case demonstrates our ability to use cell-phone-captured images of building exteriors as input data for our experimental tool, automatically creating a GFT 3D building model, with interiors, and placing the generated model into a VBS3 database, without modeler intervention.

[1] VBS is a registered trademark of Bohemia Interactive Simulations within the United States and/or other countries.

PRODUCTION PROCESS CONSTRAINTS

The challenge of our research was to create a capability that could integrate into the traditional training terrain database production process. Figure 2 captures the primary data flows and processes performed in the generation of a typical SE Core terrain database. The blue boxes represent the current runtime database production process. Boxes in green depict the addition of the capability to automatically extract enhanced attribute information and speculate the interior layout of the procedurally-generated 3D building models.

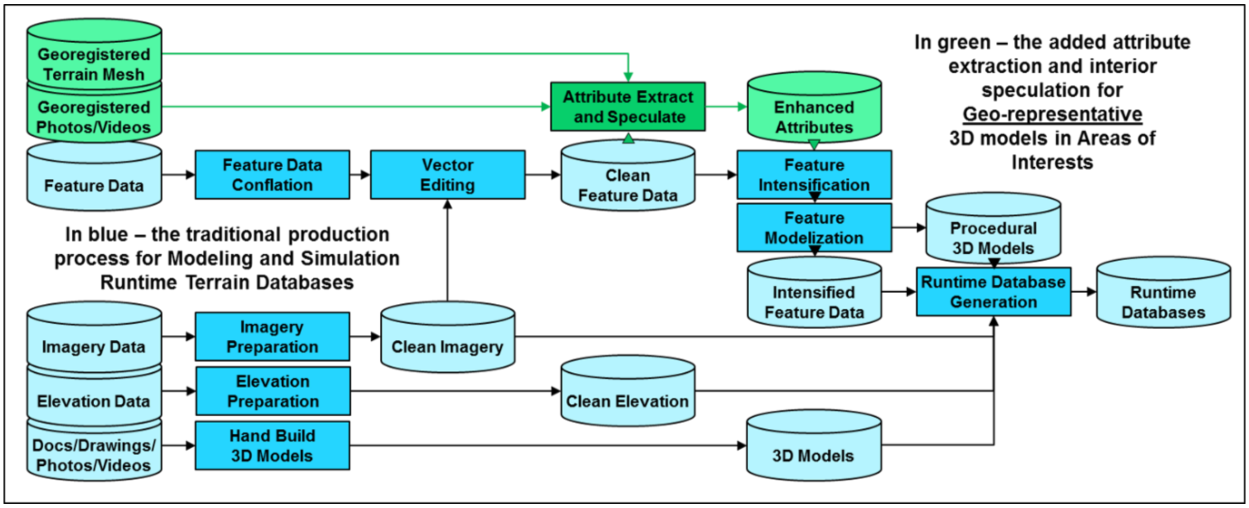

Figure 2 Traditional Database Production Process with Enhance Process

The importance of our approach—that of extracting additional attributes and procedurally constructing the 3D model—may not be obvious to the reader. Simply put, the 3D building models must have all the required geometry and attributes to support the visual rendering systems and the simulated reasoning systems. Additionally, 3D models must include working doors and windows, artificial intelligence routes, shadow volumes, collision volumes, etc. Of equal importance, 3D building models inside an AOI cannot look or behave differently than 3D building models outside of an AOI; this is to prevent the introduction of artificial cues to the trainee regarding the importance of a specific geographic area. Lastly, and most importantly, the representation of a building in feature form is much more usable and flexible than the "bag of polygons" 3D model representation.

TECHNOLOGY LIMITATIONS

Our research began by developing an understanding of the limits and strengths of the current photogrammetry, and Structure-from-Motion (video) technologies. Generation of a terrain database from imagery can result in realistic looking geo-specific buildings, especially when viewed from a distance. Figure 3 shows a Google Earth view of London.

Although such data can be visually impressive, and potentially useful as source data for training database generation, this type of source data can be very expensive to procure and source data for many desired locations may not be available, thus limiting universal adoption.

Figure 3 Google Earth Perspective View

Usability of Photogrammetry-Derived Terrain

Photogrammetry-derived source data can also suffer from a number of artifacts. Figure 4 shows a view of Google Earth in an area of London illustrating one of the limitations of photogrammetry-derived terrain – poorly formed geometry. Realistic buildings for training should obey basic rules of building design, i.e., have straight edges, vertical walls, and roofs that are separate and distinct from walls.

Figure 4 Google Earth Close-up View

Photogrammetry-derived terrain can result in many hundreds or thousands of polygons. Excessive polygon counts can impact real-time rendering performance and streaming throughput. Polygon mesh simplification can reduce polygon counts, but it is difficult to do without compromising feature shapes.

Photogrammetry-derived terrain requires unique textures for all surfaces, easily generating unmanageable volumes of texture data for a relatively small area, impacting storage requirements and streaming throughput. For web browsing, slow texture updates are acceptable. For real-time training applications, slow texture paging is unacceptable.

Achieving an acceptable texture resolution for geo-specific buildings for training in dense urban terrain can be challenging. Even the highest-resolution Google Earth imagery is far too low for most training scenarios. A brute-force solution of increasing the texture resolution imposes new constraints on source data acquisition and radically increases runtime texture storage and streaming throughput requirements. Figure 5 and Figure 6 contrasts the high-resolution image of Google Street view and the low-resolution texture of the Google Earth reconstructed polygon mesh.

In addition to the issues described above, such reconstructed meshes have windows and doors that are only defined by texture colors, which are of limited usefulness in many training scenarios.

Figure 5 Google Street View Image

Figure 6 Google Earth Building

Clear, unobstructed imagery from collected sources is difficult to achieve. Without careful processing, the occulting objects can become embedded into the building textures, as seen in Figure 7, from reconstructed drone imagery. An even worse case is illustrated in Figure 8, where occulting tree coverage was of sufficient density to distort the building geometry reconstruction in addition to texture embedding.

Figure 7 Tree in Building

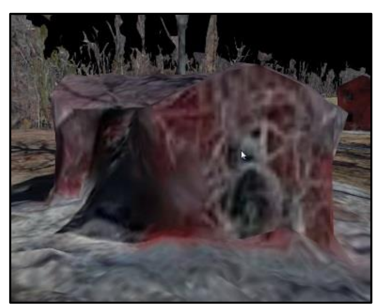

Figure 8 Deformed Building

Affordability of Photogrammetry-Derived Terrain

Obtaining reconstructed building meshes for large-areas utilizing brute-force techniques can become prohibitively expensive. Obtaining the raw source data may require contracting to data providers to utilize specialized drones, aircraft, or other equipment to obtain source data at the qualities and resolutions desired.

Availability of Photogrammetry-Derived Terrain

Source data at the required resolution and fidelity are not always available, due to security, logistical, or other data use considerations. Some standard sources of commercially available data such as Google datasets are not available for secure and military sites. Modeling and generation of training databases for military bases, MOUT training sites, and associated cantonment areas require specialized access and secure data handling processes which may rule out many commercial data providers.

PROCEDURAL MODEL GENERATION

Procedurally-generated models are constructed using a set of operators and construction primitives, along with a specialized building grammar. This procedural grammar includes functionality such as extrusion operators to create building volumes from footprints, slicing operators used to delineate building floors and to carve out apertures from walls, roof operators creating hip and gable roofs, and instancing operators to place specific door and window models and any pre-modeled building features. The building grammar allows operators to be modularized into reusable rules, and also supports language constructs to allow conditional and iterative processing.

Given such building grammar and the associated construction primitives, specific scripts may be written for each type of structure desired. Each script embeds specific rules for generation of that type of structure and exposes input parameters that can affect the generation of a specific building instance of that type. For example, a script to generate a church structure may include specialized routines to construct a steeple. Input parameters may guide the placement, size, and type of steeple, or even its inclusion in the final model. Lacking any specific inputs, a default church model will be created from a footprint. The script may utilize heuristics to derive the default steeple location from the footprint data.

The default script settings also allow pseudorandom variations based on seed values, so that a single script can generate large variations in external appearance and forms. When used to generate city blocks or neighborhoods, such variations allow generation of large tracts of buildings without repeating appearances.

Model Categories/Classes

Procedural building scripts are divided into classes, such as churches, parking garages, hangars, residential, commercial, etc. Examples of some of the building classes are shown in Figure 9.

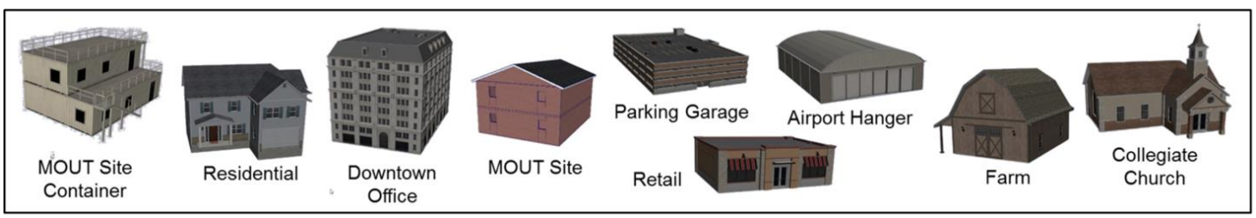

Figure 9 Examples of Procedural Building Classes

Subdividing Classes

Even among one class of building features, such as residential, it has proven useful to further subdivide and generate unique scripts for specific types of buildings in that class. Different types of residential structures have wildly varying appearances. Rules for generating mobile homes, ranch-style houses, or middle-east residences will have markedly different constraints and properties. Sample residential buildings of various types can be seen in Figure 10.

Figure 10 Examples of Procedural Residential Building Types

Procedural Model Advantages

Constructing models procedurally provides a number of advantages over building models using only photogrammetric techniques. Procedural models utilize geo-typical textures that can be reused for multiple models. These geo-typical textures are high resolution and meet requirements for close-up viewing and interaction. Because the textures are reused, overall texture storage requirements are minimized.

These geo-typical textures can contain rendering layers such as specular maps, normal maps, reflection maps, emissive maps, dirt maps, material maps and other components necessary to enable high-quality physically-based rendering on systems that support higher level rendering. Figure 11 and Figure 12 show representative exteriors and interiors using procedural model - rendered in the Unreal®[2]4 game engine.

Figure 11 Rendering w/Dirt and Normal Textures

Procedurally-generated models can have cut-in apertures for window openings and doorways. Such apertures may include window and door models that open and close. Procedurally-generated interiors can be coupled to each building-type script. These interiors respect aperture placement to prevent walls from being placed across apertures.

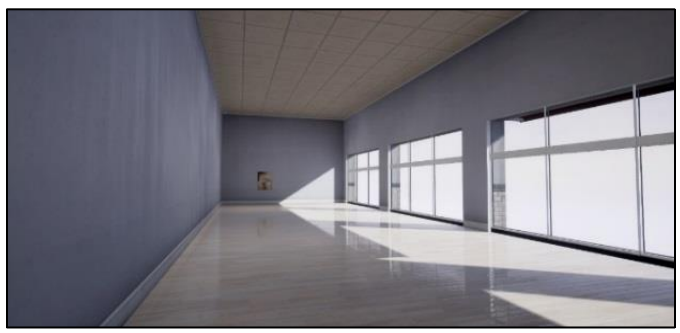

Figure 12 Rendering w/Normal Maps

RESEARCH PROCESSING PIPELINE

Given the maturity of AFE and PMG, our processing pipeline utilized existing building footprint feature data, either from traditional geographic information system (GIS) data sources or derived from LIDAR or Photogrammetric processing. Our research's processing workflow integrates well with the traditional terrain database workflow, exemplified by the SE Core production process. The Conform GIS tool from GameSim®[3] was utilized as the framework for our enhanced attribute extraction and expanded procedural model generation capabilities. Conform is currently used on SE Core for both automated feature modelization and procedural model generation.

[2] Unreal is a trademark of Epic Games, Inc. within the United States and/or other countries.
[3] GameSim is a registered trademark of GameSim, Inc. within the United States and/or other countries.

Working within the context of our site-visit use-case, we assumed that we could obtain geo-registered photographs for each building and that PMG art assets and construction scripts existed for generation of the target 3D models (or that we could develop them). In our experimental approach, we associated photographs with the appropriate footprint, processed the images to extract the desired building attributes, assigned these enhanced attributes to the appropriate footprint feature and PMG construction scripts, and finally generated the "digital twin" of the real-world 3D building. Figure 13 shows an example plan view where the red box icons indicated the photograph locations associated with the blue highlighted building footprint. Figure 14 displays a MOUT building with an extracted and procedurally-generated digital twin.

Figure 13 Footprint w/Geo-registered Images

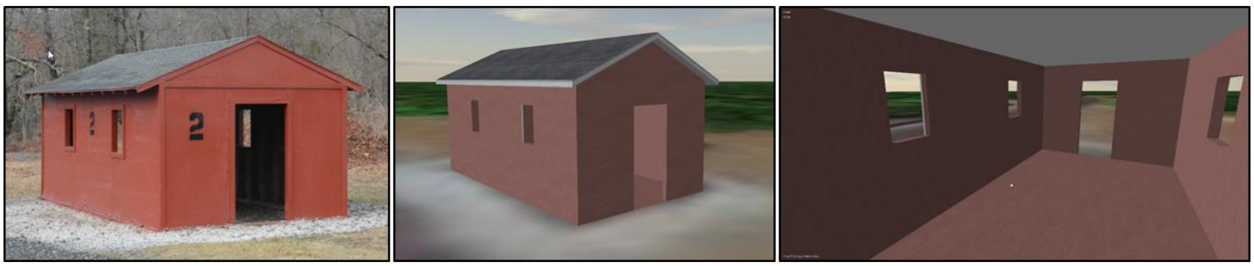

Figure 14 MOUT Building with Procedurally-Generated Digital Twin

EXTRACTION OF ENHANCED ATTRIBUTES FROM IMAGES

Two primary technologies were investigated for feature attribute extraction from imagery: 1) Image Processing and 2) Deep Learning. Both technologies have the potential to identify and extract the desired information.

Extraction using image processing

Our first experiments used image processing techniques such as region-boundary detection, Canny edge-detection, and Hough line-detection to identify wall locations from perspective camera images. Wall locations allowed generation of an intermediate ortho-rectified wall image, which was utilized to identify and extract apertures. The purple outline in Figure 15 shows a primary wall detected from a camera image of a small MOUT building using an area-fill function. Figure 16 and Figure 17 shows the results of the ortho-rectified images from the detected walls, from which the outline shows the window apertures, using edge-detection image processing.

Figure 15 Wall Detection

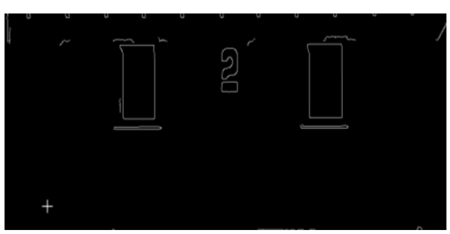

Figure 16 Edge Detection

Figure 17 Aperture Detection

The OpenCV toolkit was used as the primary platform for image processing functions. The identification and extraction of the desired features using image processing met with mixed results. Nevertheless, with footprint data available to assist in the processing, we were able to reliably extract the desired aperture data for roughly 80% of the site visit building images.

Deep Learning Image Extraction

Our second experiments utilized deep learning techniques via trained Convolutional Neural Networks (CNNs). A variety of CNN networks were trained to recognize and extract various building features, such as wall corners, aperture locations, roof types, exterior materials, and colors. As we experimented with CNN capabilities it became apparent that we could iteratively analyze the images, and on each iteration constrain both the image area and the CNN recognition to successfully extract the desired attributes. For the training site photographs, the CNN, once trained, proved very successful – 100% identification.

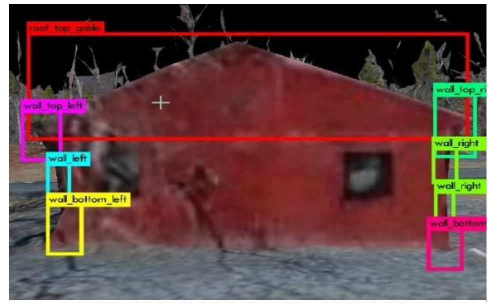

Figure 18 CNN Wall and Roof Detection

As discussed earlier, we also tested our extraction on a drone-collected photogrammetrically-derived terrain mesh of the same training area of our site visit photographs. To ensure consistency within our production process we created a viewer that could extract screenshots of the mesh as “virtual photographs.” We also created a depth buffer extraction tool to generate an image of the depth, again to ensure we could use all of the available information with a consistent approach.

One advantage of the terrain mesh was observed. We could use range-gating techniques to eliminate features that occulted the desired feature. Figure 18 illustrates the CNN results for wall and window detection stages. One advantage of CNNs is that they can be trained with lower resolution data, such as MOUT imagery derived from drone-captured data, and still give satisfactory results. Figure 19 shows building aperture detection using CNN with partially occulted and low-resolution source data.

Figure 19 Aperture Detection Low Resolution

Utilizing Neural Net Stages

In our final workflow, three distinct network stages were utilized. Stage One detected wall-corners, building-height, roof features, entrances, and external stairs and railings. Stage Two detected roof materials and colors, and exterior materials and colors. Neural nets for Stage Two were trained on small rectangular areas of material from the training instances, and thus returned many detected areas on the source imagery. The multiple color and material return values for all sides of the structure were compiled, and the most common return values were selected as the designated color and material for the entire building. Stage Three detected building apertures. Aperture size and location on each wall, in normalized wall-coordinates, were returned, along with each aperture type.

Several additional neural net stages were considered for inclusion. The addition of an initial neural net stage could be used to distinguish between broad types of building automatically, such as distinguishing between MOUT buildings, residential, containers, etc. The inclusion of a final CNN stage allowed the code to make determinations about the specific window and door models to be utilized for the apertures.

Training the Neural Nets

Neural nets were trained by utilizing training images which were manually marked with the locations of the features to be detected. Typically, anywhere from 25 to 100 marked images were utilized in training the nets. This was sufficient to demonstrate the proof-of-concept for our test cases.

One unique innovation involved automation of neural net training by utilizing existing PMG capability. In the process of creating artwork for a specific geographic location, we utilized PMG to create “generic” procedural models with the desired style for that location. Screen captures of these generic procedurally-generated models were then used as CNN training images. In the future, we propose to fully automate much of the neural net training by using these generic models, replacing the step of manually marking the training images with automated marking obtained from the knowledge of feature locations implicitly available from the procedural generation process. Figure 20 displays a procedural 3D model being used in training the neural net.

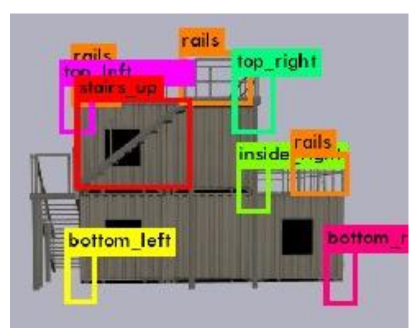

Figure 20 Training on PMG Models

PROCEDURAL GENERATION OF GEO-REPRESENTATIVE 3D MODELS

To illustrate the values of each stage and every attribute extracted we demonstrate incremental and progressive improvement to a geo-representative procedurally-generated model. With progressive attribute refinement, each additional piece of input information serves to refine the generated output, improving faithfulness to the desired real-world target. Figure 21 through Figure 31 show the incremental progress from default residential building to geo-representative building.

Residential house - Randomly-assigned Defaults

Figure 21 represented a procedurally-generated house based on a building footprint and a "residential house" building function type. The color and style are based on rules and randomly-selected building characteristics.

Figure 21 Default Residential House

Detection of Building Sides - One Story

We identify the wall outline for each building side. Using this outline, along with footprint edge-lengths, building height is computed from the image and passed procedurally to generation scripts. Figure 22 shows the results of front wall detection, transforming the default two-story building into a single-story building of the appropriate height, as seen in Figure 23.

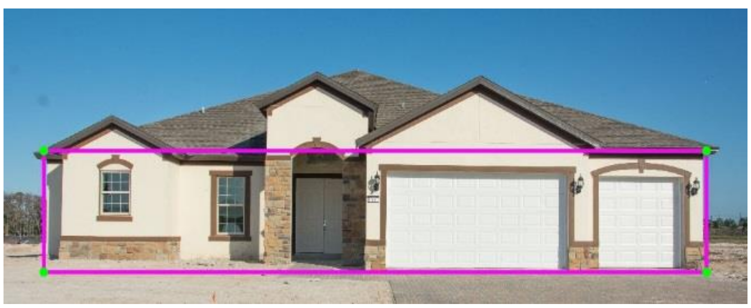

Figure 22 Front Wall Detection

Figure 23 Building Height

Detection of Exterior – Tan Stucco

We identify the exterior color and material by comparing selected areas on each building side to preexisting color/material palettes available to the scripts. Residential building materials include brick, siding, and stucco, each with a number of color variations. Figure 24 show results after a color/material of "tan stucco" was identified.

Figure 24 Color and Material

Detection of Roof Attributes – Brown Hip

Roof type (hip, gable, flat) is detected by assessing all available camera images and detecting key features that indicate the base roof-type (e.g., large gable covering one side). Figure 25 shows the step where hip roof-type is detected and passed to generation routines. Roof color and material are also detected.

Figure 25 Roof Type and Color

Detection of Aperture Locations – Right Side Garages

Apertures are identified with image processing and neural net techniques. Aperture type is determined heuristically, based on aperture size and location, or from neural net detection. Figure 26 shows apertures detected on the front of the residential building, while Figure 27 shows the results of the apertures in the procedural model.

Figure 26 Aperture Detection

Figure 27 Right Side Garages

Aperture information for each building side is passed to the procedural scripts. If no aperture information is included for a building side, default apertures are generated for the building side.

Detection of Gable Locations – Front Gables

Front facing gable roof sections are detected, as shown in Figure 28. Detected gable front protrusions are created based in the detected areas. Front entrances are also detected and used to designate the main entrance overhang and porch area. Figure 29 shows the resultant effects.

Figure 28 Gable Detection

Figure 29 Front Gables

Detection of Appendages – A/C Unit

We currently detect three types of appendages: chimneys, A/C units, and utility boxes. Once detected, location, type, and size information are passed to the procedural build state. Pre-built models for ac-units and utility boxes are instanced at the detected size in the designated areas. If detected, chimneys are procedurally built using script logic. Figure 30 shows the A/C unit CNN detection. Figure 31 shows the addition of the A/C unit to the procedural model.

Figure 30 A/C Unit Detection

Figure 31 A/C Unit

INTERIOR SPECULATION

The default interior building layout utilizes heuristics to assign room types and sizes. Some level of control is allowed per building type, but no specific room type assignments are made. Our "speculator" function adds the concept of interior speculation about room types by extraction of attributes from the building exterior. Some speculations were relatively straightforward— e.g., garage doors open into interior garages. Others may be more subtle—small windows may be associated with bathrooms, or upper floor windows may be associated with bedrooms. The speculator function assesses apertures and can pass an interior room hint with every aperture to the interior generation software. Figure

32 shows three apertures, where hints are being passed, and the resultant room generation with the hints. Speculation rules for each building type provide support for a variety of interiors.

Certain training tasks require building detail only in limited areas, such as the first few floors of a multiple story high rise building. Upper stories, may not require interiors or apertures. Generation of unnecessary detail can greatly increase polygon counts. Procedural generation can utilize input parameters to limit detail only to required areas. Using hints from the ground floor enhanced attributes can be used to create the ground floor layout.

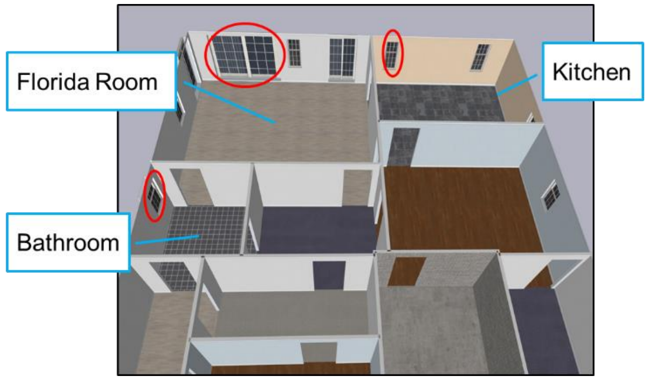

Figure 32 Interior Room Layout with hints

LESSONS LEARNED

Many lessons were learned as we worked through our experiment and developed our automated approach. Below we have identified a few key lessons:

Features are Flexible

Extracting information from photographs or photogrammetrically-derived terrain mesh, and representing the data as enhanced feature attributes makes it possible to:

- integrate the enhanced attribute data with existing geospatial feature data
- edit errors in extracted data
- generate variable fidelity 3D models for a variety of run-time systems
- reuse the data in the future to build higher fidelity models for run-time systems not yet built

Features are Efficient

Representing features in their abstract forms (verses polygons) are:

- more efficient to store
- much more efficient to stream
- easily tailored for use in simulated reasoning systems
- effortlessly transformed to be viewed by visual rendering systems
- suitable for analytical processing

Identifying Sub-Feature enables Speculation of Interiors

Speculating the interior layout of a building is readily enabled by identifying the apertures and appendages.

Procedural Modeling provides Flexibility

Generating 3D models procedurally allows for:

- changing the seasons – not constrained by the date of data capture
- fictitious location generation – easily change the feature data
- a consistent appearance and representation of the database regardless of available data

Deep Learning Performs Best Contextual

By constraining the problem-space for our deep learning tools, very reliable results are possible.

RECOMMENDATIONS AND FUTURE WORK

From our experiments, we have identified the need for a standard for procedural 3D model generation. We recommend a standard for rules and a standard for art asset generation. The Esri®[4] Computer Generated Architecture (CGA) specification provides such a set of rules (Esri, 2018). CGA's are the shape grammar of Esri's CityEngine, used to generate architectural 3D content. To complement the CGA standard, it is recommended that we define standards for

[4] ESRI is a registered trademark of Environmental Systems Research Institute, Inc. within the United States and/or other countries.

art assets that support the procedural generation tools. These standards need to define texture spatial resolution, wrapping and tiling schemes, map types, and any other attributes that ensure reusability.

Our future work will build upon the existing Conform framework, adding automation, expanding building types and refining feature and sub-feature extraction. We plan to add two additional stages to our Neural Net Stages. First, adding an initial neural net stage to distinguish between broad types of building automatically. And, second, adding a stage following the aperture and appendages stage to determine the specific window and door models to be utilized for the apertures. We plan to implement a fully automated neural net training system using procedurally-generated models, replacing the manual marking with automated marking obtained from the knowledge of feature locations implicitly available from the procedural generation process. We will expand the detection of appendages to support roof-vents, stand-pipes, and skylights.

ACKNOWLEDGMENTS
With great sorrow, we honor Khang Cat, originally our fourth author who unexpectedly passed away on April 12, 2018. Khang supported our research, providing expertise in photography and geo-located images. He will be missed.

REFERENCES

Collins, Robert (n.d.). CSE486, Penn State Lecture 25: Structure from Motion. Retrieved April 14, 2018. http://www.cse.psu.edu/~rtc12/CSE486/lecture25.pdf

Cura, Rémi (2016). Inverse procedural Street Modelling: from interactive to automatic reconstruction. Modeling and Simulation. Université Paris-Est, 2016. English. <NNT : 2016PESC1034>

Darell, Richard (n.d.). 3D Laser Scanning Technology Maps Interior & Exterior of Buildings. Retrieved April 14, 2018. http://www.bitrebels.com/technology/3d-laser-scanning-technology/

Deepakrishna, S, Ediriweera, S, & Gunatilake, A A J K (2016) AUTOMATIC FEATURE EXTRACTION FROM SATELLITE IMAGES USING LVQ NEURAL NETWORK. Asian Conference of Remote Sensing October 2016

Eckman, Stephen & Moore, Ronald & Johnson, Mark & Munro, Jaeson (2015). Automated Modelization in Terrain Database Production. Interservice/Industry Training, Simulation and Education Conference (I/ITSEC) 2015

Esri (n.d.). Computer Generated Architecture (CGA) shape grammar of CityEngine™. Retrieved April 14, 2018. http://desktop.arcgis.com/en/cityengine/latest/cga/cityengine-cga-introduction.htm

Hof, Robert D. (n.d.). MIT Technology Review Deep Learning. Retrieved April 27, 2018 https://www.technologyreview.com/s/513696/deep-learning/

Kern, Philip (2006) AUTOMATED FEATURE EXTRACTION: A HISTORY. MAPPS/ASPRS 2006 Fall Conference

Nishida, Gen, Bousseau, Adrien, & Aliaga, Daniel G. (2018) Procedural Modeling of a Building from a Single Image. EUROGRAPHICS 2018 Volume 37 (2018), Number 2

O'Neil-Dunne, Jarlat (2011) Automated Feature Extraction - The Quest for the Holy Grail. LiDAR Magazine, 2011 Vol. 1 No. 1, Page 18.

Spicer, Ryan, McAlinden, Ryan, & Conover, Damon (2016). Producing Usable Simulation Terrain from UAS-Collected Imagery. Interservice/Industry Training, Simulation and Education Conference (I/ITSEC) 2016

Tosh, Andrew, Snyder, Eric, & Ramos, Pedro (2016). Interactive Procedural Ground Texturing and Model Scattering for Synthetic Environments. Interservice/Industry Training, Simulation and Education Conference (I/ITSEC) 2016